Universidade Federal da Bahia
Instituto de Matemática

Programa de Pós-Graduação em Ciência da Computação

TOFFA-DAS: AN APPROACH TO CONDUCT TRADE-OFF ANALYSIS FOR DYNAMICALLY ADAPTABLE SOFTWARE

Michelle Larissa Luciano Carvalho

TESE DE DOUTORADO

Salvador, Bahia – Brasil
24 de Setembro de 2020

MICHELLE LARISSA LUCIANO CARVALHO

TOFFA-DAS: AN APPROACH TO CONDUCT TRADE-OFF ANALYSIS FOR DYNAMICALLY ADAPTABLE SOFTWARE

Esta Tese de Doutorado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Doutora em Ciência da Computação.

Orientador: Eduardo Santana de Almeida
Co-orientador: Paulo Cesar Masiero

Salvador, Bahia – Brasil
24 de Setembro de 2020

Ficha catalográfica elaborada pela Biblioteca Universitária de Ciências e Tecnologias Prof. Omar Catunda, SIBI - UFBA.

C331 Carvalho, Michelle Larissa Luciano.

ToffA-DAS: An Approach to conduct Trade-off Analysis for Dynamically Adaptable Software / Michelle Larissa Luciano Carvalho. – Salvador, Bahia - Brasil, 2020.

182 f.

Orientador: Prof. Dr Eduardo Santana de Almeida

Co-orientador: Prof. Dr Paulo Cesar Masiero

Tese (Doutorado) – Universidade Federal da Bahia. Instituto de Matemática e Estatística, 2020.

1. Software. 2. Engenharia de Software. 3. Computação. I. Almeida, Eduardo Santana de. II. Masiero, Paulo Cezar III. Universidade Federal da Bahia. IV. Título.

CDU 004

TERMO DE APROVAÇÃO

MICHELLE LARISSA LUCIANO CARVALHO

TOFFA-DAS: AN APPROACH TO CONDUCT TRADE-OFF ANALYSIS FOR DYNAMICALLY ADAPTABLE SOFTWARE

Esta Tese de Doutorado foi julgada adequada à obtenção do título de Doutora em Ciência da Computação e aprovada em sua forma f nal pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia.

Salvador, 24 de Setembro de 2020

Prof. Dr. Eduardo Santana de Almeida
Universidade Federal da Bahia (UFBA)

Prof. Dr. Paulo Cesar Masiero
Universidade de São Paulo (USP)

Prof. Dr. Rafael Augusto de Melo
Universidade Federal da Bahia (UFBA)

Profa. Dra. Rita Suzana Pitangueira Maciel
Universidade Federal da Bahia (UFBA)

Profa. Dra. Cecília Mary Fischer Rubira
Universidade Estadual de Campinas (UNICAMP)

Profa. Dra. Rossana Maria de Castro Andrade
Universidade Federal do Ceará (UFC)

Dedico esta tese aos meus pais, Magali M. Luciano Carvalho e Valmir de Jesus Carvalho.

AGRADECIMENTOS

"Tudo tem o seu tempo determinado, e há tempo para todo propósito debaixo do céu" (Eclesiastes 3:1).

Sinto alívio e satisfação por ter conseguido finalizar o doutorado com êxito. Me privei de tantas coisas para realizar essa etapa na minha vida, mas valeu a pena. Os desafios me fizeram amadurecer e aprender bastante.

Agradeço primeiramente a Deus, pois Ele me deu forças para não desistir e sabedoria para fazer um trabalho com diligência.

Agradeço aos meus amados pais Magali e Valmir pelo apoio incondicional e paciência. Meus pais são fundamentais para o meu crescimento pessoal e evolução acadêmica.

Agradeço a minha irmã Laísa pelos conselhos e companhia. Como minha melhor amiga, minha irmã está sempre presente e me dá os melhores conselhos.

Agradeço a cada um dos meus amigos e familiares pela compreensão. Eles sempre me deram apoio e entenderam minha ausência em diversos momentos.

Agradeço ao meu orientador Eduardo Almeida por ter acreditado em mim do começo ao fim. Amadureci muito com os ensinamentos.

Agradeço também ao meu co-orientador Paulo Masiero. Todos os ensinamentos foram extremamente importantes para minha formação.

Agradeço aos colaboradores e colegas do grupo *RiSE*. A ajuda de cada um foi essencial para fechar esse ciclo.

Finalizo meus agradecimentos com as seguintes palavras: *"O importante é saber o que quer. Sem objetivos não existe progresso. É preciso ter desejos, sonhos, planos. Se você tiver uma visão e de fato implementá-la, estará lançando as bases do seu sucesso."*(William Douglas e Rubens Teixeira)

"You cannot be anything you want to be. But, you can be everything God wants you to be."

—MAX LUCADO

RESUMO

Os processos de engenharia das linhas de produtos de software dinâmico (LPSD) visam projetar software dinamicamente adaptável (SAD), aumentando a flexibilidade para a geração de um grande número de configurações. Isso resulta em uma explosão de espaço de configuração do software, tornando a análise mais difícil e complicando o trabalho do desenvolvedor. Nesse sentido, os engenheiros de software precisam encontrar uma combinação de funcionalidades do sistema que possam satisfazer simultaneamente as restrições especificadas em modelos de funcionalidades e de contexto, requisitos não funcionais (RNFs) e preferências das partes interessadas. Isso significa que eles têm que medir várias configurações até encontrar as viáveis, caracterizando o processo de configuração do produto em um problema de otimização complexo. A maioria dos estudos existentes não focam nas interações entre as informações contextuais e os RNFs ao lidar com a seleção de funcionalidades para atender aos objetivos de qualidade desejados. Além disso, tais estudos não usam nenhuma estratégia de planejamento para apoiar o processo de seleção de configuração. Com base nessas lacunas de pesquisa, propomos uma abordagem que *(i)* gerencia as funcionalidades e contextos do sistema; *(ii)* facilita a compreensão de como os produtos de uma LPSD podem se comportar a partir de uma determinada mudança de contexto, e *(iii)* permite realizar análise trade-off a fim de encontrar configurações válidas e viáveis, que atendam as restrições e as interações entre as informações contextuais e RNFs. Com o objetivo de apoiar a modelagem de variabilidade de contexto, propusemos a técnica estendida de modelgame de funcionalidade sensível ao contexto (eCFM) para lidar com restrições entre contextos. Em seguida, definimos a abordagem Análise Trade-off para SAD (ToffA-DAS) para lidar com o processo de seleção de configuração que abrange as interações entre as informações contextuais e os RNFs. Também propusemos uma estratégia para analisar as mudanças de contexto, a fim de definir modelos de adaptação para cada priorização das funcionalidades do sistema, contextos e RNFs. Por fim, evoluímos nossa abordagem e chamamos de ToffA-DAS PLUS (ToffA-DAS+). O ToffA-DAS é baseado na técnica de programação linear inteira, enquanto o ToffA-DAS + usa um algoritmo genético. Realizamos um conjunto de estudos empíricos a fim de avaliar a proposta desta tese. Primeiramente, realizamos uma pesquisa para avaliar o eCFM do ponto de vista da expressividade para modelar as restrições de contexto e facilidade de uso. Na verdade, a análise foi focada na compreensibilidade da modelagem de variabilidade contextual. Em seguida, realizamos um estudo baseado em simulações para reunir evidências iniciais sobre a viabilidade do uso do ToffA-DAS. Esse estudo é baseado em como conduzir uma análise de trade-off e definir modelos de adaptação a partir das configurações viáveis encontradas na análise. Também realizamos um estudo exploratório para avaliar como as configurações obtidas pela execução do ToffA-DAS afetam o nível de satisfação geral dos RNFs. Por fim, avaliamos a evolução da nossa abordagem em comparação com a versão anterior. Como resultado do primeiro estudo, o eCFM foi considerado uma técnica com grande expressividade para representar regras de adaptação entre contextos e funcionalidades do sistema, além da facilidade de uso e organização com o agrupamento de contextos. Portanto, concluímos que os engenheiros de software podem levar em consideração o uso da técnica eCFM para modelar SAD. No segundo estudo, ToffA-DAS apresentou resultados consistentes de acordo com os cenários do mundo real, satisfez os valores de utilidade estimados e as restrições do modelo. O terceiro estudo mostrou que o conjunto de configurações gerado pela execução do ToffA-DAS proporciona altos níveis de satisfação dos RNFs. No último estudo, coletamos evidências de que o ToffA-DAS+ sugere mais soluções a partir de possíveis configurações válidas do modelo. Com base nos estudos mencionados acima, evidenciamos que nossa abordagem pode ser útil quando os engenheiros de software precisam de ajuda na compreensão de como projetar uma variedade de opções configuráveis para LPSD. É baseado no princípio de que cada opção de configuração deve ser viável para atender a certas mudanças contextuais sem perder a qualidade do serviço. Com o uso de nossa abordagem, os engenheiros de software podem analisar e simular exaustivamente uma solução antes de implementá-la.

Palavras-chave: Software dinamicamente adaptável, linhas de produtos de software dinâmicas, gerenciamento de variabilidade, análise de modelo de funcionalidades, engenharia de requisitos, otimização, modelo de adaptação

ABSTRACT

The Dynamic Software Product Lines (DSPL) engineering processes aim to design Dynamically Adaptable Software (DAS) by increasing the flexibility for the generation of a huge number of configurations. It results in a software configuration space explosion making the analysis more difficult and complicating the developers work. In this sense, software engineers need to find a combination of systems features that can simultaneously satisfy constraints specified in feature and context models, Non-functional Requirements (NFRs), and stakeholders preferences. It means that they have to measure many configurations until finding the feasible ones, characterizing the product configuration process in a complex optimization problem. Most of the existing studies do not focus on the interactions between contextual information and NFRs when dealing with feature selection to meet the desired quality objectives in DAS. In addition, such studies do not use any planning strategy to support the configuration selection process. Based on these research gaps, we propose an approach that *(i)* manages the systems features and contexts; *(ii)* facilitates the understanding of how DSPL applications can behave from a certain context change, and *(iii)* enables to conduct trade-off analysis in order to find valid and feasible configurations, which meet the constraints and the interactions between contextual information and NFRs. Aiming to support the context variability modeling of DAS, we proposed the Extended Context-aware Feature Modeling (eCFM) technique to deal with constraints among contexts. Next, we defined the DAS Trade-off Analysis (ToffA-DAS) approach to deal with the configuration selection process embracing interactions between contextual information and NFRs. We also proposed a strategy to analyze context changes in order to define adaptation models for each prioritization of the systems features, contexts, and NFRs. Finally, we evolved our approach and named it as DAS Trade-off Analysis PLUS (ToffA-DAS+). ToffA-DAS is based on the integer linear programming technique, whereas ToffA-DAS+ uses a genetic algorithm. We performed a set of empirical studies in order to evaluate the proposal for this thesis. First, we conducted a survey to evaluate eCFM from the viewpoint of expressiveness to model the context constraints and easiness of use. Indeed, the analysis was focused on the comprehensibility of contextual variability modeling. Next, we performed a study based on simulations to gather initial evidence about the feasibility of using ToffA-DAS. It is based on how to conduct trade-off analysis and define adaptation models from feasible configurations found in the analysis. We also conducted an exploratory study to evaluate how the configurations obtained by the execution of ToffA-DAS affect the overall satisfaction level of NFRs. Finally, we evaluated the evolution of our approach in comparison with the previous release. As a result of the first study, the eCFM was considered a technique with a great expressiveness to represent adaptation rules among contexts and system features, besides the easiness of use and organization with the grouping of contexts. Therefore, we argue that the software engineers may take into account the use of eCFM technique to model DAS. In the second study, ToffA-DAS presented consistent results in accordance with the real-world scenarios and satisfied the estimated utility values and model constraints. The third study showed that the set of configurations generated by ToffA-DAS execution provide high satisfaction levels of NFRs. In the last study, we collected evidence that ToffA-DAS+ suggests more solutions from then possible valid configurations of the model. Based on the aforementioned studies, we evidenced that our approach can be handy when software engineers need assistance in the understanding of how to design a variety of configurable options for DSPL applications. It is based on the principle that each configuration option must be feasible to meet certain contextual changes without losing service quality. With the usage of our approach, software engineers can exhaustively analyze and simulate a solution before implementing it.

Keywords: Dynamically adaptable software, dynamic software product lines, variability management, feature model analysis, requirement engineering, optimization, adaptation model

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CCF	Combination of context features
CFM	Context-Aware Feature Model
CSP	Constraint satisfaction problem
DAS	Dynamically Adaptable Software
DSPL	Dynamic Software Product Lines
eCFM	Extended Context-aware Feature Modeling
FODA	Feature-oriented Domain Analysis
GA	Genetic algorithm
GORE	Goal-oriented Requirement Engineering
ILP	Integer Linear Programming
MAPE-K	Monitor, Analyze, Plan, Execute, and Knowledge
NFRs	Non-functional Requirements
SPL	Software Product Lines
SPLE	Software Product Lines Engineering
SPLOT	Software Product Lines Online Tools
SXFM	Simple XML Feature Model format
TGMC	Tropos Goals Model
ToffA-DAS	Trade-off Analysis for Dynamically Adaptable Software
VMTs	Variability Modeling Techniques

PART I

OVERVIEW

Chapter 1

The past does not have to be your prison. You have a voice in your destiny. You have a say in your life. You have a choice in the path you take. — Max Lucado

INTRODUCTION

Dynamic Software Product Lines (DSPL) engineering is a paradigm aimed at providing software systems with the capability of handling adaptations at runtime. A DSPL application monitors the environment and adapts its behavior according to changes in the execution environment or user requests. Then, software engineers have used DSPL engineering to provide support for developing Dynamically Adaptable Software (DAS) by capturing and managing dynamic variability. In fact, a DAS itself is seen as a DSPL application [7].

Dynamic variability occurs due to product variations in accordance with the contextual changes [8]. Thus, it consists of the application customization that can happen at runtime. This customization to context changes corresponds to product derivation in the product line terminology [9]. In this scenario, DSPL applications should be capable of handling context identification and must be prepared to dynamically adapt their behavior to meet distinct scenarios. Such adaptations comprise activating and deactivating software system features [8]. This chapter consists of six sections:

Section 1.1 introduces and motivates this study;

Section 1.2 discusses the objectives of the thesis and presents our research question;

Section 1.3 presents the research design, which involves background, approach, and empirical studies;

Section 1.4 presents the main contributions of this work;

Section 1.5 defines the topics out of the scope; and

Section 1.6 finally presents a roadmap about the chapters of this proposal.

1.1 MOTIVATION

According to Bencomo et al.[10], DSPL applications should be prepared to deal with the following dimensions of variability: *structural variability* and *environment* or *context variability*. These dimensions can be modeled as *variation points*, *i.e.*, as specific locations where decisions can be made to express *variants* (application configurations) [11]. In the first dimension, the variation points represent different configurations that can be derived by considering the constraints defined in the model. Thus, it refers to the configuration of the system's features. In the second dimension, the variation points represent the properties of the environment. It represents the variants of contextual information relevant to the system based on the environment where it resides [4, 12].

Although DSPL application customization occurs at runtime, it is important that the software engineer identifies at design time the possible adaptations and the information, which can affect such customization. This encompasses *variability modeling*, which consists of one the most important activities in DSPL engineering [4]. It aids software engineers in handling diverse information, such as contexts and non-functional requirements (NFRs). Contexts are information computationally accessible and upon which behavioral variations depend [13], whereas NFRs consist of internal system properties and are recognized as an important factor for the success of software projects [14].

Sousa et al.[15] reported a set of research opportunities related to quality evaluations of DAS, such as the definition of thresholds for quality measures, the development of approaches for prioritization of NFRs according to DAS operations and domains, besides the conflict mapping among NFRs. The authors state that the system quality evaluation is not a trivial task and must be made not only at run-time but also at design time to check the system capacity to meet self-adaptive operations. Dealing with NFRs makes the configuration selection process more difficult since such properties tend to be qualitative and may not be easily mathematically quantifiable (*e.g.*, specifying resiliency and efficiency) [16, 17]. Quantifying NFRs often relies on domain knowledge and may not be feasible in a specific DAS, which must meet certain contextual changes [17].

Indeed, among the major challenges that software engineers face with regard to the modeling and development of DSPL applications [4, 18, 19] the following are highlighted: *(i)* identification and modeling of the context variability that influences the dynamic behavior [4]; *(ii)* prediction of a set of possible dynamic adaptations that may occur in accordance with several changes in the environment [4, 18]; *(iii)* analyzing the existing or new constraints in the feature model to avoid unfeasible products [20]; and *(iv)* the accurate representation of the impact of features over contexts and NFRs aiming to identify feasible configurations, which satisfy the stakeholder's preferences and constraints [19].

It is necessary to deal with the dimensions of *structural variability* and *context variability* since unanticipated conditions can trigger failures and inconsistent reconfiguration at run-time. In order to avoid such failures and inconsistencies, software engineers can simulate at design time, a set of possible adaptations that meet the variability dimensions. Next, s/he can choose which adaptations should be developed based on those that satisfy specific quality requirements. These dimensions are handled using the DSPL engineering processes in order to model and systematically develop applications by exploring adaptability [21].

Some existing Variability Modeling Techniques (VMTs) are used to map and manage both dimensions, *structural variability* and *context variability*. Nevertheless, such VMTs are limited regarding to expressiveness for modeling the *context variability*, *i.e.*, it lacks the definition of constraints among contexts [22, 23]. It is an important property since in the real environment there are contexts that cannot occur at the same time. Thus, software engineers need to rely on a VMT that copes with such context representation.

The main objective of designing DAS using the DSPL engineering processes is increasing flexibility for application customization. Such flexibility is related to the generation of a huge number of configurations and helps making DAS extensible, besides achieving a good quality of service. However, those numerous configurable options result in a software configuration space explosion and lead to real challenges to developers. This explosion makes the analysis more difficult, as different configurations can be conflated together and generally complicates the application understanding tasks [24].

Software engineers must have to find a system's feature combination that can simultaneously satisfy constraints specified in feature and context models, NFRs, and stakeholder's preferences. It means that they have to measure many configurations until finding the feasible ones. Thus, the product configuration process in DSPL engineering can be viewed as a complex optimization problem [19]. When dealing with feature selection to meet desired quality objectives in DAS, most of the existing studies are not focused on the interactions between contextual information and NFRs. In addition, such studies do not use any strategy to support the selection of the most suitable configuration [25].

1.2 OBJECTIVE

The objective of this work is to propose an approach that *(i)* manages both dimensions, *structural variability* and *context variability*; *(ii)* facilitates the understanding of how DSPL applications can behave from a certain context change; and *(iii)* enables to conduct trade-off analysis in order to find feasible and valid configurations that meet the constraints and the interactions between contexts and NFRs. Thus, we describe the specific objectives defined in this proposal in the following:

Research Goal 1: Define a VMT to represent context constraints and favor the model organization and comprehensibility.

Research Goal 2: Propose a decision support approach to deal with the configuration selection process of DAS.

Research Goal 3: Develop and empirically evaluate an optimization model to identify feasible configurations that better meet stakeholder's preferences, the variability of system features, contexts, NFRs, and constraints.

On the basis of such defined goals, we established the research question that drives this investigation:

> **How to precisely express the impact of the system's features over NFRs and contexts aiming to identify feasible configurations that satisfy stakeholder's preferences in DAS projects?**

Such an approach can be handy when software engineers need assistance in the understanding of how to design a variety of configurable options for DAS projects. It is based on the principle that each configuration option must be feasible to meet certain contextual changes without losing service quality.

1.3 RESEARCH DESIGN

This section describes the research design employed as the basis of this work. We split this investigation into three main parts: Background; Variability Modeling; and Configuration Selection Process. Figure 1.1 shows a diagram with these macro parts and an overview of the sub-activities, which we detail next.

Background. The initial part comprises an overview of the basic concepts from SPL engineering as well as DSPL engineering. In addition, it is composed of relevant topics to construction and execution of this research such as the connection between SPL and DSPL fields, requirements engineering, planning the development of DSPL projects, and optimization methods.

Variability Modeling. The second part comprises the proposal of a technique to model context variability and its evaluation. As a means of better understanding the Extended Context-aware Feature Modeling (eCFM) technique, we present the steps to model DAS projects by using it. We also recommend some tips to software engineers that intend to model DAS with the eCFM technique. Furthermore, we performed a survey to assess the comprehensibility and easiness of use of this technique.

Configuration Selection Process. The third part comprises the definition of an approach to support the trade-off analysis in DAS projects. It uses the *Utility-based* planning to formalize the knowledge obtained from stakeholder's preferences, the variability of system features, contexts, NFRs, and constraints. To this end, we defined an optimization model to ensure feasible configurations to satisfy the requirements. We then proceed with the evaluation of the feasibility of using the approach and defined adaptation models that can be eventually developed. We also conducted an exploratory study when our approach was assessed regarding the satisfaction of NFRs. Next, we decided to evolve our approach and conduct an additional empirical study to compare both releases.

1.4 STATEMENT OF THE CONTRIBUTIONS

We evidenced in our previous study [26] that a DSPL application should also deal with evolution in terms of functionality and adaptation capabilities when the demand arises for new requirements for existing applications or new configurations. Such efforts of the previous work do not handle the interactions between contexts and NFRs and do not use any strategy to provide support and guidance to the software engineers in finding configurations in accordance with stakeholder's requests. In addition, it does not address the challenges related to the configuration selection process, as presented in Section 1.1. Based on that, we decided to investigate DSPL engineering from a variability-modeling perspective.

We designed a comprehensive approach to support the trade-off analysis of DAS projects, named Trade-off Analysis for Dynamically Adaptable Software (ToffA-DAS). It deals with the configuration selection process considering interactions between contextual information and NFRs and identifies the feasible and valid configurations by embracing *domain analysis*, *modeling*, *prioritization*, *contribution*, and *optimization*. In this sense, we used *utility-based* planning as a strategy to deal with trade-off analysis. Although it is focused on supporting software engineers at design time, our approach can be feasible in the development of DAS.

Based on the artifacts generated from the execution of ToffA-DAS, it is possible to define dynamic adaptation models that meet specific scenarios. These models represent the transitions among contextual changes that will fulfill the satisfaction of relevant NFRs and also suit as a baseline to implement the system to be adapted at run-time. Therefore, with the usage of our approach, software engineers can exhaustively analyze and simulate a solution before implementing it. The contributions of this work are described in more detail, as follows:

- *Challenge (i) and (ii)* - Aiming to support the context variability modeling of DAS projects, we proposed the eCFM technique [27]. It consists of an extension of the Context-Aware Feature Model (CFM) [28] to deal with constraints among contexts states. For modeling DAS with eCFM, the software engineer must perform context analysis to identify context-aware properties and how they affect the system configuration;

- *Challenges (iii) and (iv)* - We defined the ToffA-DAS approach to deal with the configuration selection process embracing interactions between contextual information and NFRs. Such an approach uses a feature model with constraints among context, which is specified through eCFM technique. In addition, ToffA-DAS is based on a utility function that makes it possible to define a weighted mean of the differences between properties describing the service provided by the application and properties representing user needs, *i.e.*, the weights (utility values) represent stakeholder's priorities [9]. Based on the utility values, we define an optimization model that recommends feasible configurations by considering a diversity of adaptation rules and constraints of the feature model;

- *Challenge (iv)* - We proposed a technique to conduct trade-off analysis in DAS projects by changing the prioritization of modeling elements. This is important due to interactions among adaptations triggered by different contexts activated at the same time. Indeed, such a trade-off analysis should be done during the requirements specification, and decisions should be taken by the software engineers aiming to satisfy the needs of their stakeholders;

- *Challenge (ii)* - We also proposed a strategy to analyze context changes in order to define adaptation models. Thus, software engineers can define such adaptation models for each prioritization of contexts and NFRs. The advantage of this model is to support predicting a set of possible adaptations at design time, then it can be eventually developed and handled at runtime;
- *Challenge (iii) and (iv)* - We presented the DAS Trade-off Analysis PLUS (ToffA-DAS+), an evolution of the ToffA-DAS approach. ToffA-DAS is based on the Integer Linear Programming (ILP) technique to deal with a mono-objective problem, whereas ToffA-DAS+ uses a genetic algorithm (GA) and the SAT solver technology. Such changes enable to handle a multi-objective optimization problem and simplify the feature model satisfiability analysis.
- *Challenge (iv)* - We implemented a tool to provide support for the usage of ToffA-DAS+ and tackle the configuration selection problem. By using our approach in combination with the tool, the software engineer can model the DAS projects and handle the trade-off between contextual information and NFRs by evaluating and simulating a set of feasible solutions.

We provided the source code of the optimization model based on an ILP solver [29], the tool based on a GA and SAT solver, besides all the data used to run the studies[1] for replication and further details. Table 1.1 shows a list of the publications related to the thesis topic in order to get an overview of our contributions so far. Moreover, we are currently submitting other papers to report the remaining results.

1.5 OUT OF SCOPE

The following topics are out of the thesis scope:

Tool support. Automation of all the configuration selection process described in this work.

Trade-off. Investigation of the feasibility of using the approach for trade-off analysis during the execution of DAS.

Other strategies. Employment of the other strategies, such as *Rule-based* and *Goal-based* for dealing with the configuration selection process [31].

1.6 ORGANIZATION OF THE THESIS

This thesis proposal is structured in four parts and two appendices. Figure 1.1 shows a schematic overview of the thesis proposal structure. Apart from the present Introduction Part, the remainder can be outlined in the following way:

[1] https://sites.google.com/view/dspl-life-cycle/home

Table 1.1: Publications during the Ph.D. research.

Paper Title	**Venue**	**Participation**
Thesis related publications		
Dynamically Adaptable Software is all about Modeling Context Variability and Avoiding Failures [27]	**IEEE SOFTWARE'17**	*Significant*
ToffA-DAS: Trade-off Analysis for Dynamically Adaptable Software (Under review)	**JSS'19**	*Major*
ToffA-DAS+: An approach to identify feasible and valid configurations using genetic algorithm and the SAT solver technology (Under review)	**IST'20**	*Major*
Related topics publications		
On the Implementation of Dynamic Software Product Lines: A Preliminary Study [30]	**SBCARS'16**	*Major*
On the implementation of dynamic software product lines: An exploratory study [26]	**JSS'18**	*Major*

Part II - Background: This part provides background concepts on the topics involved in this investigation.

Chapter 2 (Main Concepts and Foundations): Basic concepts regarding Software Product Lines (SPL) and DSPL engineering. It also presents an overview of the differences and similarities between both, SPL and DSPL. In addition, it discusses concepts related to requirement engineering and planning of the DAS projects. Finally, the concepts regarding optimization are discussed.

Part III - Approach and Empirical Studies: This part motivates and defines in detail the extended technique to model *context variability* and an approach to identify valid and feasible configurations in DAS projects. In addition, it reports the empirical studies performed to evaluate such a technique and approach.

Chapter 3 (Extended Context-aware Feature Modeling Technique): We present the extended Context-aware Feature Modeling (eCFM) technique and findings of the survey.

Chapter 4 (Trade-off Analysis for Dynamically Adaptable Software): We present our approach by describing five steps that the software engineer should perform in order to deal with configuration selection process of DAS projects.

Chapter 5 (Feasibility of using the ToffA-DAS approach): We discuss the insights concerned with configuration selection process of DAS projects. In addition, we describe how our approach can support software engineers for the reasoning of adaptability and present an exploratory study addressed to evaluate our approach.

Chapter 6 (Evolution of the ToffA-DAS approach): We describe how to combine ToffA-DAS with GA and SAT solver technology to support the configuration selection process of DAS projects. Besides, we present an exploratory study that evaluates the evolution of approach, named ToffA-DAS+.

Part IV - Conclusions: This part concludes the thesis.

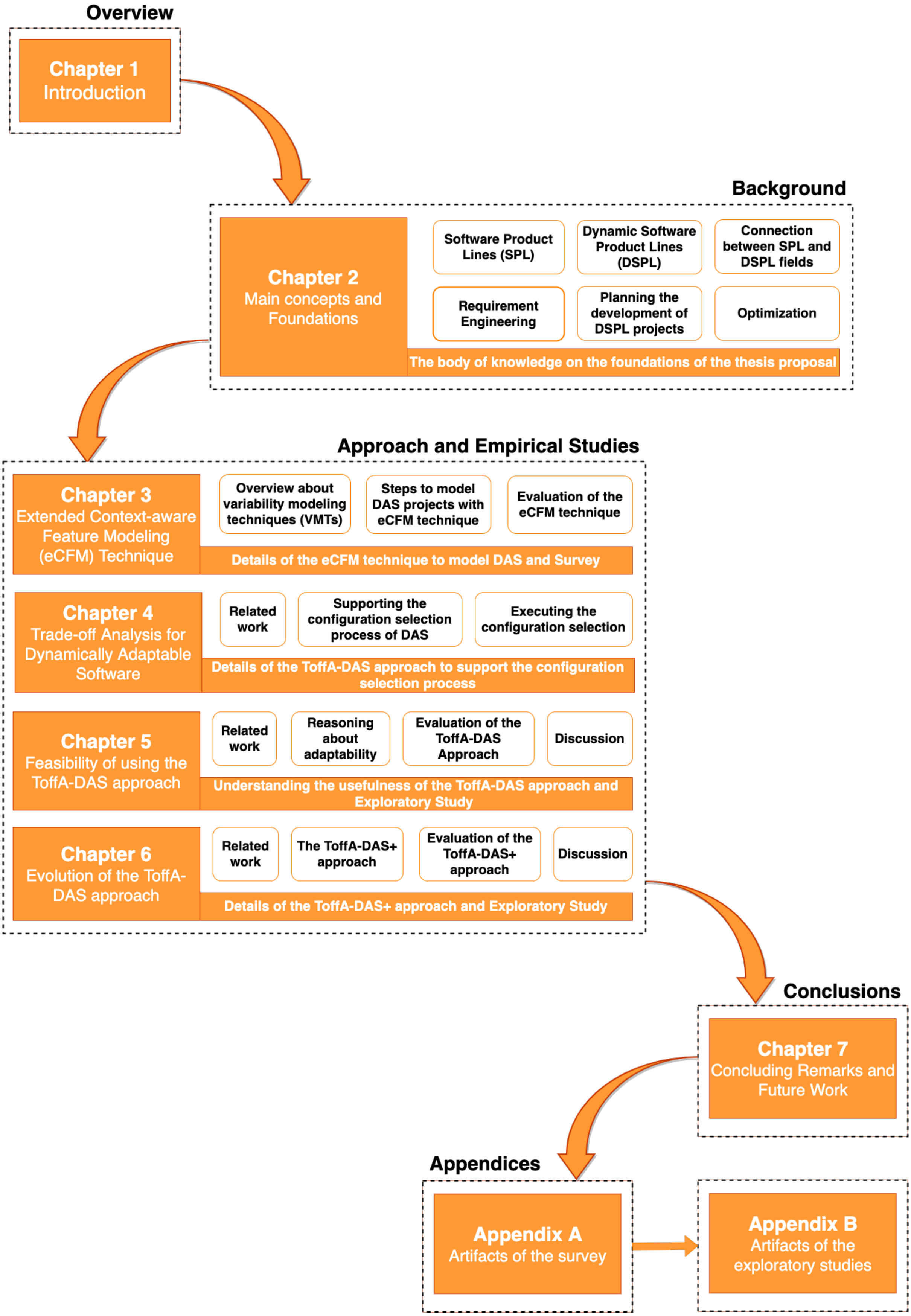
Chapter 7 (Concluding Remarks and Future Work): This chapter summarizes and outlook on steps towards accomplishing the thesis investigation. It also includes the empirical studies observations and further steps to be performed as future work.

Part VI - Appendices: Finally, this part presents the artifacts used in the survey and exploratory studies.

Appendices A (Artifacts of the Survey)

Appendices B (Artifacts of the Exploratory Studies)

Figure 1.1: Schematic overview of the thesis structure.



PART II

BACKGROUND

Chapter 2

One of the major keys to success is to keep moving forward on the journey, making the best of the detours and interruptions, turning adversity into advantage. –John C. Maxwell

MAIN CONCEPTS AND FOUNDATIONS

The competitiveness of the market and diversification in software development has been a key issue for employing new engineering practices. In this way, some companies in the software industry have adopted the Software Product Lines Engineering (SPLE) approach aiming at faster product development with high quality and low cost. It consists of an emergent software engineering paradigm promoting reuse through the software life cycle. SPLE allows that development companies supply the large demand for software systems using platforms and mass customization [32].

The SPLE cycle includes two main processes [32]: *domain engineering* and *application engineering*. While the former aims to define which artifacts are common and which ones are variable, the latter derives the products by using the common and variable artifacts defined in domain engineering. The local where the variation occurs represents the variation points that allow to include variable artifacts. The variation ability is known as *variability*. It consists of a characteristic, which can be common only in some products [33]. Moreover, this variability is often expressed in terms of features, and it also appears to be high-level abstractions that shape the reasoning of stakeholders [34].

Variability management is an important activity that differentiates SPL from conventional software engineering. Its purpose is to identify, design, implement, and trace the flexibility in the SPL. The development of product lines, however, needs to be adapted to new requirements given the emergence of new technologies and services which cope with a flexible adaptation of software and changing needs.

The goal of this chapter is to describe the basis for understanding the connection between both, SPL and DSPL fields, as well as, context variability modeling, planning strategies that are employed for selecting desirable system features, and optimization strategies. The remaining of the chapter consists of three main sections, as follows:

Section 2.1 presents the concepts related to SPL;

Section 2.2 presents the concepts related to DSPL;

Section 2.3 presents an overview of the differences and similarities between both, SPL and DSPL. In addition, it discusses open issues in DSPL engineering by comparing it with the SPL engineering;

Section 2.4 provides background concepts on the requirements engineering;

Section 2.5 describes context variability modeling in DSPL engineering and planning types;

Section 2.6 presents the concepts related to optimization; and

Section 2.7 presents the chapter summary.

2.1 SOFTWARE PRODUCT LINES (SPL)

According to Northrop[1], Software Product Lines are families of software products that share a set of common features and they can efficiently attend to software mass customization. SPLs are developed based on core assets, *e.g.*, reusable software components, domain models, architecture description, requirements statements, specifications, and documentation. Each member of the SPL is known as a variant, which is instantiated according to its needs and rules of the common architecture. The reference architecture consists of a large number of components that can be connected through interfaces and provide support to mass customization [32].

The concept of software families was first introduced by Dijkstra[35] who proposed a model of family-based development, where differences in design decisions distinguished family members. Parnas[36] characterized families as groups of items that are strongly related by their commonalities, where commonalities are more important than the variations among family members. In addition, Kang et al.[37] contributed to the SPL concept introducing the Feature-oriented Domain Analysis (FODA) method. The FODA method uses the result of the identification and classification of commonalities and variabilities in the software domain to build assets.

The purpose of the use of SPLE is to reduce the engineering overall effort to produce a set of similar products, through planned software reuse. The fundamental principles of SPLE practices consist in identifying the common and variable parts and supporting a range of products to maximize reusable variations and eliminate waste of generic implementation of components used only once [21].

2.1.1 Essential Activities and Benefits

The development of an SPL involves three essential activities: *Core Asset Development, Product Development* and *Management*, which are showed in Figure 2.1. Each rotation circle represents one key activity. All three are connected together as they would in motion, showing that all three are closely related and highly interactive.

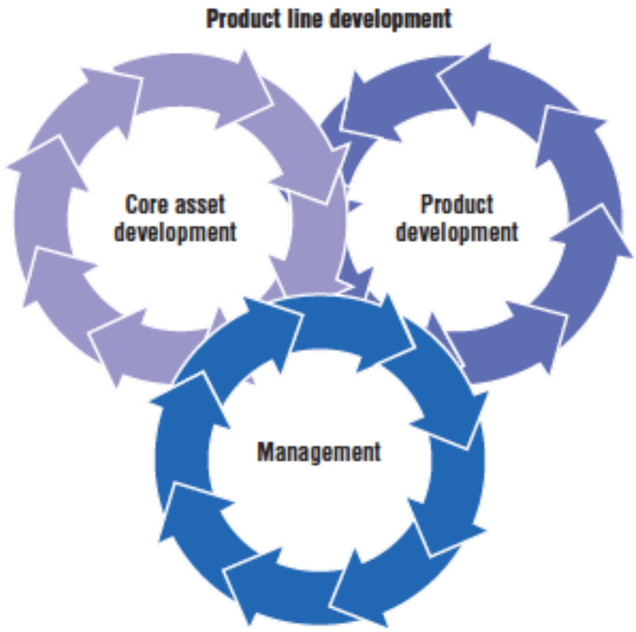


Figure 2.1: Essential product line activities from Northrop[1]

- **Core Asset Development** – It consists of analyzing the SPL domain and searching the reuse opportunities within of the scope, motivated by the domain engineering;

- **Product Development** – It is part of the application engineering and focuses on combining and configuring the core assets to develop specific components to instantiate a new product; and

- **Management** – It is an activity that supports and coordinates the Core-Asset activity and Product Development. The management throughout the development encompasses the separation of the product line in three parts: *common components*, *variable parts*, and *individual products* as their own specific requirements.

Linden, Schmid e Rommes[33] stated that companies usually adopt the SPL approach strongly based on economic considerations. SPL supports large-scale reuse, which implies lower costs, shorter time to market, and improve the quality of the resulting products. Despite these benefits, it is necessary for some initial investment, which requires to build reusable assets changing the organization. The use of the SPL usually reaches a break-even after about three products.

Pohl at al. [32] define some benefits in adopting the SPL approach:

- **Reduction of development cost** – When core assets are reused in different products generated, the cost to create all the systems is reduced. The costs to develop a few systems in SPLE are higher than in traditional software engineering. However, using the SPL approach, the costs are significantly lower for larger systems quantities;
- **Quality improvements** – Reusable assets are more tested as the number of products increases. This implies significantly in the detection and correction of faults, thereby increasing product quality;
- **Reduction of time-to-market** – The product release time is significantly reduced when it relies on the SPL approach. Initially, the time is high because it is necessary to develop reusable artifacts. Afterward, the time to market is reduced because many artifacts are reused to build new products; and
- **Reduction of Maintenance Effort** – When an artifact is modified, the changes are propagated for all products. In SPL, it is possible to reuse test procedures that decrease the maintenance effort.

2.1.2 Commonalities and Variabilities in SPL

SPLE establishes a systematic software reuse strategy. The goal is to identify commonality (common functionality) and variability points among applications within a domain and build reusable assets to benefit future development efforts [32]. Core assets are considered the essence of the SPL [1]. They consist of configurable elements of an SPL such as architecture, reusable software components, and domain models, where the architecture is the main element of this set. Linden at al. [33] separate the variability in three types, as follows:

- **Commonality** – It is common assets to all the products;
- **Variability** – It is common assets to some products; and
- **Specific products** – It is required for a specific member of the family (it cannot be integrated into the set of the family assets).

In the SPL context, both commonalities and variabilities are specified through features. A feature consists of a prominent or distinctive user-visible aspect, quality, or characteristic of a system [37]. SPL engineers consider features as central abstractions for the product configuration since they are used to trace requirements of a customer to the software artifacts that provide the corresponding functionality. In this sense, the features communicate commonalities and differences of the products between stakeholders, and guide structure, reuse, and variation across all phases of the software lifecycle [38, 39].

According to Kang et al.[37] and Muthig at al. [40], features can be classified as follow:

- **Mandatory feature** – It represents the common functionality that must be presented in all products of the family;
- **Optional feature** – It represents functionality that may be part of a product;
- **OR feature group** – It allows the selection of one or more features of this group; and
- **XOR (alternative) feature group** – They are mutual-exclusive functionality, *i.e.*, it belongs to a group of features from which no more than one feature must be selected.

Figure 2.2 shows the feature model of Mobile Game SPL adapted from Pascual et al.[2]. It is composed by four control systems, follows:

1. The `optional` feature **Multiplayer** is decomposed in both `alternative` features **Local** and **Online**;
2. The `optional` feature **Sound** encompasses the `optional` feature **Vibration** and the `mandatory` feature **Quality** that, in turn, is decomposed in two `alternative` features (**128kbps** and **256kbps**);
3. The `optional` feature **Connectivity** is decomposed in both `Or` features **Network**, which encompasses two `alternative` features (**EDGE** and **Wifi**) and **Bluetooth**; and
4. The `mandatory` feature **Graphics Quality** is decomposed in both `alternative` features (**Low** and **High**);

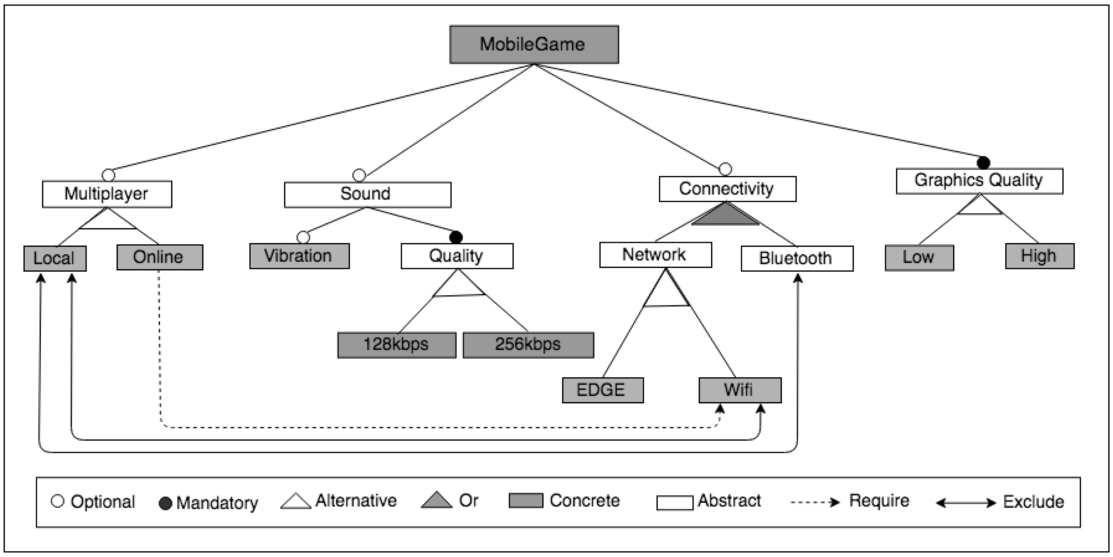
It also is possible to represent the relationships between features, which are named as constraints. Such constraints are specified as `A requires B` or `A excludes B` statements. Figure 2.2 states, for instance, that in the case the **Online** feature is selected, it is necessary to select the **Wifi** feature. In contrast, in the case the **Local** feature is selected, the **Wifi** and **Bluetooth** features must be deselected.

The variability is viewed as being the capability to change or customize a system. It allows developers to delay some design decisions, *i.e.*, the variation points. A variation point is a representation of a variability subject, for example, the type of lighting control that an application provides. A variant identifies a single option of a variation point. Using the same example, two options of lighting control can be chosen for the application (*e.g.*, the user or autonomic lighting) [32].

Although the variation points identified in the context of SPL hardly change over time, the set of variants defined as objects of the variability can be changed. This is the SPLE focus, *i.e.*, the simultaneous use of variable artifacts in different forms (variants) by different products [41].

The variability management is an activity responsible to define, represent, explore, implement, and evolve SPL variability [33] by dealing with the following questions:

Figure 2.2: Feature model of the Mobile Game SPL adapted from Pascual et al.[2]



- Identifying what varies, *i.e.*, the variable property or variable feature, which it is the subject of the variability.

- Identifying why it varies, based on the needs of the stakeholders, user, application, and so on.

- Identifying how the possible variants vary, which are objects of the variability (an instance of a product).

Another important concept in the variability of SPL is the binding time. It consists of the moment that a certain sub-process in application engineering binds variability introduced by the corresponding sub-process in domain engineering [32]. In conventional SPL, the binding time can occur at different times, such as compilation, linking, and runtime. These binding times are explained as follows:

- **Compile-time** – Select the variant before the actual program compilation or at compile-time;

- **Link-Time** – Select the variant during module or library linking (*i.e.*, the variability point is bound at link time when a compiled module is linked to the variability point); and

- **Runtime** – Select variant during program execution (*i.e.*, at any time during the use of the system, the functionality may be added, deleted, or both).

Dynamic adaptive systems (highly configurable) increase the need to deal with variability at runtime, because it modifies its internal structures dynamically, and consequently, its behavior in response to internal and external incentives [42]. For this reason, some researchers introduced the dynamic SPL approach to deal with changes in the environment and the user requests during runtime.

2.2 DYNAMIC SOFTWARE PRODUCT LINES (DSPL)

Emerging domains, such as mobile, ubiquitous computing, and software-intensive embedded systems demand a high degree of adaptability from the software. The capacity of this software to reconfigure and incorporate new functionality can provide significant competitive advantages. This new trend market requires SPL to become more evolvable and adaptable [43]. More recently, DSPL became part of this field.

The DSPL approach has emerged within the SPLE field as a promising means to develop SPL that incorporates reusable and dynamically re-configurable artifacts [21]. Thus, researchers introduced the DSPL approach enabling to bind variation points at runtime. The binding of the variation points happens initially when the software is launched to adapt to the current environment, as well as during operation to adapt to changes in the environment [21].

According to Hinchey *et al.* [6], the DSPL practices are based on *(i)* explicit representation of the configuration space and constraints that describe permissible configurations at runtime on the level of intended capabilities of the system; *(ii)* the system reconfiguration that must happen autonomously, once the intended configuration is known; and *(iii)* at the traditional SPLE practices.

2.2.1 Essential Activities and Benefits

The development of a DSPL involves two essential activities: *monitoring* the current situation for detecting events that might require adaptation and *controlling* the adaptation through the management of variation points. In that case, it is important to analyze the change impact on the product's requirements or constraints and planning for deriving a suitable adaptation to cope with new situations. In addition, these activities encompass some properties, such as *automatic decision-making, autonomy and adaptivity*, and *context-awareness* [8, 21].

The runtime variability can help to facilitate automatic decision making in systems where human intervention is extremely difficult or impossible. For this reason, the DSPL approach treats automatic decision making as an optional characteristic. The decision to change or customize a feature is sometimes left to the user [43]. However, context awareness and autonomy and adaptability are treated the same way.

The adoption of a DSPL approach is strongly based on the need of dealing with dynamic variability rather than market forces. It supports the configuration and extension of capabilities at runtime, which implies more flexible changes [6, 21]. Given these characteristics, DSPL would benefit from research in several related areas. For example, it can provide the modeling framework to understand a self-adaptive system based on Service-oriented Architecture (SOA) by highlighting the relationships among its parts, as well as, in the automotive industry where the need for post-deployment, dynamic and extensible variability increases significantly [43, 44, 45].

2.2.2 Development Cycle

The processes within the DSPL lifecycle differ from the processes of the traditional SPLE. A DSPL is not built as part of an SPL (at development time), *i.e.*, its architecture is a single system architecture that provides a basis for all possible adaptations of the system [21]. It means that the whole variability may be achieved at runtime.

Both *domain engineering* and *application engineering* processes aim at the systematic development of the system and its use by exploring the adaptability. The principles of DSPL characterize the processes as follows [6]:

- **Domain Engineering** – It is a process that identifies the possible adaptations and their triggers, besides the set of possible system reactions, focusing on construction; and

- **Application Engineering** – It handles the variations in such a way that the system itself performs the reconfiguration.

The DSPL must be able to consult the dynamic variability model defined in the domain of engineering in order to identify possible adaptations. Accordingly, the reference architecture must support the variations described by the variability model and provide support for the entire range of adaptations handled during application engineering. The reference architecture is a core asset that takes the entire development process into account since it provides a basis to check the validity of configurations according to context conditions [46].

2.2.3 Dynamic Variability

Dynamic variability occurs due to product variations that appears in the execution environment or the product itself. These variations depend on the context variations which are computationally accessible information extracted by monitoring the execution environment or the current state of the product. In this case, an adaptation to a given context corresponds to a product configuration of the DSPL [8].

Systems in DSPL should be prepared to identify contexts unknown at design time. After that, systems must be prepared to add dynamically new features to meet new requirements or simply to improve the current state of the system when new features become available [47]. The simple transition from an SPL where the variability is bounded at development time to a system that adapts its behavior by binding variability at runtime has several consequences. Variability is no longer simply an engineering artifact, in DSPL the variability model is the core artifact to guide the system adaptation. The DSPL should be able to query the runtime variability model to identify adaptations [6].

Bencomo *et al.* [47] associated the need to support unanticipated adaptation in dynamically adaptive systems to two types of variability: *(i)* environment or context variability to represent the conditions and events that can modify the current architecture of the system and *(ii)* to specify the architecture of the system which will evolve at runtime. In order to satisfy the requirements for the new context, the system may add new features or organize the current configuration.

Additionally, it is possible to divide dynamic adaptation into two different types: *(i)* dynamic behavior, the systems deal with new environmental conditions unknown during development, and *(ii)* dynamic reconfiguration, it is necessary that variations of behavior be predetermined before execution. During the execution, the current state of the system is evaluated and the appropriate variants are chosen [47].

Implications for Dynamic Reconfiguration – According to Bosch e Capilla[43], the following factors often imply that earlier versions of systems components can be replaced by new versions:

- Variation points change during the system operation (*i.e.*, they become increasingly dynamic);
- The set of variants for a variation point can be extended after system deployment and while the system is operating;
- Systems increasingly select the variation as they seek to maintain or achieve a certain adaptation.

Therefore, the products generated from DSPL provide new variants and different configurations dynamically. System features can trigger reconfigurations at runtime when needed, and these features can adapt system behavior to different scenarios. These reconfigurations consist of activating, deactivating, and updating of the system's features [48].

Open variability – The evolution capability has so far not been investigated in detail in the context of DSPL. However, it can be addressed aiming to investigate its impact. Emerging domains require changes and extensions to the design in terms of both functionality and adaptation capabilities. DSPL should deal also with the evolution of user needs and execution environments in ways not foreseen at the time of initial design [8].

The DSPL approach can make easier the modification of the system implementation when it is needed to change the initial configuration space. Thus, it is possible to integrate further adaptation capabilities by exploiting the variability model due to the *open variability*, since it consists of extending the system with new variations at runtime [8]. Such changes can be the addition, removal, or modification of products or transitions among products. The time when a mechanism is open for adding new variants is mainly decided by the development and runtime environments, and the type of feature that is represented by the variation point [41].

2.3 CONNECTION BETWEEN SPL AND DSPL FIELDS

Although the central concept of DSPL is a traditional SPL, there are differences between both approaches. SPL and DSPL are compared based on variability goals, binding time, the stakeholders who decide the variability, models that define the variation points or control the adaptations, and mechanisms to implement the variation points. The DSPL engineering is, in turn, an emerging research field that must deal with variability modeling and implementation by handling context information and quality aspects, as well as, unanticipated changes [8]. In this sense, this section also describes the prospect of changes in terms of improvement of software artifacts by comparing both study fields.

2.3.1 Differences and similarities between SPL and DSPL

A product derived from DSPL differs from other instantiated from SPL by the capacity to adapt through the binding of variation points at runtime. In order to manage the variability in DSPL, it is necessary to create reconfiguration rules, *i.e.*, to adopt different adaptation policies [49]. A modification, when detected at the operational context, actives the product reconfiguration to provide context-relevant services or collect quality requests (including safety, reliability, and performance) [50].

DSPL is similar to traditional SPL regarding reconfiguration focused on functional capabilities. However, the variability is not simply an engineering artifact that is present before runtime. DSPL applications must be able to consult the variability model to identify adaptation because this model is the core artifact for guiding system adaptation [6]. Table 2.1 shows the relationships between both approaches.

According to Hinchey *et al.* [6], the difference between SPL and DSPL is based on a set of innovations, including: *(i)* variability modeling, which describes the differences among the systems; *(ii)* reference architecture that supports the variations; *(iii)* business scoping that encompasses the understanding of entire domain or business field, and *(iv)* both life cycles, such as *domain engineering* and *application engineering*.

Additionally, a product obtained from a traditional SPL to a specific configuration can be quite tested before reaching the hands of the user [51]. However, a product generated from DSPL provides new variants and different configurations, which are obtained at runtime. In this way, any flaw in the reconfiguration of DSPL directly impacts the user experience, because it happens when the system is already under its control [52]. Thus, some studies have been performed to investigate the aspects of the reconfiguration of product lines at runtime aiming to deliver high-quality software [52, 53, 54, 55, 56].

The removing and changing of product features are some examples of reconfiguration for SPL. These features are bound statically, where the user selects the desired features. Then, a generator creates the corresponding software product containing exactly the necessary features [38]. In the DSPL engineering, however, the product dynamic reconfiguration at runtime refers to remove and change the features developed, *i.e.*, it updates dynamically the system configuration after deployment [48].

DSPL does not deal with an entire SPL in the traditional sense. It is considered as a single system that adapts its behavior when the variability is rebound during operation [57]. The dynamic reconfiguration approach uses a mapping of DSPL features for components, *i.e.*, it allows the developer to specify adaptation rules for reconfiguration components [58]. Thus, the selection of mechanisms that support the runtime decisions consists of important activity. It must enable to implement the variation points to adapt the applications according to the reconfigurable artifacts defined at design time.

Table 2.1: Adapted from Hinchey at al. [6]

Software products lines	Dynamic software products lines
Variability management describes different possible systems.	Variability management describes different adaptations of the system.
The reference architecture provides a common framework for a set of individual product architectures.	DSPL architecture is a single system architecture, which provides a basis for all possible adaptations of the systems.
Business scoping identifies the common market for the set of products.	Adaptability scoping identifies the range of adaptation the DSPL supports.
Two life cycle approach describe two engineering life cycles, one for domain engineering and one for application engineering.	The DSPL engineering life cycles aims at the systematic development of the adaptive system, and the usage life cycle exploits adaptability in use.

2.3.2 Open issues in DSPL engineering

The change can be initiated in order to correct, improve, or extend assets or products. Many of the practices of a successful SPL initiate, manage, or consume these changes. Both conceptual techniques and software tools are available to assist in the management of these changes [59]. Evolution has been widely studied in SPL engineering. Most existing work has focused on the evolution of *problem space* [60]. However, evolving a variability model may also affect the *solution space* and vice versa. The problem space refers to the system's specifications established during the domain analysis and requirements engineering phases, whereas the solution space refers to the related assets created during the architecture, design, and implementation phases [61].

In contrast, existing research only recently started to investigate evolution in DSPL engineering, and especially its impact on the running system. Evolving DSPL poses significant challenges as both problem and solution spaces. The evolution of problem or solution spaces can lead to inconsistencies within the given space, between spaces, and with respect to rules for the runtime adaptation of the system [62]. Variability models that are used in a DSPL have to co-evolve and be kept consistent with the systems they represent to support reconfiguration even after changes to the systems at runtime [62]. However, there is limited work to support co-evolution. For instance, a set of the evolution of the problem space and the solution space, as well as remapping operators to avoid inconsistencies between the two spaces, is described by Seidl *et al.* [63].

Helleboogh et al.[64] proposed the notion of super-types to describe the evolution of variability models at runtime. Capilla et al.[4] use super-types to automate the modification of variants in a feature model at runtime. However, these approaches are limited to a given set of changes *e.g.*, the addition of a variant, like other kinds of changes, cannot be automated [65].

Talib et al.[66] presents a classification of required operations for the jointly evolving problem and solution space in a DSPL. However, they use the general term variability model to describe any model of the variability of a software system. In addition, they analyzed the impact of evolution operations on the consistency of the DSPL and architecture of a tool-supported approach that addresses some issues and supports the evolution of DSPL. They presented some requirements for DSPL and categorized them in terms of dynamic reconfiguration and evolution. However, these requirements do not take into consideration quality aspects (non-functional requirements).

According to Hinchey at al. [6] there are many open issues in DSPL engineering including the following: *(i)* a few support exists for DSPL evolution; *(ii)* enlarging the existing approaches that focus on modeling of configuration options to capture context description and decision making; *(iii)* DSPL reconfiguration has focused mainly on functional capabilities while addressing quality characteristics to only a limited extent; *(iv)* dealing with overly constrained situations at runtime remains a concern and; *(v)* how to better extend variability modeling so that developers could use it as a basis to interpret contexts and support autonomous systems.

We evidenced such issues in our previous studies [26, 27]. Thus, it is important to investigate them in more detail, especially that DSPL applications should deal with evolution in terms of functionality and adaptation capabilities when the demand arises for new requirements in existing products or new configurations. Based on that, we decided to investigate DSPL engineering from a variability-modeling perspective and propose a decision support approach to help the software engineers on the accurate representation of the impact of features over NFRs and contexts for the identification of feasible configurations.

2.4 REQUIREMENTS ENGINEERING

NFRs are considered as imposed constraints on the software or qualities that the software product must-have. According to these two perspectives, NFRs can be identified by considering several elements such as constraints, concerns, goals, and quality attributes. Their relevance degree can also vary depending on the different types of software or application domains.

Though NFRs are as important as functional requirements, they are neglected, poorly understood, and not adequately considered in the development of single systems and traditional product lines [67]. The same happens in the DSPL engineering field since just a few studies address product configuration issues considering NFRs information [14, 15]. Poor requirements elicitation results in many failures in software systems [68]. For this reason, it is essential to understand how a system must behave aiming its dependability.

The Goal-oriented Requirement Engineering (GORE) provides means to support the requirements elicitation and decompose them into well-defined entities and reason about the alternatives to meet them [69]. In this way, goal-oriented modeling has been recognized as a suitable strategy to represent the objectives of a software system and stakeholder's preferences [70]. Goal models may then be used as part of the entire system life-cycle, in particular, to improve the continuous delivery of software systems [71].

Goal models fit well with the early domain engineering phases by supporting different alternatives for satisfying stakeholder's preferences [72, 73]. They are composed by *goals*, *hard goals*, and *soft goals* depending on the precision of their satisfaction level in terms of stakeholder's intentions. *Goals* consist of the objectives of the stakeholders. *Hard goals* represent the system's features and have satisfaction level clearly defined, whereas *soft goals* express NFRs and have a subjective satisfaction level that cannot be defined in a clear-cut way and cannot be fully evaluated [74]. Thus, *hard goals* and *soft goals* may be judged as satisfied or unsatisfied to different degrees and at different stages of the development of DSPL applications [72].

Additionally, the GORE approach supports the conceptual modeling of variable and common requirements of a DSPL application by representing them as goals [72]. Such *goals* consist of objectives or desirable states for software systems and are decomposed through AND/OR relationships with *hard goals*. In the AND decomposition all *hard goals* should satisfy its parent *goal*, whereas, in the OR decomposition, at least one *hard goal* satisfy its parent *goal* [73] (see Figure 4.3 in Chapter 4).

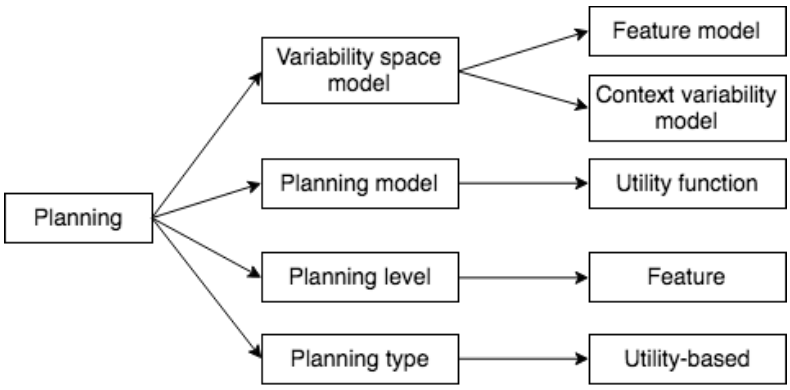
2.5 PLANNING THE DEVELOPMENT OF DSPL PROJECTS

Bashari *et al.* [3] proposed a framework to classify different dimensions of adaptation realization in DSPL projects. The framework dimensions are organized in a taxonomy that uses the steps of the `MAPE-K` loop. Hinchey e Schmid[6] proposed the development of DSPL applications based on the `MAPE-K` loop. It defines how systems adapt their behavior to keep their goals controlled based on control systems or optimization strategies. The `MAPE-K` loop is composed of four activities: `Monitor`, `Analyze`, `Plan`, and `Execute`, including the base of `Knowledge` to support model representations. Thus, it supports the autonomic condition of DSPL by mapping contexts to variants and offering them a better selection of the possible alternatives and according to goals to be accomplished [4].

In this work, we focus on the planning step (see Figure 2.3) to define a *planning model* based on the *utility function*. It aims to support the software engineers in decision-making during the modeling initial phase of DSPL projects. The DSPL community has devoted most efforts to develop planning approaches [8]. Hallsteinsen et al.[9], for instance, defined a utility-based approach to represent desirability of the current configuration in the current context. In this sense, the goal of adaptation is the maximization of the value of this function by considering the system properties and its context. In cases when the utility value is unacceptable, the planner uses a brute-force technique to find a configuration with the highest predicted utility value and adapts to it [3].

In general, a planner uses information provided in the context model to decide whether an adaptation is necessary and find, within the *variability space model*, the variant that satisfies such an adaptation [3]. Each desired variant is represented by a system's feature that, in turn, is specified as a *planning level*. Therefore, the planner decides what features should be active in the system according to its context.

Figure 2.3: Dimensions related planning step adapted from Bashari, Bagheri e Du[3]



2.5.1 Variability Modeling

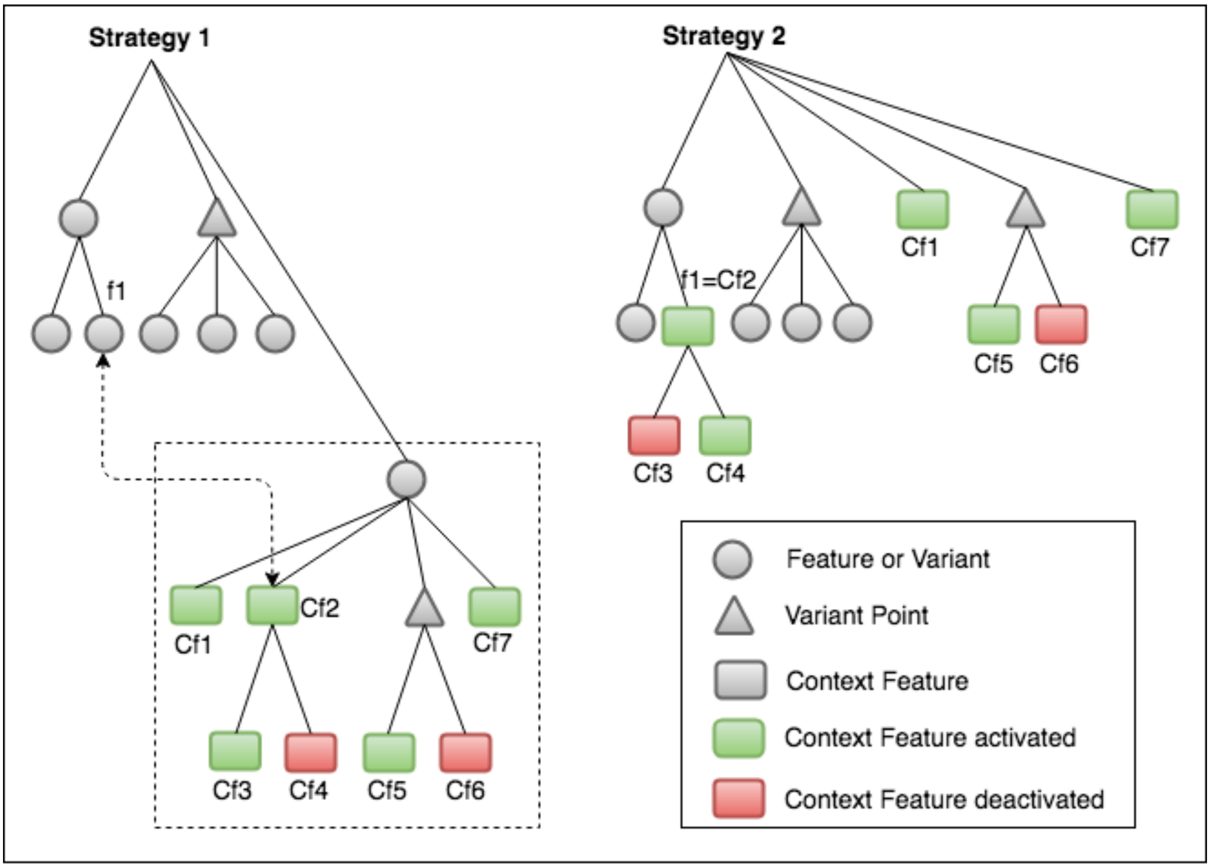
In DSPL engineering, feature models are commonly used as a variability model due to their power to represent complex variability of software systems. Nevertheless, a context variability model also must be defined in this *variability space modeling* since it represents the contextual triggers for variation [3]. Context variability deals with the diversity of the contextual changes that influence the dynamic behavior of systems [4].

Hartmann e Trew[11] introduced the notion of context variability to identify context features and capture the common and variable information of contexts during the initial modeling phase. In DSPL engineering, the variability modeling task handles both the system's features and context features as well as dependencies between them. The software engineers, in turn, use the feature model to manage system variability. It consists of a model language widely employed to represent the system's features in a hierarchical structure [75]. However, such language was extended with purpose also to model the context variability.

Figure 2.4 shows two strategies to model context variability. Both strategies are linked to the feature model. In the first strategy, the feature model includes a branch in which the software engineer models context features separately from the system's features. It is more reusable when there are several context features to be represented. Whereas in the second strategy, the context and system's features are represented under the same model. It reduces the number of dependencies between context and system's features and simplifies the model [4].

Additionally, the modeling of Non-functional Requirements (NFRs) is also an important activity in DSPL engineering, once makes it possible to identify interdependence constraints between NFRs and features to a specific context adaptation scenario [76]. However, feature models do not capture NFRs explicitly neither influence these properties to achieve alternative configurations of a product variant. Indeed, it rather complexes to represent NFRs in DSPL models, since software engineers need to consider, for instance, configuration rules, features constraints, stakeholder's preferences given a particular context [77]. For this reason, we decided to model contexts and NFRs information in an independent way by using both, goal model and eCFM at the modeling step of our

Figure 2.4: Strategies for context variability modeling adapted from Capilla et al.[4]



approach.

2.5.2 Planning strategies

Planning type is related to the strategies that are employed for selecting the most suitable variant of the system. According to Huebscher at al. [31], the *planning types* can be categorized into three groups, as follows:

- *Rule-based* planning: This planning strategy employs ECA rules (Event Condition-Action) or state-transition diagrams to define adaptations at design time. Some *rule-based* planners allow the modification of the rules at runtime [78]. However, the usage of this planning requires not only the enumeration of possible system reconfigurations but also the thorough knowledge of the operating the environment at design time. Indeed, *rule-based* planning requires that the software engineer in-depth expertise of the operating environment [8].

- *Goal-based* planning: Using this planning, the possible adaptations that can occur to meet contextual changes are figured out at runtime, unlike *rule-based* planning where such actions are specified at design time. Therefore, the high-level goals of the adaptation are formally defined and figured out by the planner. Afterward, the problem of detecting the most suitable action is reduced to a satisfiability problem (SAT) [38, 79] or a constraint satisfaction problem (CSP) [80, 81, 82], which is then solved using an SAT-solver or a CSP-solver. In this way, the goals of the adaptation are expressed by constraints *over* system quality; and

- *Utility-based* planning: In this planning strategy, the problem of feature selection can be handled by means of optimization methods in order to find the feasible configuration for that feature model [2, 83]. For this, it is necessary to predict the utility function(s) to approximate the fulfillment of stakeholder's preferences in different situations. This prediction function is used to find the configuration that has the highest predicted utility value. Such utility consists of a quantitative value to represent the weight of the system's feature [84].

We investigated the *state-of-the-art* in the search of a planning strategy that supports the software engineers in the quantitative representation of the impact of features over contexts and NFRs in order to identify feasible configurations. The feasible configuration can be found using a heuristic, which approximates the impact of features over those elements in the utility value. Therefore, we noticed that *utility-based* planning, in conjunction with an optimization method, comprises a suitable strategy to achieve such an objective.

2.6 OPTIMIZATION

Optimization consists of a principle underlying the analysis of complex decision problems. It involves the selection of values for a number of interrelated variables, by focusing on an objective designed to quantify performance and measure the quality of the decision. Depending on the formulation, such an objective is maximized (or minimized) subject to the constraints that may limit the selection of decision variable values [85]. In general, an objective function consists of a way of assigning value to a possible solution that reflects its quality on a scale. Conversely, a constraint represents a binary assessment reporting whether or not a given requirement contains solutions to a problem in terms of optimization [86].

Optimization, then, should be regarded as a tool of conceptualization and analysis. The problem formulation always involves a trade-off between the conflicting objectives of *(i)* building a mathematical model sufficiently complex to accurately capture the problem description and *(ii)* building a model that is tractable. Thus, skill in modeling, to capture the essential elements of a problem, and good judgment in the interpretation of results are required to obtain meaningful conclusions [85].

We investigated the *state-of-the-art* in the search of an optimization method that can be applied in the feasible configuration selection process. We noticed that the Integer Linear Programming (ILP) is one of the most popular modeling technique to studies based on simulations. However, the research community has diverted its attention to soft computing techniques such as Genetic Algorithm (GA) [87]. We describe such a methods, as follows:

- **ILP** is a method for optimization problems that can be applied in a large number of applications such as, in feature selection by helping application engineers in the product configuration activity [88], in an airline wishes to schedule its flight crews, and in an oil company that wants to decide where to drill for oil. This means that, if we can specify the objective as a linear function of certain variables and constraints on resources as equalities or inequalities on those variables, then we have an ILP problem [89].

  ILP solves a series of linear equations to satisfy the conditions of the problem while optimizing an objective function. The problem is mathematically designed to find a set of non-negative integer variables, denoted by $x = \{x_1, x_2, ..., x_n\}$, to maximize or minimize a linear objective function of x, denoted by $f(x) = f(x_1, x_2, ..., x_n)$, subject to a set of linear constraints of x, denoted by $c(x) = \{c_1(x_1, x_2, ..., x_n), c_2(x_1, x_2, ..., x_n), ..., c_m(x_1, x_2, ..., x_n)\}$. Any setting of those variables x that satisfies all the constraints is named as a feasible solution to the ILP [85, 89, 90].

- **GA** is applied to optimization problems in many fields, such as machine learning, traveling salesman problems, pattern recognition, and so on [91, 92, 93]. It is a random search algorithm that provides a robust searching method for the optimum solution to complex problems and uses a process similar to biological evolution to improve upon them [94].

  In a GA, the problem is represented by a population of bit strings, usually referred to as chromosomes. Thus, each chromosome (string) comprises a number of genes (bits), which represent the individual decision variables of the problem. The variables represented in the chromosome can be processed in an evaluation function or fitness function, which is in effect the objective function in order to generate new chromosomes that contain the best characteristics of two-parent chromosomes. As with the process of natural selection, chromosomes with the highest fitness have the greatest chance of contributing to future generations [92, 94].

  First, an initial population of chromosomes is generated. All chromosomes are then evaluated for fitness and good ones are selected as parents. There are three fundamental operators involved in manipulating these chromosomes and moving to a new generation: *(i)* the **selection** assures that chromosomes are copied to the next generation with a probability associated with their fitness values; *(ii)* **crossover** is an extremely important part of a GA. Such an operator takes two randomly selected chromosomes and exchanges their genes to generate two new chromosomes; and *(iii)* the **mutation** permits new genetic material to be inserted into a population, *i.e.*, it randomly selects a position in the chromosome and changes the corresponding value according to a given probability. In short, new chromosomes are created by inheritance and with variation [95, 96, 97].

Initially, we decided to use ILP as an optimization method since our problem can be represented by means of a linear function and linear constraints. However, we understood that is possible to obtain feasible solutions in an aspect of the system by meeting multi-objectives. Thus, we exchanged the optimization method by applying a GA. GAs are numeric optimization methods widely used to solve multiobjective optimization problems [126]. As pointed out in Goldberg *et al.* [91], a GA searches for feasible solutions from a population of decision variable sets, not a single decision variable set, such as an ILP problem. It means that a GA typically uses a coding of the decision variable set, not the decision variable itself.

2.7 CHAPTER SUMMARY

This chapter presented an overview of SPL beyond basic concepts related to commonality and variability. It included the motivations and benefits of applying SPLE practices in software development. These practices have focus on the detailed design and the implementation of reusable software assets. In this sense, variability management activity is considered essential for the entire SPL life cycle.

This chapter also presented an overview of DSPL engineering beyond basic concepts related to runtime variability. It included the motivations and benefits of applying such an approach in software development that requires increasing adaptation in runtime. It also presented the basis for understanding the connection between both, SPL and DSPL fields discussing its similarities and differences, beyond open issues in DSPL engineering. Lastly, this chapter described how to plan the development of DSPL projects by considering key aspects as variability modeling of features and contexts, requirement elicitation, and optimization.

Next chapter presents the extended Context-aware Feature Modeling (eCFM) technique to model *context variability* in DSPL engineering. In addition, it describes the empirical study performed for gathering evidence regarding the expressiveness and ease of use of the eCFM technique.

PART III

APPROACH AND EMPIRICAL STUDIES

Chapter

3

Don't wait for inspiration.—John C. Maxwell

EXTENDED CONTEXT-AWARE FEATURE MODELING (ECFM) TECHNIQUE

Dynamically Adaptable Software (DAS) can be considered as a Dynamic Software Product Line (DSPL), in which variabilities are bound at runtime. Thus, practitioners have used DSPL engineering to support DAS development. This has helped DAS developers to cope with context variability and the large number of configurations [98]. Among the DSPL engineering activities, the variability modeling activity is one of the most important, since it guides the software engineer to handle the diversity of contexts that influence the system's dynamic adaptations [4]. Indeed, there are different approaches supporting DSPL variability modeling [64, 72, 99].

To help in the choice of one approach, in a previous study Souza et al.[22] presented a ranking of DSPL Variability Modeling Techniques (VMTs). Moreover, some techniques were empirically evaluated with a controlled experiment [23]. The results of this experiment showed evidence that Context-aware Feature Modeling (CFM) [99] is the most effective VMT to model DAS variability. However, it does not allow to define constraints among contexts and its limited context representation makes understanding models more difficult, particularly large ones. In this sense, we go beyond the efforts of Souza et al.[22] and Souza et al.[23] and present a systematic way for DAS modeling, by providing an extended version of the CFM technique (eCFM). The remainder of this chapter is organized, as follows:

Section 3.1 presents an overview about Variability Modeling Techniques;

Section 3.2 describes the steps that a software engineer should perform to model DAS projects with the eCFM technique;

Section 3.3 reports the analysis and the interpretation of the results of the survey and threats to validity; and

Section 3.4 draws concluding remarks and points out future directions.

3.1 OVERVIEW ABOUT VARIABILITY MODELING TECHNIQUES (VMTS)

This section discusses existing VMTs and the evidence that indicates Context-aware Feature Modeling (CFM) as one of the most suitable techniques to model Dynamically Adaptable Software (DAS) variability. Souza, Santos e Almeida[22] revised the DSPL literature to find the main VMTs. Then, they performed a ranking of these approaches and identified ten VMTs. Next, they evaluated these techniques according to a set of twelve criteria based on DSPL properties [4, 100]. For instance, the assessment addressed whether the technique supports the modeling of *context information*, *(de)activation of features*, and the *adaptation triggers*.

The existing VMTs have used different ways to support context variability modeling. For example, one of them extends the *Tropos Goals Model* by adding context requirements (TGMC) [72]; another uses a *Meta-Variability Model* that introduces the meta variability concept [64]; also, there are some VMTs extending the traditional feature model with a context model, such as the CFM technique [99]. In a CFM model, each context information can be related to a system feature through a require or exclude dependency, establishing the adaptation rules.

Throughout their study, the authors identified the CFM and TGMC techniques as the best ones in the ranking results. Both techniques are similar with regard to dynamic elements that they can model, such as context information and the adaptation rules defined by the relationship between contexts and system features. Moreover, they evaluated both techniques by means of a controlled experiment [23]. The results indicated that participants were more effective in modeling DAS using CFM than TGMC.

Nevertheless, we identified that CFM presented a limited expressiveness to model the context variability. It does not allow defining constraints among contexts and this is an important property since in the real environment there are contexts that cannot occur at the same time. For example, the constraint among the times of day (*e.g.*, *day*, and *night*) perceived by a smart home application cannot be explicitly modeled in the CMF technique.

3.2 STEPS TO MODEL DAS PROJECTS WITH THE ECFM TECHNIQUE

An expressive and correct model for developing DAS is a first-class concern. In this section, we present the steps to model DAS by using the eCFM technique. Such steps are detailed in Figure 3.1, which shows a running example in the smart home domain. A Smart Home is a kind of DAS that by means of sensors identifies context changes in the house and can (de)activate its features related to home automation (*e.g.*, temperature control).

To use the eCFM, the domain/software engineer must perform the *context analysis* task [4] during the DAS domain analysis, identifying context-aware properties and how they affect the system configuration. A literature review and an analysis of existing applications can support this task.

In particular, stakeholders should identify four main elements for specifying DAS in eCFM: *(i)* **functional requirements** represented as system features; *(ii)* **variability**

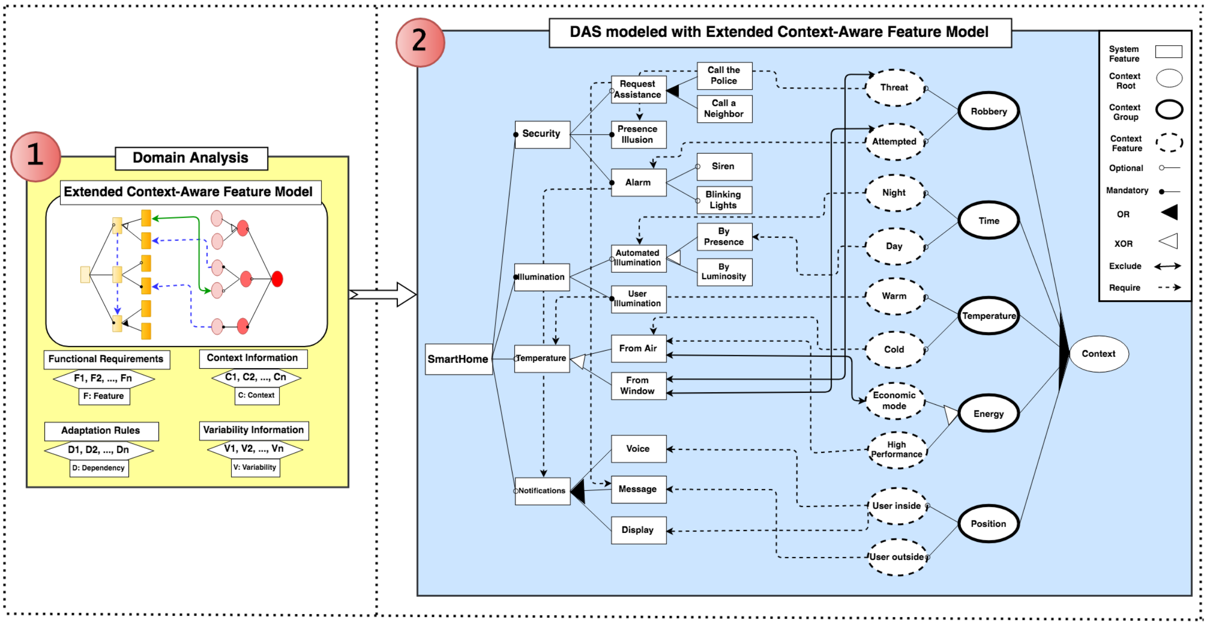


Figure 3.1: Approach for DAS modeling

information specified as feature relationships of the type `mandatory`, `optional`, `OR`-group or `XOR`-group; *(iii)* **context information**, which are described as context states; and *(iv)* **adaptations rules**, which are specified as dependencies between features and contexts.

The eCFM technique follows the main concepts of the CFM approach as shown in Figure 3.2(a) and Figure 3.2(b). However, we improved the model expressiveness with other types of context variability, such as `alternative` and `optional` group of context features. The main concepts used in eCFM are represented, as follows:

- **Context Feature** – a relevant context state to the software, which can be active or inactive, *i.e.*, the context can be occurring or not in the surrounding environment;

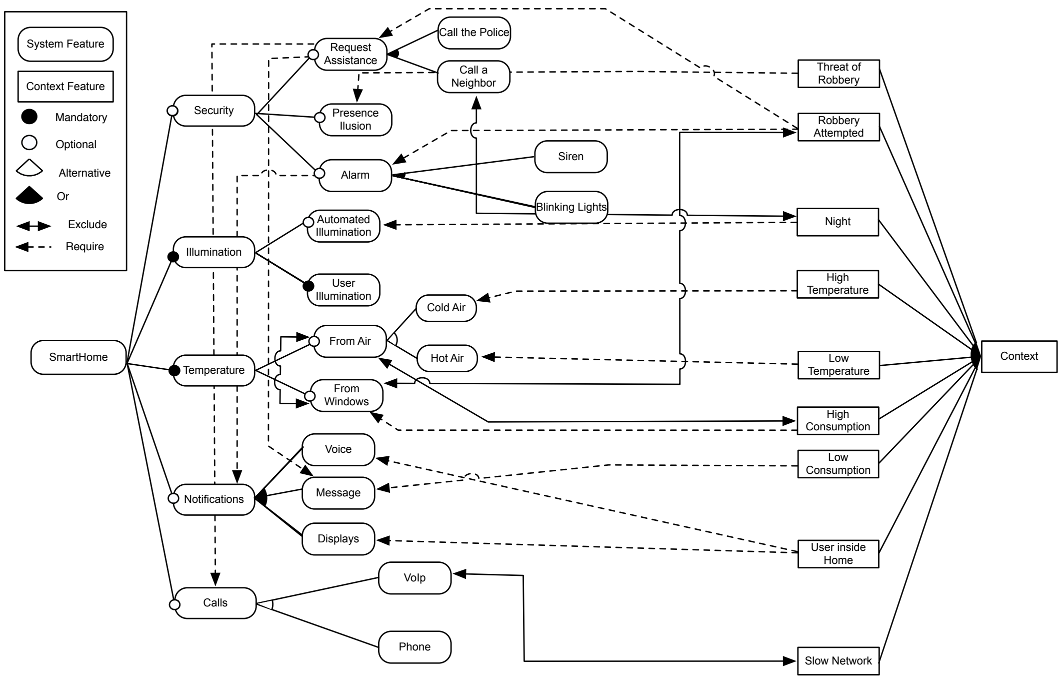
- **Context Group** – a set of context features that can specify `XOR`-group/`OR`-group relationships or `optional` context features. For example, the `XOR`-relationship between **Economic Mode** and **High Performance** in Figure 3.1 (step 2). It can also encompass one or more `optional` context features, for instance, the context group **Position** is composed by both context features, **User inside** and **User outside**; and

- **Context Root** – the root of the context model that aggregates the context groups in an `OR`-relationship.

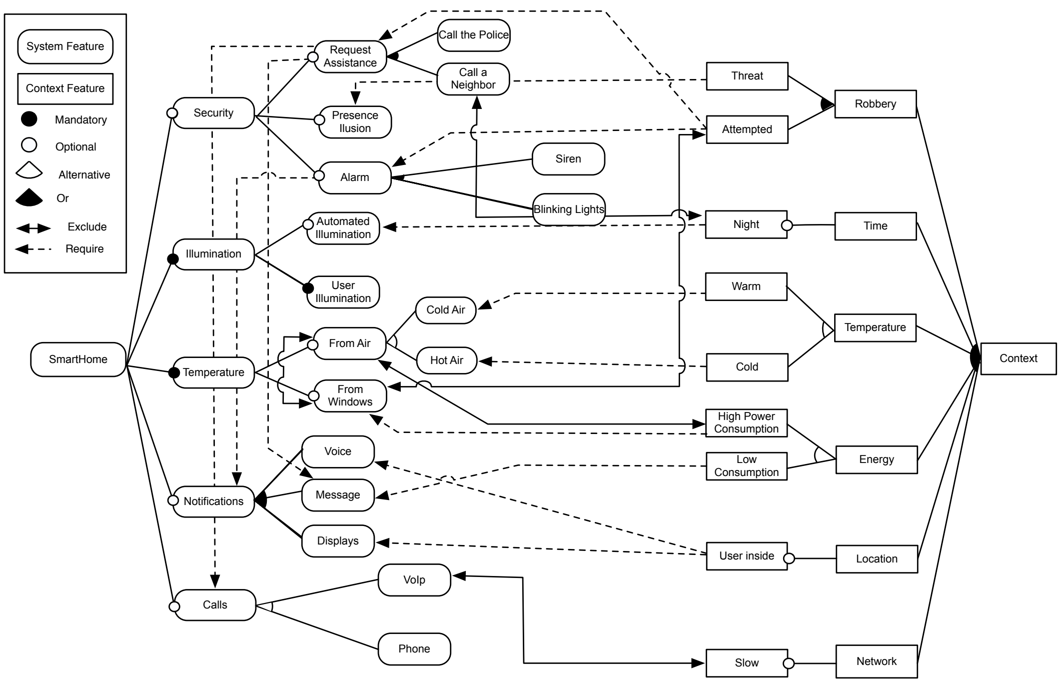
Some of the main benefits of the eCFM are: *(i)* it allows to model constraints among context features increasing the context variability expressiveness; and *(ii)* the group concept allows to organize the context features in different categories defined by the software

Figure 3.2: (a) Smart home modeled with CFM (b) Smart home modeled with eCFM

(a)



(b)



engineer (*e.g.*, based on purposes or context source) favoring the model organization and comprehensibility.

For instance, in Figure 3.1, one of the functional requirements of the smart home (step 2) is represented by the system's feature **Temperature** that controls the environment temperature. This feature has two variations with an `OR`-group: **FromAir** and **FromWindow**. The **FromAir** feature has an `exclude` dependency relationship with the context feature **EconomicMode**. So, this `exclude` dependency relationship specifies an adaptation rule that is triggered by the context feature **EconomicMode**.

In general, the software engineer must understand the context, what it is, and how it behaves. Then, during the eCFM modeling, the engineer can reason over how such contexts affect the DAS. This information is available from the domain analysis. In this way, we recommend some tips to software engineers that intend to model DAS with eCFM technique, as follows:

- **Use suggestive context and feature names.** This is important to ensure extensibility and maintenance of the model;

- **Avoid the use of exclude-dependency.** We recommend preferentially the use of `require`-dependency since it makes the model more intuitive; and

- **Define context groups according to the context source.** To name context groups is interesting to consider the notion of a context source (*e.g.*, sensors, middleware). For a large model with several context features, consider using groups to organize hierarchically the context features.

- **Avoid `require` and `exclude` relationships between parent features and context features.** We recommend the software engineers avoid to model and develop DSPLs with such adaptation rules in order to prevent conflicts at runtime.

3.3 EVALUATION OF THE ECFM TECHNIQUE

This section presents the design and planning of the survey performed to evaluate the CFM and eCFM techniques. In addition, it describes the analysis and the interpretation of the results and threats to validity.

The survey was designed by two Ph.D. students from the Federal University of Bahia and one Ph.D. student from the Federal University of Ceara. Such students were responsible for defining *(i)* the background and feedback forms, *(ii)* the presentation content; and *(iii)* the guideline describing instructions on how to model DAS using the VTMs under evaluation, besides the tasks to be performed for the training of the subjects. Then, one of those students proceeded with the survey execution in an academic environment (software engineering lab) with fifteen Computer Science students from the Federal University of Bahia (six Ph.D. and nine M.Sc. candidates).

3.3.1 The Survey Definition

We conducted a survey to evaluate the comprehensibility of contextual variability modeling by comparing the CFM and eCFM techniques. We asked the subjects to perform tasks of mapping, reading, and modifying elements of two DSPL projects: a smart home system and a mobile application. With the execution of such tasks, it is possible to obtain in the feedback forms, more concrete answers, and with a technical foundation.

3.3.1.1 Objective .

The study aimed at **analyzing** the CFM and eCFM techniques **for the purpose of** comparing them **with respect to** *comprehensibility* and *easiness* of use **from the point of view of** Software Engineers and researchers **in the context** of two variability models in the smart home and mobile domain. Table 3.1 summarizes the design for the empirical assessment addressed to DSPL modeling.

Table 3.1: Study Design

Goal	- Evaluating the eCFM and CFM for the purpose of comparing them with regard to its comprehensibility and easiness of use
Context	- Two variability models in the smart home and mobile domain tasks - Overall 15 graduate students as subjects
Independent Variables	- VMTs used (Treatments: eCFM and CFM) - Application domain (Smart Home and Mobile Cases) - Order of treatments
Dependent Variables	- The strengths and weaknesses of each VMT

3.3.1.2 Research Questions (RQs) .

To achieve the work goal, we identified two main research questions, as follows:

RQ1. **Which is the most expressive variability modeling technique to deal with DAS projects?** This research question analyzes which of the two VMTs is the most expressive concerning the representation of adaptation rules between the system's features and contexts.

RQ2. **Which is the easiest variability modeling technique to model DAS projects?** This research question analyzes which of the two VMTs is the easiest to understand the modeling, model the DAS elements, include new elements, and model complex scenarios of DAS projects.

3.3.2 The survey Planning

This section discusses the planning and the procedures to be followed in order to perform the survey (see artifacts of the survey in A). The empirical study was organized in the following steps:

- Firstly, we requested that all subjects answer a *background form* to characterize them according to their experience and expertise;
- Secondly, we performed the training of participants with the use of presentation slides, containing concepts related to the survey. In addition, we used hands-on exercises on the basis of a small project;
- Thirdly, the subjects were randomly divided into two groups as shown in Table 3.2. One group started using the eCFM technique and then the CFM technique, whereas another group did the opposite;
- Next, we provided a guideline with instructions on how to model DAS using both techniques CFM and eCFM and a description of the tasks to be carried out by the participants;
- After the execution of the tasks, we applied a feedback form for participants to report strengths and weaknesses of each VMT, besides of difficulties and problems found during the empirical study;
- We then collected the data from background and feedback forms and task answers executed by the participants; and
- Finally, we performed an analysis of the results.

Table 3.2: The survey design

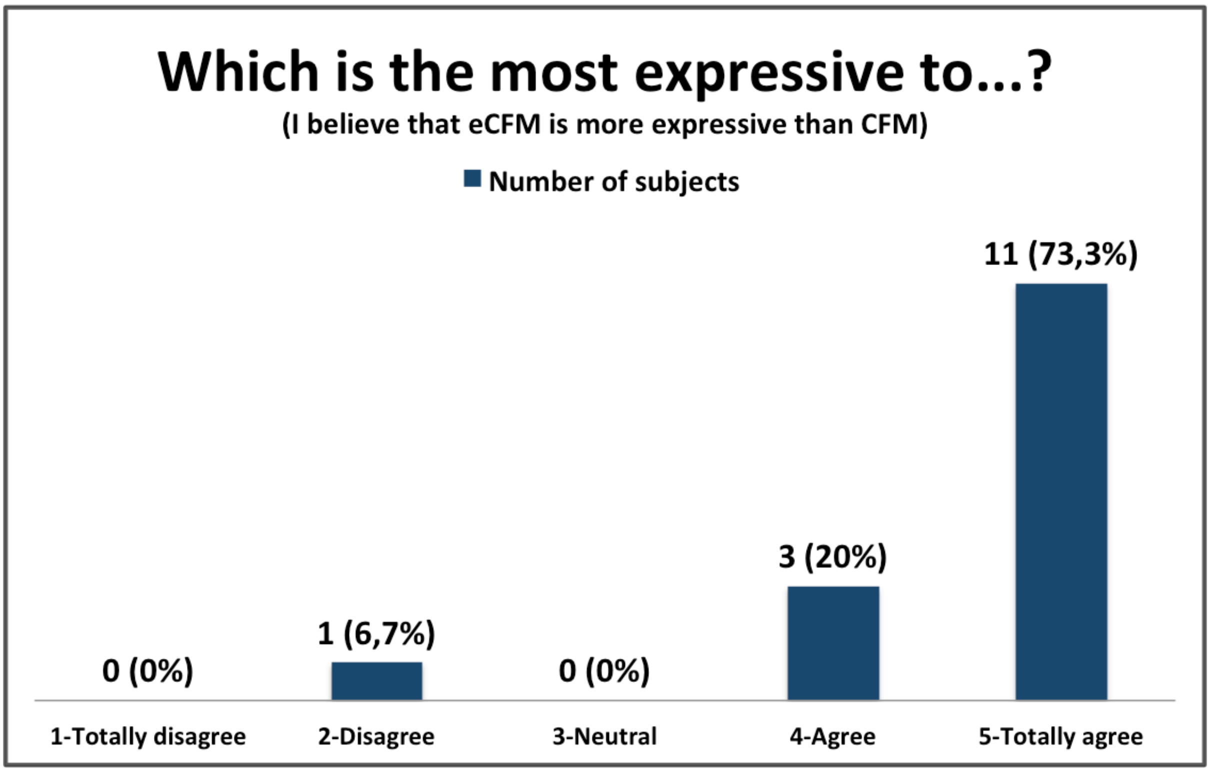
Group	**Task A**	**Task B**
1 (7 subjects)	Smart home, eCFM	Mobile, CFM
2 (8 subjects)	Smart home, CFM	Mobile, eCFM

3.3.3 Analysis and Interpretation

We conducted a survey aiming to compare the CFM and eCFM techniques according to their expressiveness and easiness of use. Inspired by Hadar et al.[101], we asked the participants to perform tasks of mapping, reading, and modifying some modeling elements of two DAS projects of the smart home and mobile application domains. Following we provide an in-depth analysis of the gathered data.

Fifteen graduate students (six Ph.D. and nine M.Sc. candidates) performed the study tasks at the Federal University of Bahia, Brazil. These students have a strong knowledge

Figure 3.3: The survey results for RQ1



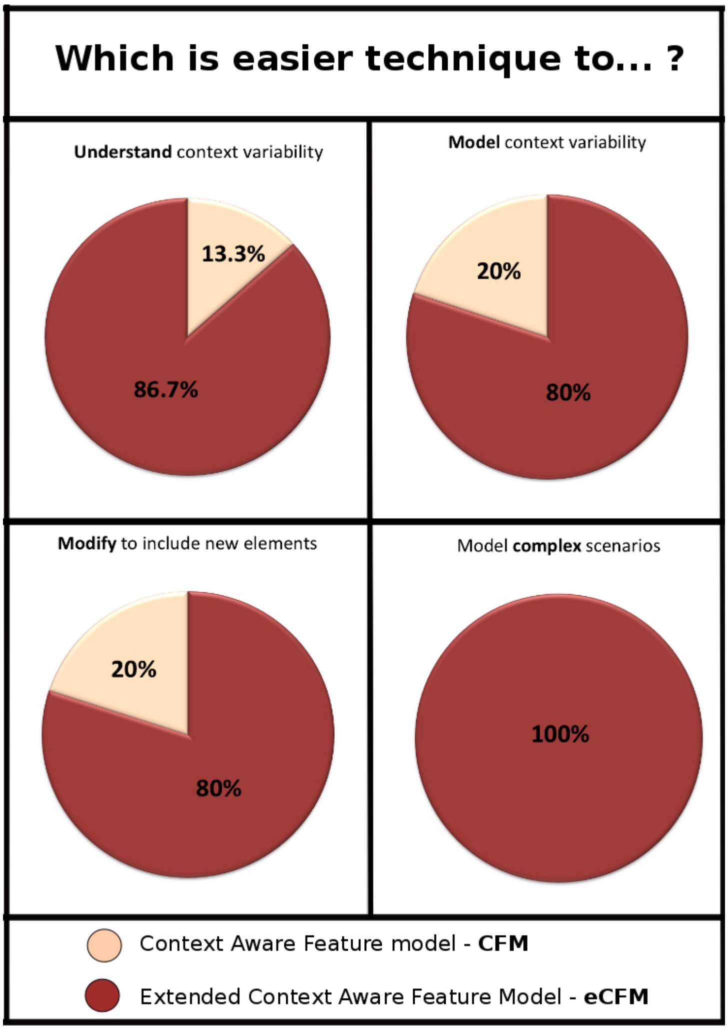
background, which can be evidenced according to their technical profiles. Among them, eight ones have large experience in the software industry with more than five years of activity. Additionally, they have been involved in academic or industrial projects, since eleven of them have worked in SPL projects, and seven have worked in DSPL projects. Furthermore, thirteen students have used software modeling techniques to deal with their projects.

Figure 3.3 presents an overview of the results that answer **RQ1: Which is the most expressive variability modeling technique to deal with DAS projects?** In order to gather some study participants feelings, we asked in a 5-point Likert scale ranging from *"Totally disagree"* to *"Totally agree"* about eCFM expressiveness. Such analysis was based on the following statement *"I believe that eCFM is more expressive than CFM"*. From the total, 93.3% agreed or totally agreed that eCFM is more expressive than CFM, *i.e.*, fourteen subjects argued that the eCFM is the most expressive technique concerning the representation of adaptation rules between the systems features and contexts. Only one subject disagreed regarding it.

Figure 3.4 present an overview of the results that answer **RQ2: Which is the easiest variability modeling technique to model DAS projects?** We asked about the easiest technique and obtained the following results: *(i)* understanding the context variability modeling: CFM - 13.3% and eCFM - 86.7%; *(ii)* modeling the DAS elements: CFM - 20% and eCFM - 80%; *(iii)* including new elements: CFM - 20% and eCFM -

80%; and *(iv)* modeling complex scenarios of DAS projects: 100% of the volunteers chose eCFM. We believe that these results were due to aspects, as follows:

Figure 3.4: The survey results for RQ2



- Contexts organized in groups make eCFM more understandable, since context features related (*e.g.*, **Warm** and **Cold**) are concentrated in a context group (*e.g.*, **Temperature**);
- Context groups make possible to model `alternative` and `OR`-dependency among contexts, which makes the modeling more expressive and allows higher flexibility to software engineer; and
- The concept of context group guides the designer to include new context information in a predefined model branch.

We noticed that the participants provided the best impressions related to eCFM, when we asked which technique is more suitable for DSPL modeling. One participant stated, for instance, *"eCFM, because it has the support of the model context that may not be present at the same time and it has groups of contexts"*. Another one also demonstrated his/her concern about expressiveness: *"eCFM, because you can group the contexts and make models in a more concise manner"*. In addition, other participants talked about the understandability: *"eCFM, for it's more clear as to where to find the contexts and in which situations to apply them"*. In general, the eCFM technique was considered more suitable to model the system's features and contexts in DSPL engineering.

3.3.4 Threats to Validity

We believe the main threats to the survey are those related to construction, internal and external validity.

Construction validity. As the sample for the study was small, we decided that all subjects should perform both treatments. In this way, we divided the subjects into two groups where each one applied the treatments in a different order, thus, the learning effect was balanced by reducing the threats to the construction validity.

Internal validity. The random diversity of the subjects is a potential threat affecting internal validity. In this sense, we selected the subjects with a similar background and applied a training session in order to balance their knowledge. Thus, it reduces the threat to the internal validity of the survey.

External validity. This study is one of few in this research topic, thus the findings of the analysis can be used as a baseline for comparing it with other studies. In addition, the artifacts developed may support replications since they were developed in detail and reviewed by several researchers to reduce the threat to the external validity of the survey.

3.4 CHAPTER SUMMARY

In this chapter, we proposed an extension of a VMT for DSPL modeling. We performed a survey with fifteen graduate students and two DAS with modeling artifacts to compare eCFM and CFM techniques. The results showed evidence that eCFM has benefits in the expressiveness that improves the model comprehensibility. Moreover, according to the evaluation results, eCFM seems easier to use than CFM for modeling the context variability.

In addition, we asked to subjects which technique is the most suitable to model DAS projects. As a result, all participants answered eCFM. Among their justifications to this answer, some of them stated that eCFM has a greater expressiveness power to represent adaptation rules between contexts and system features. Other answers highlighted the possibility to represent constraints among contexts using `alternative` groups. In addition, easiness of use and organization with the grouping of contexts were also cited by the participants as eCFM advantages against the CFM technique.

The next chapter presents a modeling approach to support the configuration selection process of DSPL. It is based on utility functions to formalize the knowledge obtained from stakeholder's preferences, the variability of system features, contexts, NFRs, and constraints.

Chapter

4

Strive for excellence, not perfection, because we don't live in a perfect world. —Joyce Meyer

TRADE-OFF ANALYSIS FOR DYNAMICALLY ADAPTABLE SOFTWARE

Essentially, we propose an approach, named ToffA-DAS, to manage both dimensions *structural variability* and *context variability*. It is addressed to the modeling of the system's features and contexts by meeting specific quality requirements. In this sense, ToffA-DAS is composed of eCFM and goal model as key references to assist communication with stakeholders. In addition, such an approach is used to support the configuration selection process of Dynamically Adaptable Software (DAS) by considering that contexts can influence the way of satisfying the NFRs of each model variant and vice versa [72].

We argue that the specification of DAS can be done by using the approach proposed in this section since it facilitates software engineers achieving consensus with stakeholders and understanding their preferences and needs. The ToffA-DAS approach aims to support trade-off analysis and can be reduced to an optimization model composed by a *utility function*. It uses the *Utility-based* planning [31] as a strategy to assist software engineers with trade-off analysis. This strategy can be used to express the priorities of users over services provided by an application [9]. Additionally, it uses a solver based on the ILP technique [89] to run the configuration process. As a result, it is possible to identify feasible configurations that meet all constraints. The remainder of this chapter is organized, as follows:

Section 4.1 discusses related work;

Section 4.2 presents our approach by describing five steps that the software engineer should perform in DSPL engineering;

Section 4.3 describes an example of how the configuration selection is performed; and

Section 4.4 presents concluding remarks.

4.1 RELATED WORK

Among the DSPL engineering activities, the variability modeling activity is one of the most important. There are different approaches supporting this activity [64, 72, 99]. It guides the software engineer to handle the diversity of contexts that influence the system's dynamic adaptations [4]. From the point of view of variability modeling, it is essential to provide support for the configuration selection process since it is considered as a complex optimization problem [19].

When dealing with feature selection to meet desired quality objectives in DSPL engineering, most of the existing studies do not focus on the interactions between contextual information and NFRs. In addition, such studies do not use any strategy to support the selection of the most suitable configuration [25]. As pointed out by Huebscher *et al.* [31], there is a planning type named *Utility-based* planning that enables us to find feasible configurations in accordance with the utility values that represent the desirable variants. The utility value is approximated by a utility function over contexts and NFRs. Thus, the feasible configurations can be found using a heuristic that approximates the impact of features over contexts and NFRs in the utility value.

Studies presented by Hallsteinsen *et al.* [9], Franco *et al.* [102], Edwards *et al.* [103], Paucar *et al.* [104], Esfahani *et al.* [105], Greenwood *et al.* [106], Guedes *et al.* [107], Nascimento *et al.* [108], and Sanchez *et al.* [109] use the utility function to approximate the fulfillment of stakeholder's preferences in different situations. Among them, only Franco *et al.* [102], Edwards *et al.* [103], and Paucar *et al.* [104] do not consider the trade-off between contextual information and NFRs in the decision making. Conversely, Hallsteinsen *et al.* [9], Esfahani *et al.* [105], Greenwood *et al.* [106], Guedes *et al.* [107], Nascimento *et al.* [108], and Sanchez *et al.* [109] propose approaches that model the interactions between such conflicting elements. Although those approaches assume the *Utility-based* planning as a strategy to formalize the knowledge obtained and deal with the interactions between contexts and NFRs, the authors do not consider or apply all modeling characteristics as follows:

- **Prioritization** – the degree which each the variable features satisfy the *soft goals*;
- **Satisfaction levels** – the relevance degree of *contexts*, *goals*, and *soft goals*; and
- **Contribution** – the impact of features over *contexts*, *goals*, and *soft goals.*

This is an important research gap since the number of product configurations increases exponentially with the number of features and many configurations satisfy the same requirements. Using all these modeling characteristics can better assist software engineers in analyzing and simulating solutions before implementing them.

Table 4.1 shows a comparison among the insights found in related work and our approach (ToffA-DAS). We inserted the studies presented by Pascual *et al.* [2], Hallsteinsen *et. al.* [9], Esfahani *et al.* [105], Greenwood *et al.* [106], Guedes *et al.* [107], Nascimento *et al.* [108], Sanchez *et al.* [109], Goldsby *et al.* [74], Parra *et al.* [80], Sawyer *et al.* [81], Ali *et al.* [110], and Gamez *et al.* [111], since they consider the trade-off between contextual information and NFRs.

Table 4.1: Related work summary

Approach	Information		Strategy	Modeling Characteristics		
	NFRs	Context	Utility-based planning	Satisfaction Level	Prioritization	Contribution
Pascual *et. al.* [2]	●	●	○	●	○	○
Hallsteinsen *et. al.* [9]	●	●	●	○	●	○
Goldsby *et. al.* [74]	●	●	○	●	●	○
Parra *et. al.* [80]	●	●	○	○	○	○
Sawyer *et. al.* [81]	●	●	○	●	○	○
Esfahani *et. al.* [105]	●	●	●	●	○	○
Greenwood *et. al.* [106]	●	●	●	○	●	○
Guedes *et. al.* [107]	●	●	●	●	●	○
Nascimento *et. al.* [108]	●	●	●	○	○	○
Sanchez *et. al.* [109]	●	●	●	○	○	○
Ali *et. al.* [110]	●	●	○	○	○	●
Gamez *et. al.* [111]	●	●	○	○	○	○
Welsh *et. al.* [112]	●	●	○	●	●	○
ToffA-DAS	●	●	●	●	●	●

●: Included; ○: Not included

Among them, we ticked which ones use *Utility-based* planning. The table cells (Table 4.1) that are highlighted in a different color depict such studies, which are presented by Hallsteinsen *et al.* [9], Esfahani *et al.* [105], Greenwood *et al.* [106], Guedes *et al.* [107], Nascimento *et al.* [108], and Sanchez *et al.* [109]. The *Utility-based* planning is considered a suitable strategy when the software engineers need to express priorities of users over services provided by an application [9]. The priorities, in turn, are represented as weights in the utility function aiming to direct the choice of a feasible solution. Esfahani *et al.* [105], Greenwood *et al.* [106], Nascimento *et al.* [108], and Sanchez *et al.* [109], for example, propose the usage of this strategy to perform trade-off analysis at runtime. Nevertheless, our approach supports the trade-off analysis at design time as well as the works presented by Hallsteinsen *et al.* [9] and Guedes *et al.* [107]. Therefore, it aims to assist software engineers during the initial modeling phase. In addition, our approach takes into account the combination of several modeling characteristics such as *satisfaction level*, *prioritization*, and *contribution* aiming to promote a more accurate representation of how DAS should operate in real-world environments.

4.2 SUPPORTING THE CONFIGURATION SELECTION PROCESS OF DAS

This work proposes an approach, named ToffA-DAS, to support the configuration selection process of DAS using the DSPL engineering principles. It is composed of eCFM and goal model as key references to assist communication with stakeholders. In addition, our approach is used to manage the two dimensions of variability, since it supports for modeling of *structural variability* by meeting specific requirements of *environmental variability*.

Figure 4.1 shows the steps of the approach proposed for supporting the configuration selection process in DSPL engineering. We explain it through a running example in the self-adaptive wireless sensor network domain named *GridStix* [74, 81, 113]. It establishes an automated mechanism to warn about floods on rivers. We illustrate the approach with this running example within boxes along this section.

Figure 4.1: The DAS Trade-off Analysis (ToffA-DAS) approach

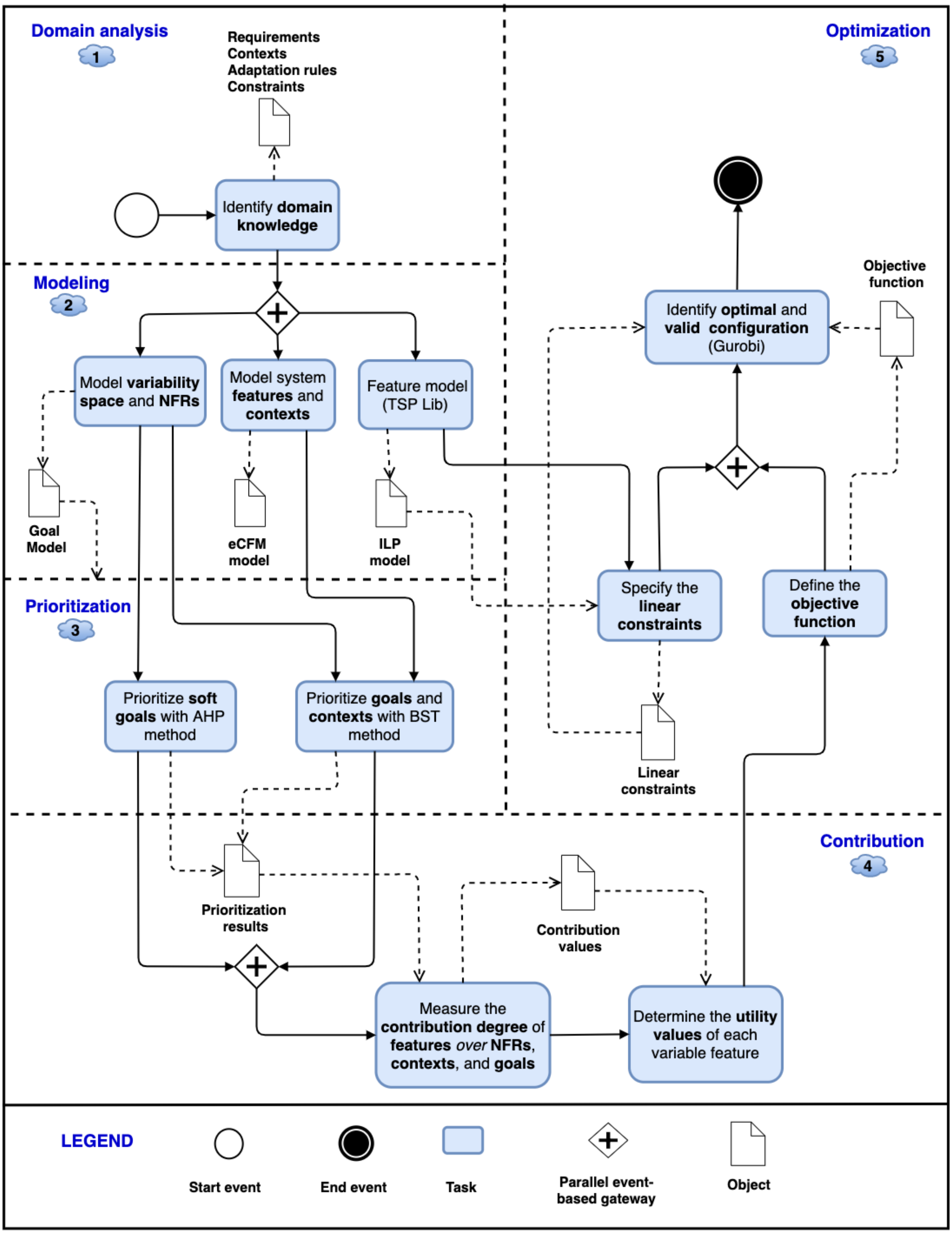
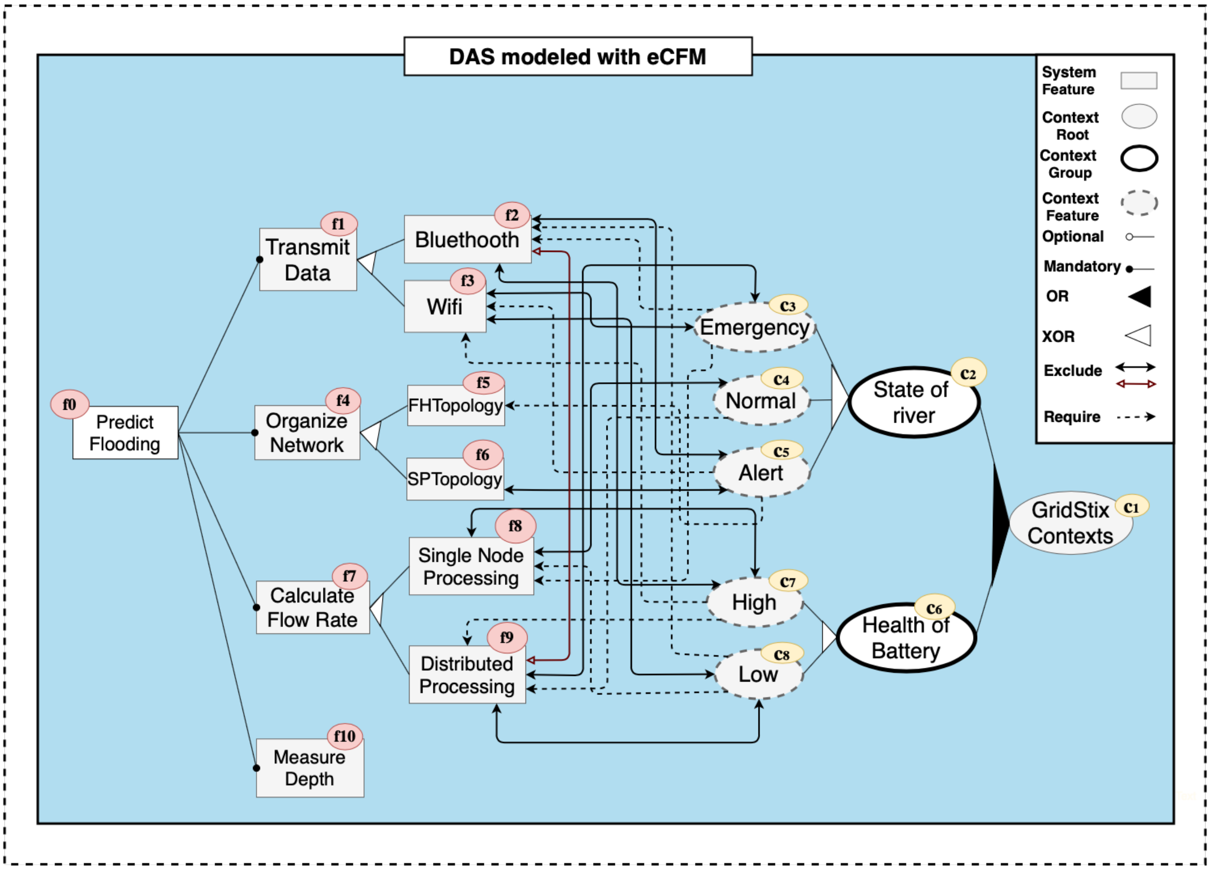


Figure 4.2: *GridStix* DAS modeled with eCFM. It represents all features, contexts features, context groups, context root, and their respective relationships.

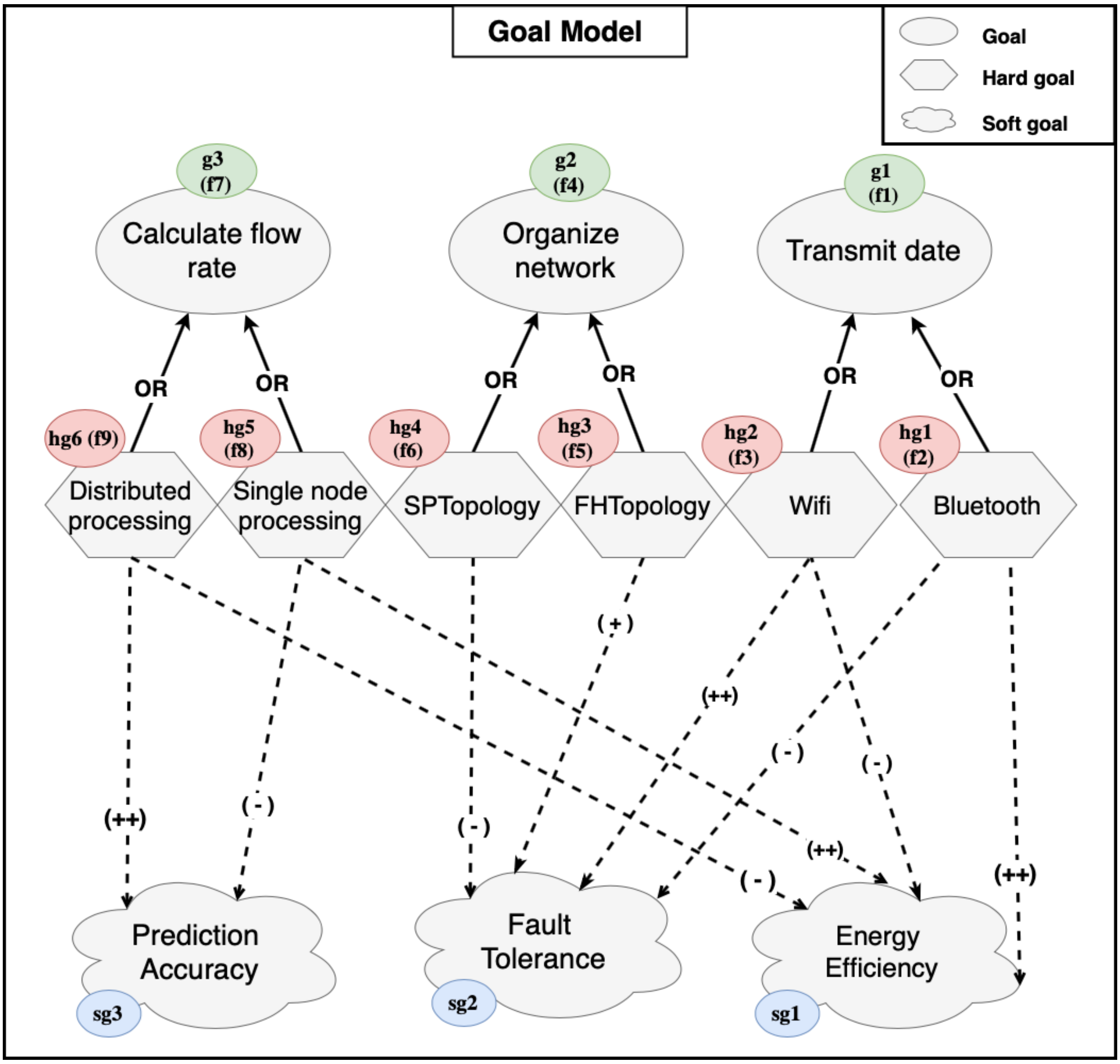


4.2.1 Domain analysis

STEP 1–Domain analysis is first performed by the software engineer to identify the following main elements for modeling DSPL as depicted in Chapter 3, as follows:

- **Functional requirements** represented as system features;
- **Variability information** specified as feature relationships of the types mandatory, optional, `OR`-group or `XOR`-group;
- **Context information** relevant to the DSPL;
- **Adaptations rules**, which are specified as dependencies between features and *contexts*;
- **Stakeholder's preferences**, which are defined based on the system's features;
- **Variability space**, which consists of the set of variable features that influences the decision making process; and

Figure 4.3: *GridStix* DAS modeled with Goal Model. It represents goals, hard goals, soft goals, and their respective relationships.



- **A set of required NFRs** that should be defined by the software engineer with the support, for instance, of a catalog such as the one proposed by Uchoa *et al.* [114].

4.2.2 Modeling

STEP 2–Modeling comes next. The software engineer should model the DSPL using eCFM and goal model techniques. The eCFM specifies both variabilities, feature and context. Figure 4.2 illustrates a flood-warn DSPL modeled with **eCFM**. For building such model, the software engineer should use the knowledge acquired at **STEP 1**, as follows:

For building such a model, the software engineer should use the knowledge acquired at **STEP 1**, as follows:

- **Functional requirements** and the **variability information** leads to the DSPL features and their relationships;
- **Contexts information** are used to define **Context Root**, **Context Groups**, and **Context Features**;
- **Adaptation rules** are represented in accordance with the **require** and **exclude** relationships. In the first relationship, the feature strongly satisfies the *context*, whereas, in the second, the feature is strongly denied by the *context*; and
- The dependencies between **context features** with their respective **context groups** should be represented in accordance with `XOR`-group, `OR`-group or `optional` relationships. For `XOR`-group relationship, two or more **context features** cannot occur at the same time, *i.e.*, only one **context feature** is mandatory to satisfy its **context group**. In contrast, in `OR`-group and `optional` relationships, all **context features** may satisfy their **context group**.

The running example encompasses four control systems, as follows:

- **Transmit data** (`mandatory` feature) is composed by two `alternative` features such as **Bluetooth** and **Wifi** for internode data transmition;
- **Organize network** (`mandatory` feature) is composed by two `alternative` features such as **FHTopology** and **SPTopology** for routing data between nodes;
- **Calculate flow rate** (`mandatory` feature) is composed by two `alternative` features such as **Single Node Processing** and **Distributed Processing** to facilitate the operation of nodes for extended periods of time.; and
- **Measure depth** (`mandatory` feature) of water.

Both **State of River** and **Health of Battery** were identified as **context groups**, which can influence possible adaptations. Each one is composed by three and two **context features**, respectively. For instance, whether *contexts* **Low** and **Emergency** are detected in the environment execution, the feature **Bluetooth** may be activated (require relationship). In contrast, the feature **Wifi** may be deactivated (exclude relationship). The same reasoning can be applied for all adaptation rules.

In the running example, we consider that all **context features** have `XOR`-group relationship with their respective **context group**. For instance, *contexts* c_7 (**High**) and c_8 (**Low**) cannot occur at the same time in a real world environment.

The goal model (see Figure 4.3) encompasses the high-level *goals*, *hard goals*, *soft goals*, and mapping links among them. For building such model, the software engineer should use the variability space, the set of required NFRs, and stakeholder's requirements identified at **STEP 1**. The mapping link between *goals* and *hard goals* represents the `AND`/`OR` relationships, as follows:

- The `AND` relationship means that all *hard goals* are mandatory to satisfy the *goal*;
- The `OR` relationship means that each individual *hard goal* can satisfy its *goal*; and
- The mapping link between *hard goals* and *soft goals* defines the satisfaction level: **satisfied** (++) = 1, **weakly satisfied** (+) = 0.5, **undecided** (?) = 0, **weakly denied** (-) = -0.5, and **denied** (- -) = -1.

Figure 4.3 shows the goal model of the running example, which is composed by three *soft goals*: **Energy Efficiency** (sg_1), **Fault Tolerance** (sg_2), and **Prediction Accuracy** (sg_3). In addition, we defined the following *goals*:

- **Transmit date** (g_1), which includes the *hard goals* **Bluetooth** and **Wifi** (hg_1 and hg_2);
- **Organize network** (g_2), which includes the *hard goals* **FHTopology** and **SP-Topology** (hg_3 and hg_4); and
- **Calculate flow rate** (g_3), which includes the hard goals **Single node processing** and **Distributed processing** (hg_5 and hg_6).

We used the TSP library [115] to numerically represent the features and constraints that were defined in the model (see **STEP 2** in Figure 4.1). Such library of sample instances is used by a solver named Gurobi [29], which is based on the ILP technique to run the configuration process and find feasible and valid configurations that meet all constraints.

Figure 4.4 shows an example of TSP library defined for *GridStix* DSPL. The dimension consists of the total of the system's features (in the running example this number is equal to 11). Each section represents the relationship between the system's features. In section **FEATURE MODEL**, for instance, we represented 10 hierarchical relationships between child feature and its respective parent feature. The child feature **Transmit Data** (f_1) has a hierarchical relationship with the parent feature **Product Flooding** (f_0), whereas the child feature **Bluetooth** (f_2) has a hierarchical relationship with the parent feature **Transmit Data** (f_1). The same reasoning is applied to all relationships.

Figure 4.4: TSP library

```
NAME : GRIDStix
COMMENT : (TSP library)
TYPE : DSPL
DIMENSION : 11
FEATURE_MODEL_SECTION : 10
1 0 1
2 0 4
3 0 7
4 0 10
5 1 2
6 1 3
7 4 5
8 4 6
9 7 8
10 7 9
MANDATORY_SECTION : 4
1 0 1
2 0 4
3 0 7
4 0 10
OPTIONAL_SECTION : 0

OR_SECTION : 0

ALTERNATIVE_SECTION : 3
1 2 3
2 5 6
3 8 9
REQUIRE_SECTION : 0

EXCLUDE_SECTION : 1
1 2 9

EO
```

Since all modeling elements are already captured and modeled, then goals, soft goals, and *contexts* should be prioritized according to the stakeholder's preferences. The prioritization results can be used as parameters to measure the contribution of the system's features (*hard goals*) connected to the *contexts*, *goals*, and *soft goals*.

4.2.3 Prioritization

Since all modeling elements are already captured and modeled, then *goals*, *soft goals*, and *contexts* should be prioritized according to the stakeholder's preferences. The prioritization results can be used as parameters to measure the contribution of the system's features (hard goals) connected to the *contexts*, *goals*, and *soft goals*.

The **STEP 3 – Prioritization** consists of the relevance degree of *goals*, *contexts*, and *soft goals*. Two ranking and prioritization methods are used: Analytical Hierarchy Process (AHP) [116] and Binary Search Tree (BST) [117]. We evidenced that these ranking and prioritization methods are used to better meet stakeholder's preferences. With the AHP method, for instance, it is possible to check the consistency of the results based on a ratio scale. Indeed, the AHP method brings a scalability problem for larger projects. However, it can be reduced with the use of tools to support the configuration process described in this work [117]. The combination of both methods, BST and AHP help us to communicate with stakeholders and identify the potential preferences in terms of system quality.

4.2.3.1 Prioritization of goals - The BST method was used to rank the selected goals, which is an efficient method for prioritizing large-scale items [117]. In this method, each goal is represented by a node of the tree and has sub-nodes, which can be *sub goals* or *hard goals*. The tree is organized according to priorities of *goals* and *hard goals*. Then, such elements must be ordered from right to left in order to specify the sub-tree. It means that the right side of the sub-tree contains requirements with higher priority than the left side of the sub-tree. The prioritization of goals can be conducted according to the process described by [118], as follows:

We identify at the variability space, *goals* and , *hard goals* related to functional requirements. Next, we selected a *goal* with a higher priority and put it as a root node. After that, another , *goal* g_b can be selected and compared with the root , *goal* in terms of its importance. If g_b has a lower priority than the root node g_a, it needs to be compared with the left sub-node and so forth. This process should be repeated until all *goals* have been compared and inserted into the BST. This same process should be made with, *hard goals*.

> First, three high-level , *goals* related to functional requirements (g_1 – **Transmit date**, g_2 – **Organize network**, and g_3 – **Calculate flow rate**) and six hard goals (hg_1 – **Bluetooth**, hg_2 – **Wifi**, hg_3 – **FHTopology**, hg_4 – **SPTopology**, hg_5 – **Single node processing**, and hg_6 – **Distributed processing**) have been identified. Next, each , *goal*, starting with g_1, was compared with all others (Figure 4.3) and the following order or priority has been established g_2, g_1, and g_3.

Thereafter, the tree should be traversed from right to left to identify the rank of , *goals* and normalize their values. Then, a normalized rank value is assigned to each , *goal* according to the formula:

$$rankValue = \frac{1}{1 + [rank]} \tag{4.1}$$

The rank is a series of crescent natural numbers starting from 1. Thus, the $rankValue$ may be transformed into a scale ranging from 0 (exclusive) to 1 (exclusive), where the value close to zero means the lowest priority and the value close to one means the highest priority.

> Table 4.2 shows the rank value for each *goal* and the related normalized $rankValue$ of the running example.

4.2.3.2 Prioritization of contexts - It was used the same steps aforementioned described for prioritization of *contexts*.

> Following the running example, two high-level *contexts* needed to be prioritized (c_2 – **State of river** and c_6 – **Health of Battery**). Starting with c_2 we compared it

with c_6 in terms of its importance and defined the following order: c_2 and c_6. After that, we calculated the rank values and normalized them as shown in Table 4.2.

Table 4.2: Prioritization using the BST method (GridStix DSPL)

	Goal			Context	
	g_2	g_1	g_3	c_2	c_6
$rank$	1	2	3	1	2
$rankValue$	0.5	0.33	0.25	0.5	0.33

4.2.3.3 Prioritization of soft goals - We can determine the priority of *soft goals* according to the stakeholder's preferences using the AHP method. In addition, a specific *hard goal* may influence *soft goals* of other *hard goals*. It is also based on a pairwise comparison process to generate a ranked list of *soft goals*. In addition, it evaluates and checks the consistency of judgments.

Assuming SG as a set of *soft goal*, a comparison matrix $A\big[n,n\big]$ must be created to show the relative importance of each pair of *soft goal*. The *soft goals* are compared against each other according with the scale of importance: 1 (**equal**), 3 (**moderate**), 5 (**strong**), 7 (**very strong**), and 9 (**extreme**), respectively [116].

If the *soft goal* in column j is preferred to the *soft goal* in row i, then the inverse of the rating is given ($a_{j,i} = \frac{1}{a_{i,j}}$), *i.e.*, we put the actual judgment value on the right side of the matrix diagonal row and the reciprocal value in the left side of the diagonal.

The next step is to normalize the comparison matrix and calculate the importance value ($iValue$) for each *soft goal* (equation 4.2). Totaling the numbers in each column does this step. Each entry in column j is then divided by the column sum to yield its normalized score. As a result, the sum of each column is 1. The $iValue$ is used to measure the contribution of features over NFRs. It is calculated as follows:

$$iValue = \frac{\sum_{i=1}^{n} A\big[i,j\big]}{n} \tag{4.2}$$

Table 4.3 shows the complete comparison matrix $A\big[3,3\big]$. It shows the relative importance of each pair of *soft goals*. In addition, Table 4.4 shows the normalized matrix of this running example and the $iValue$ for each *soft goal*. The $iValue$ is used to measure the contribution of features over NFRs.

The software engineer can decide between different and equal priority. For instance, whether the soft goals sg_1, sg_2, and sg_3 have the same priority, their rank values are equal to 1. The same reasoning can be applied to all *goals* and *contexts*.

Table 4.3: Relative importance matrix $A[n, n]$ (GridStix DSPL)

	sg_1	sg_2	sg_3
sg_1	**1**	3	3
sg_2	0.33	**1**	1
sg_3	0.33	1	**1**
sum	1.66	5	**5**

Table 4.4: The normalized matrix

	sg_1	sg_2	sg_3	$iValue$
sg_1	0.6	0.6	0.6	**0.6**
sg_2	0.2	0.2	0.2	**0.2**
sg_3	0.2	0.2	0.2	**0.2**
sum	1	1	1	**1**

4.2.4 Contribution

STEP 4 – Contribution consists of the impact of features over *goals*, *soft goals*, and *contexts*. It is performed after prioritizing all the model elements.

4.2.4.1 Features over contexts - We use a diagrammatic reasoning approach introduced by Ali *et al.* [72] to calculate the contribution degree of features over *contexts* (equation 4.3). It shows that **context groups** are decomposed into **context features** by `AND`–decomposition (`Mandatory`-group or `XOR`-group) and `OR`–decomposition (`OR`-group or `Optional`-group). Such *context* are satisfied by means of **context features**, as follows:

$$Cont\Big(f_i, c_{fi}\Big) = \sum_{\forall c \in C | f \to c} rankValue\Big(c_{gi}\Big) \times satValue\Big(c_{fi}\Big) \times impDegree\Big(c_{fi}\Big) \quad (4.3)$$

The function $rankValue$ calculates the priority value of a **context group** (c_{gi}). The function $satValue\Big(c_{fi}\Big)$ shows to what extent each **context feature** (c_{fi}) can satisfy its **context group** by considering the `AND/OR` relationships. The function *impDegree* displays to what extent each feature can satisfy a **context feature** in the eCFM according to the `require` and `exclude` relationships. The first is represented by an impact degree with value **1**, meaning that the feature strongly satisfies the **context feature**. The second is represented by impact degree with value **-1**, meaning that the feature is strongly denied by **context feature**.

We identify the `AND/OR` relationships of each **context feature** (c_{fi}) with its respective **context group** (c_{gi}). Following the running example, we consider that all **context features** have `AND` relationship with their respective **context group**, resulting in $satValue\Big(c_{fi}\Big) = 1$. Thus, feature f_2 for instance, has contribution degree over *contexts* c_3 and c_8, as follows:

(i)

$$Cont\Big(f_2, c_3\Big) = rankValue\Big(c_2\Big) \times satValue\Big(c_3\Big) \times impDegree\Big(c_3\Big) = 0.5 \times 1 \times 1 = 0.5$$

(ii)

$$Cont\Big(f_2, c_8\Big) = rankValue\Big(c_6\Big) \times satValue\Big(c_8\Big) \times impDegree\Big(c_8\Big) = 0.33 \times 1 \times 1 = 0.33$$

(iii)

$$Cont\Big(f_2, c_3\Big) + Cont\Big(f_2, c_8\Big) = 0.5 + 0.33 = 0.38$$

4.2.4.2 Features over goals - We also use a diagrammatic reasoning approach introduced by Ali *et al.* [72] to calculate the contribution degree of features over *goals* (equation 4.11). It shows that top-level goals are iteratively decomposed into sub goals by `AND`–decomposition and `OR`–decomposition. Such goals are satisfied by means of executable tasks (*i.e.*, hard goals), as follows:

$$Cont\Big(f_i, g_i\Big) = rankValue\Big(g_i\Big) \times satValue\Big(hg_i\Big) \tag{4.4}$$

Where $rankValue\Big(g_i\Big)$ shows the priority value of a *goal* and $satValue\Big(hg_i\Big)$ shows to what extent each *hard goal* can satisfy its goal. We calculate $satValue\Big(hg_i\Big)$ based on the `AND/OR` relationships. In the `AND` relationship, $satValue\Big(hg_i\Big)$ is divided by the number of *hard goals* m. Otherwise, in an `OR` relationship, the satisfaction value is 1, since each individual *hard goal* can satisfy its *goal*.

In the running example, the *hard goals* hg_1 (feature f_2) and hg_2 (feature f_3) have an `OR` relationship with their parent *goal* g_1 resulting in $satValue\Big(hg_1\Big) = 1$ and $satValue\Big(hg_2\Big) = 1$. Thus, the impact of features f_2 and f_3 over *goal* g_1 can be computed, as follows:

(iv)

$$Cont\Big(f_2, g_1\Big) = satValue\Big(hg_1\Big) \times rankValue\Big(g_1\Big) = 1 \times 0.33 = 0.33$$

(v)

$$Cont\Big(f_3, g_1\Big) = satValue\Big(hg_2\Big) \times rankValue\Big(g_1\Big) = 1 \times 0.33 = 0.33$$

4.2.4.3 Features over soft goals - Contribution of features over *soft goals* (equation 4.12) can be identified by the mapping links in Figure 4.3 and calculated as follows:

$$Cont\Big(f_i, sg_i\Big) = \sum_{\forall sg \in SG | f \to sg} iValue\Big(sg_i\Big) \times impDegree\Big(sg_i\Big) \tag{4.5}$$

Where $iValue$ is the importance value of *soft goals* and $impDegree\Big(sg_i\Big)$ shows to what extent each feature can satisfy a *soft goal* in the goal model, based on the conversion schema for satisfaction level: $\Big(--\Big) = -1$, $\Big(-\Big) = -0.5$, $\Big(?\Big) = 0$, $\Big(+\Big) = 0.5$, $\Big(++\Big) = 1$.

Following the running example, feature f_2 (hg_1) is related to *soft goals* sg_1 and sg_2, as follows:

(vi)

$$Cont\Big(f_2, sg_1\Big) = iValue\Big(sg_1\Big) \times impDegree\Big(sg_1\Big) = 0.6 \times 1 = 0.6$$

(vii)

$$Cont\Big(f_2, sg_2\Big) = iValue\Big(sg_2\Big) \times impDegree\Big(sg_2\Big) = 0.2 \times (-0.5) = -0.1$$

(viii)

$$Cont\Big(f_2, sg_1\Big) + Cont\Big(f_2, sg_2\Big) = 0.6 + (-0.1) = 0.5$$

The **utility values** $\mathbb{C}_{fi}$ of each variable feature f_i can be determined by evaluating the contribution degree of its associated *hard goals* over *goals* and *soft goals*, besides of the contribution degree of its associated *hard goals* over *contexts* (equation 4.13), as follows:

$$\mathbb{C}(f_i) = \sum_{u=1}^{|G|} Cont\Big(f_i, g_u\Big) + \sum_{t=1}^{|SG|} Cont\Big(f_i, sg_t\Big) + \sum_{v=1}^{|C|} Cont\Big(f_i, c_v\Big) \tag{4.6}$$

Table 4.5: Utility value for features by considering Goal, Soft goal, and Context (Configuration F_1 for GridStix)

System feature	$Cont\Big(fi, Cs\Big)$	$Cont\Big(fi, Gs\Big)$	$Cont\Big(fi, SGs\Big)$	$\mathbb{C}(fi)$
f_2	0.83	0.33	0.5	1.66
f_3	-0.33	0.33	-0.1	-0.1
f_5	0	0.5	0.1	0.6
f_6	0	0.5	-0.1	0.4
f_8	0.83	0.25	0.5	1.58
f_9	-0.83	0.25	-0.1	-0.68

Table 4.5 presents the utility values for the running example that are measured, as follows:

(i) The sum of contribution values for feature f_2 over *contexts* c_3 and c_8 resulted in a *utility value* equal to 0.38 (equation 3). This process should be repeated for all features and their respective **context features** in order to complete the first column of the table;

(ii) Since the features f_2 and f_3 only contribute over *goal* g_1, the resulting *utility values* are equal to 0.5 and 0.5 (equations 4 and 5). This process should be repeated for all features and their respective *goals* in order to complete the second column of the table; and

(iii) The sum of contribution values for feature f_2 over *soft goals* sg_1 and sg_2 resulted in a *utility value* equal to 0.5 (equation 8). This process should be repeated for all features and their respective *soft goals* in order to complete the third column of the table.

The utility values are coefficients of the decision variables. The utility value $\mathbb{C}_{fi}$ will be used in the optimization model.

4.2.5 Optimization

After executing the previous steps, the software engineer should perform the **STEP 5 – Optimization**. In this sense, we used utility function as a strategy to deal with trade-off analysis. Based on the utility values, we defined an optimization model that recommends feasible configurations by considering all integrity constraints and variability of DSPL models. This optimization model ensures that the features combination satisfy the NFRs, *contexts*, stakeholder's preferences, and constraints. It is characterized, as follows:

(1) A set of **decision variables** X_{fi} whose value is 1 if the feature x_{fi} is active, 0 otherwise;

(2) An **objective function** that measures the decision variables summation by satisfying the *goals*, *soft goals*, and *contexts* represented in eCFM and the goal model (equation 4.7). The result of the problem formulation considers the utility value $\mathbb{C}_{fi}$ of the system features (*hard goals*), as follows:

$$\max \ \sum_{fi}^{n} \mathbb{C}_{fi} \cdot X_{fi}, \forall f_i \in F \tag{4.7}$$

In the running example, features that are not variable and are not represented as *hard goals* in the goal model receive value 0 as coefficient to eliminate their impact on the maximization of the objective function, as follows:

$$\begin{aligned}\max \ 0.0X_{f0} + 0.0X_{f1} + 1.66X_{f2} - 0.1X_{f3} + 0.0X_{f4} + 0.6X_{f5} \\ + 0.4X_{f6} + 0.0X_{f7} + 1.58X_{f8} - 0.68X_{f9} + 0.0X_{f10}\end{aligned}$$

(3) A set of **linear constraints** that are subject to the variability and integrity constraints of eCFM and the goal model (equations 4.8, 4.9, 4.10, 4.11, 4.12, and 4.13). They are based on the relationship model described by Kang *et al.* [75], as follows:

Let F be a set of features, $|F| = n$ and F^m be a feature model that represents a hierarchical relationship between features. Thus, the ordered pair of features $\left(f_p, f_c\right) \in F^m$ if f_c is a child of f_p.

Let $M \subseteq F^m$ be the set of pair of features with `mandatory` relation. If $\left(f_p, f_c\right) \in M$, then both features f_p and f_c must be activated or deactivated at the same time:

$$x_{fc} = x_{fp}, \left(f_p, f_c\right) \in M \tag{4.8}$$

The `optional` relationship denotes:

$$x_{fc} \leq x_{fp}, \left(f_p, f_c\right) \in F^m \tag{4.9}$$

The set $O \subseteq F^m$ denotes the `OR` relation:

$$\sum x_{fc} \geq x_{fp}, \left(f_p, f_c\right) \in O, \forall f_c \in F \tag{4.10}$$

The set $A \subseteq F^m$ denotes the `alternative` relation:

$$\sum x_{fc} = x_{fp}, \left(f_p, f_c\right) \in A, \forall f_c \in F \tag{4.11}$$

The set $R \subseteq F \times F$ denotes the `require` relation between features, thus $\left(f_r, f_k\right) \in R$ means that if f_r is activated then f_k must be activated:

$$x_{fr} \leq x_{fk}, \left(f_r, f_k\right) \in R \tag{4.12}$$

The set $E \subseteq F \times F$ denotes the `exclude` relation between features, thus $\left(f_e, f_k\right) \in R$ means that f_e and f_k must not be activated at the same time:

$$x_e + x_k \leq 1, \left(f_e, f_k\right) \in E \tag{4.13}$$

In the running example, we defined the constraints, as follows:

- `Mandatory` relations between features (Constraint 4.8): $X_{f1} = X_{f0}, X_{f4} = X_{f0}, X_{f7} = X_{f0}, X_{f10} = X_{f0}$;
- Considering the features with an `alternative` relation, if a parent feature is activated only one child will be activated (Constraint 4.11): $X_{f2} = X_{f1}, X_{f3} = X_{f1}, X_{f5} = X_{f4}, X_{f6} = X_{f4}, X_{f8} = X_{f7}, X_{f9} = X_{f7}$;
- Feature f_k will be deactivated if it has an exclude relationship with f_e (Constraint 4.13): $X_{f2} + X_{f9} \leq 1$.

After executing the previous steps, the software engineer is able to select a feasible set of features that can maximize stakeholder's preferences and meet the scarce resources. In this sense, we developed an optimization model in the C++ language that receives as input the expressions created in the previous step referring to the algebraic form of the *utility function* (equation 4.7) and the constraints 4.8 to 4.13. It uses a solver based on the ILP technique [29] to run the configuration process and find feasible configurations that meet all constraints.

4.3 EXECUTING THE CONFIGURATION SELECTION

The approach is partially automated to perform the measurements of prioritization, contribution, and satisfaction level.

> For the current example, the time spent in the execution was 0.20 seconds. In addition, the output of the ILP solver [29] suggested that the set of features $F_1 = \{f_1, f_2, \neg f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}\}$ satisfies the *soft goals* ($sg_1 - sg_3$), *goals* ($g_1 - g_3$), and the *contexts* ($c_1 - c_8$) to the aforementioned prioritization. It indicated that features f_3, f_6, and f_9 negatively influences the *soft goals* of other features, such as *energy save* and *fault tolerance*.
>
> For example, the contribution values in feature f_9 (*e.g.*, -0.83 and -0.1) indicate that that they negatively influence the *contexts* and *soft goals* in DSPL. In addition, it has an `exclude` relationship with feature f_2, which was kept in the feasible configuration.

Therefore, the DSPL developer must deal with the integrity constraints in terms of implementation by considering the adaptation rules represented in eCFM.

4.4 CHAPTER SUMMARY

The DSPL paradigm extends the traditional software product line enabling dynamic adaptations at runtime. Such adaptations are triggered by *context* changes, and they affect the product configuration and the satisfaction of the NFRs. Thus, aiming to achieve the feasible configuration, it is important to investigate how the trade-off between *contexts* and NFRs affects the product configuration.

This chapter presented a modeling approach that facilitates software engineers in achieving consensus with stakeholders and understanding their preferences and needs. It supports the configuration selection process of DAS based on utility function to formalize the knowledge obtained from stakeholder's preferences, the variability of system features, *contexts*, NFRs, and constraints. To this end, we defined an optimization model to ensure feasible configurations by satisfying the requirements.

The next chapter presents a study based on simulations to gather initial evidence about the feasibility of using ToffA-DAS. It is based on how to conduct trade-off analysis and define adaptation models from feasible configurations found in the analysis. It also presents an exploratory study to evaluate how the configurations obtained by the execution of ToffA-DAS affect the overall satisfaction level of NFRs.

Chapter

5

Patience is not simply the ability to wait - it's how we behave while we're waiting. —Joyce Meyer

FEASIBILITY OF USING THE TOFFA-DAS APPROACH

Aiming to assess the ToffA-DAS, we conducted two studies based on simulations. Firstly, we searched for reasoning on adaptability to gather initial evidence about the feasibility of using our approach. Next, we investigated how the configurations obtained affect the overall satisfaction level of NFRs. In this second study, we used two DAS in the mobile and smart home domains, respectively. The results showed evidence that the ToffA-DAS approach can select a configuration that best meets contexts and NFRs by considering their priorities. Hence, it may support software engineering in the identification of feasible configurations. This chapter consists of three main sections, as follows:

Section 5.1 discusses the insights concerned with conflict resolution in DSPL engineering. In addition, it reports studies that use *Utility-based* planning as a strategy to deal with the configuration selection process;

In **Section** 5.2, we describe how our approach can support software engineers for the reasoning of adaptability;

Section 5.3 presents the exploratory studys design and planning, the analysis, and the interpretation of the results;

Section 5.4 discusses the results of the exploratory study and the threats to validity; and

Section 5.5 presents concluding remarks.

5.1 RELATED WORK

The modeling characteristics presented in Table 4.1 (Chapter 4) are employed to support software engineers in decision-making. Aiming at identifying the approaches that have demonstrated a likely similarity with our approach, we decided to use such modeling characteristics as comparison criteria, as follows:

- The **satisfaction level** criterion aims to identify whether the authors consider, in their approach, the degree which each the variable features satisfy the *soft goals*;
- The **prioritization** criterion verify whether the approach under evaluation uses the relevance degree of *contexts*, *goals*, and *soft goals*; and
- The **contribution** criterion evaluates whether the authors use, in their approach, the impact of features over *contexts*, *goals*, and *soft goals*.

Hallsteinsen *et. al.* [9] reported conceptual discussions about how to build DAS projects based on the approach named MADAM. It uses annotations to reason about how well a variant of DSPL meets its context. For this purpose, the NFRs provided by the DAS application are compared to those required by the user and those provided by such variants. The match to the user's needs is expressed in a utility function and is used to direct the adaptation. The utility function represents a weighted mean of the differences between the NFRs provided by the DAS application and the user's preferences over those NFRs. Therefore, the weights represent the priorities of the user and the utility function calculates the benefit of a specific variant of DSPL. Although this proposal aims to automatically derive changing requirements by monitoring the context and automatically reconfigure the application while it is running, they do not mention the variability model at runtime and how the priorities are measured.

Esfahani *et. al.* [105] provided a framework named Feature-oriented Self adaptatION (FUSION), which combines feature-models with machine learning and in turn improves the accuracy and efficiency of adaptation decisions. Features and NFRs are modeled with the use of goal models. In turn, a *goal* embraces a metric, which is a measurable quantity obtained from the system execution and a utility. The utility function is used to express the user's preferences for achieving a particular metric. In other words, a *goal* defines the user's degree of satisfaction over *soft goal* (*e.g.*, response time) by achieving a specific value of the metric at runtime. The FUSION approach defines several learned functions to estimate the impact of selecting a specific set of features by considering the metrics at a given execution context.

Greenwood *et. al.* [106] presented the DiVA approach, which provides a tool-supported methodology for managing dynamic variability in DAS projects by using the DSPL engineering processes. This approach considers the specific context to which each variation is applicable, as well as, how each variant of the DSPL affects all the system and its NFRs. The DiVA approach uses model-driven techniques to model these variability elements and formalize *how* and *when* the system should adapt. In addition, it combines the strengths of both strategies, *ECA rules* and *utility-based* planning aiming to achieve efficiency, scalability, and verification of capabilities. Then, the adaptation rules are expressed as high-level *goals* to achieve and the configuration is optimized with respect to these *goals* at runtime. These rules are defined using expressions to describe the *context* that they apply and a set of priorities assigned to the NFRs. In addition, the utility functions are defined to determine how well suited a configuration is, depending on the *context*.

Guedes *et. al.* [107] proposed an approach called Contextual Goal models For Dynamic Software product lines (ConG4Das) that captures the variability of adaptive systems. It is based on a goal model to represent information such as context, NFRs, the relationship between them, and their priority. Regarding prioritization, the ConG4Das approach allows that a given *context* is ranked according to the priority of NFRs. It means that the *contexts* affect the required satisfaction level of NFRs.

Nascimento *et. al.* [108] proposed a DSPL infrastructure, called ArCMAPE to support a family of software fault tolerance techniques based on design diversity and instantiates the most suitable one through dynamic variability management. When a requisition is sent to ArCMAPE, the adaptation logic intercepts the running system. In turn, the dynamic adaptation satisfies the rules by maximizing the utility value, which is measured based on pre-defined weights for NFRs. The new configuration is chosen in accordance with input values provided by sensors and behavioral change of the running system. Therefore, the adaptation is triggered by contextual changes or changes in NFRs.

Sanchez *et. al.* [109] proposed an approach for the specification, measurement, and optimization of NFRs based on feature models. It shows how NFRs can be specified by means of feature attributes by quantitatively evaluating the trade-off among multiple NFRs to arrive at a better system's configuration. Their approach requires the mapping of the contexts, events to feature models, the specification of values for NFRs, quality metrics, and weights for the optimization steps. This approach focuses on the quantification of individual attributes and for trade-off among these different metrics, formalizing the problem as the optimization of an objective function that aggregates these metrics and quantifies stakeholder's preferences for individual attributes.

The aforementioned studies provide important information about the configuration selection process of DAS by considering the trade-off between contexts and NFRs. Among those approaches that assume *Utility-based* planning as a strategy to formalize the knowledge obtained, only Esfahani *et. al.* [105] and Guedes *et. al.* [107] use *goals* to model *how* and *why* the system operates. Both assess the satisfaction level of *soft goals* by varying according to the *context* changes. Conversely, our approach measures the degree to which the variable features satisfy the *soft goals*. In addition, it evaluates the impact of variable features over *contexts*, *goals*, and *soft goals*. Therefore, *contexts* and *soft goals* elements are handled in an independent way.

Additionally, the approach proposed by Guedes *et al.* [107] does not deal with model integrity constraints according to the relationships between feature and context. In contrast, we propose the eCFM technique to improve the context variability expressiveness by specifying real world constraints related to contextual information. Some of the main benefits of the eCFM are: *(i)* it allows to model constraints among **context features** increasing the *context* variability expressiveness; and *(ii)* the **context group** concept allows to organize the **context features** in different categories defined by the software engineer (*e.g.*, based on purposes or *context* source) favoring the model organization and comprehensibility.

In Figure 5.1 in section 5.2.2, for instance, one of the functional requirements of the self-adaptive wireless sensor network is represented by the system's feature `Transmit Data` that controls the type of communication wireless. This feature has two variations with an `XOR`-group: `Bluetooth` and `Wifi`. The feature `Bluetooth` has an `exclude` dependency relationship with the **context feature** `Emergency`. So, it specifies an adaptation rule that is triggered by the **context feature** `Emergency`.

We also use the goal model to represent variable features and the set of required NFRs for satisfying stakeholder's intentions. Nevertheless, both eCFM and goal model are integrated in order to support the software engineer in the decision-making process. Since we provide an approach that takes into account the combination of several modeling characteristics such as *satisfaction level*, *prioritization*, and *contribution*, it is possible to promote a more accurate representation of how DAS should operate in real-world environments.

5.2 REASONING ABOUT ADAPTABILITY

In this section, we describe the usefulness of the ToffA-DAS approach from three different points of view: firstly, how to conduct trade-off analysis by considering all the elements that comprise eCFM and goal model (Section 5.2.1). Secondly, which feasible configuration meets a specific combination of context features (CCF) (Section 5.2.2). Finally, how the approach supports the definition of adaptation models for DAS from feasible configurations found in the CCF change analysis (Section 5.2.3). Such a description of how using the ToffA-DAS approach is based on gathered data from different simulations.

5.2.1 Trade-off analysis

Stakeholder's preferences change over time and are hard to elicit. Thus, we proposed the trade-off analysis aiming to find valid and feasible configurations that can meet such preferences. Trade-off analysis consists of simulating changes in the prioritization of *contexts*, *goals*, and *soft goals*. For each change, software engineers must only consider relationships between the system's feature and context corresponding to a specific CCF.

Table 5.1 depicts the feasible configurations found in the trade-off analysis, whereas Table 5.2 presents for which CCF a specific feasible configuration was suggested by the ILP solver. We simulated six scenarios that correspond to a change of prioritization for soft goals, goals, and contexts. However, we considered the same satisfaction levels as defined in Figure 4.3.

The table cells (Table 5.1) that are highlighted in a different color depict that the modeling element has its prioritization changed. Observing the feasible configurations from prioritization P_3 and P_4, they indicate that the change of prioritization for *goals* did not affect the results suggested by ILP solver. We can also notice the same by verifying the Table 5.2 in P_3 and P_4. For both prioritization, the ILP solver suggested F_1, F_2, F_3,

Table 5.1: An example of trade-off analysis. The columns highlighted in a different color depict that the modeling element has its prioritization changed.

<table>
<tr><th colspan="3">Change of Prioritization for Soft goals</th></tr>
<tr><td rowspan="3">P_1</td><td>$sg_2 > sg_3 > sg_1$</td><td>$F_1 = f_0, f_1, f_2, \neg f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$</td></tr>
<tr><td>$c_2 > c_6$</td><td>$F_3 = f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$</td></tr>
<tr><td>$g_2 > g_3 > g_1$</td><td>$F_4 = f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, \neg f_8, f_9, f_{10}$</td></tr>
<tr><td rowspan="3">P_2</td><td>$sg_1 > sg_2 > sg_3$</td><td>$F_1 = f_0, f_1, f_2, \neg f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$</td></tr>
<tr><td>$c_2 > c_6$</td><td>$F_2 = f_0, f_1, f_2, \neg f_3, f_4, f_5, \neg f_6, f_7, \neg f_8, f_9, f_{10}$</td></tr>
<tr><td>$g_2 > g_3 > g_1$</td><td>$F_3 = f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$</td></tr>
<tr><th colspan="3">Change of Prioritization for Goals</th></tr>
<tr><td rowspan="3">P_3</td><td>$sg_3 > sg_2 > sg_1$</td><td>$F_1 = f_0, f_1, f_2, \neg f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$</td></tr>
<tr><td>$c_6 > c_2$</td><td>$F_2 = f_0, f_1, f_2, \neg f_3, f_4, f_5, \neg f_6, f_7, \neg f_8, f_9, f_{10}$</td></tr>
<tr><td>$g_3 > g_2 > g_1$</td><td>$F_3 = f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$
$F_4 = f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, \neg f_8, f_9, f_{10}$</td></tr>
<tr><td rowspan="3">P_4</td><td>$sg_3 > sg_2 > sg_1$</td><td>$F_1 = f_0, f_1, f_2, \neg f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$</td></tr>
<tr><td>$c_6 > c_2$</td><td>$F_2 = f_0, f_1, f_2, \neg f_3, f_4, f_5, \neg f_6, f_7, \neg f_8, f_9, f_{10}$</td></tr>
<tr><td>$g_1 > g_3 > g_2$</td><td>$F_3 = f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$
$F_4 = f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, \neg f_8, f_9, f_{10}$</td></tr>
<tr><th colspan="3">Change of Prioritization for Contexts</th></tr>
<tr><td rowspan="3">P_5</td><td>$sg_2 > sg_3 > sg_1$</td><td rowspan="3">$F_3 = f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$
$F_4 = f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, \neg f_8, f_9, f_{10}$</td></tr>
<tr><td>$c_2 > c_6$</td></tr>
<tr><td>$g_3 > g_1 > g_2$</td></tr>
<tr><td rowspan="3">P_6</td><td>$sg_2 > sg_3 > sg_1$</td><td>$F_1 = f_0, f_1, f_2, \neg f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$</td></tr>
<tr><td>$c_6 > c_2$</td><td>$F_3 = f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$</td></tr>
<tr><td>$g_3 > g_1 > g_2$</td><td>$F_4 = f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, \neg f_8, f_9, f_{10}$</td></tr>
</table>

and F_4 as feasible configurations.

In contrast, when we changed the prioritization for soft goals and contexts, the ILP solver suggested different configurations. In P_1, which the prioritization of soft goals is $sg_2 > sg_3 > sg_1$, the ILP solver suggested F_1, F_3, and F_4 as feasible configurations. Table 5.2 depicts that configuration F_1 satisfies ccf_5, whereas configuration F_3 satisfies ccf_1 and ccf_6, and configuration F_4 satisfies ccf_2, ccf_3, and ccf_4.

By changing the prioritization of soft goals to $sg_1 > sg_2 > sg_3$ (P_2), the ILP solver suggested F_1, F_2, and F_3 as feasible configurations and accordingly the CCFs that such configurations satisfy are also different. In P_2, configuration F_1 satisfies ccf_1, ccf_3, ccf_5, and ccf_6, configuration F_2 satisfies ccf_4, and configuration F_3 satisfies ccf_2. It means that soft goals can potentially affect the activation and deactivation of the system's features accordingly with their contribution values.

The same is true for contexts since are dependent on the relationships `require` and `exclude`, as well as, specification of CCFs. Such relationships and CCFs compose the adaptation rules that effect on the activation and deactivation of the system's features. In the running example, the ILP solver suggested different configurations when we changed the prioritization of contexts.

In P_5, which the prioritization of contexts is $c_2 > c_6$, the ILP solver suggested F_3 and

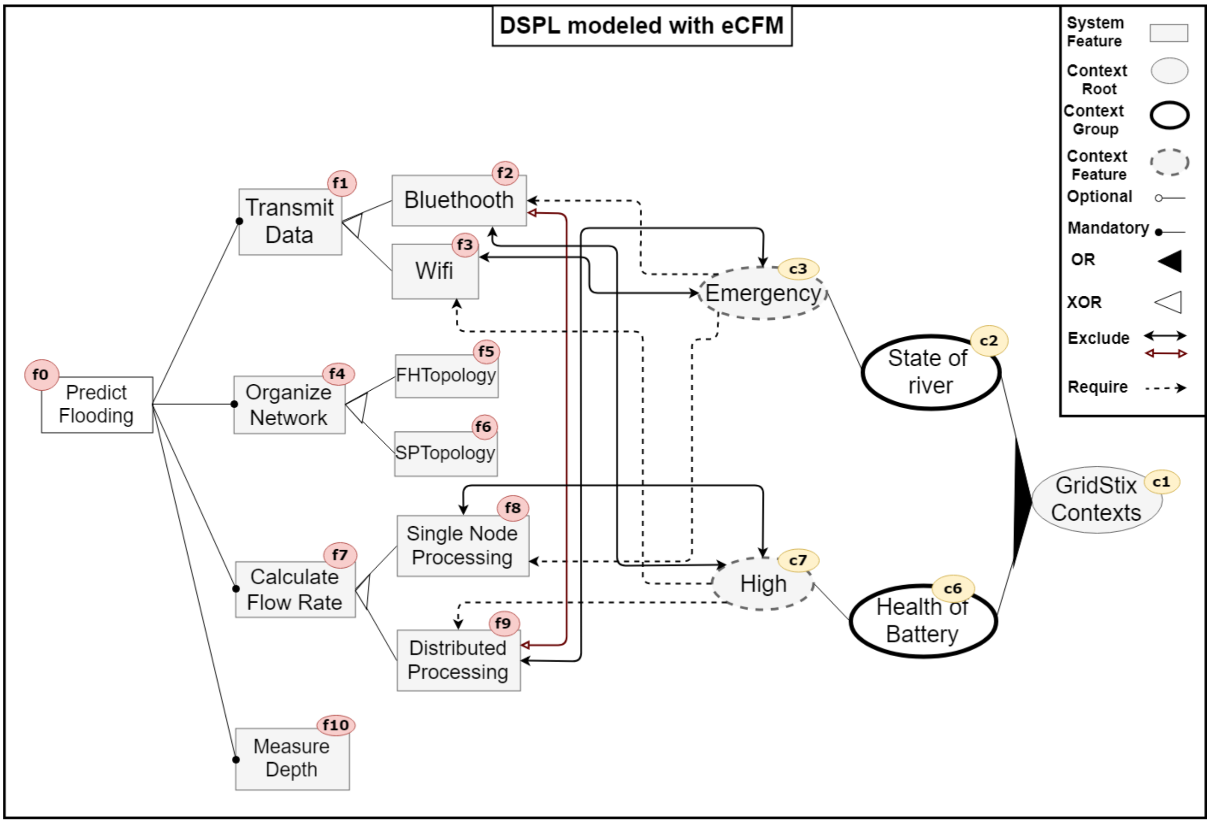
Table 5.2: Corresponding ccf for each feasible configuration suggested by solver ILP during the simulations.

Change of Prioritization for Soft goals		
P_1	$ccf_1 = < c_3,c_7 >$	F_3
	$ccf_2 = < c_4,c_7 >$	F_4
	$ccf_3 = < c_5,c_7 >$	F_4
	$ccf_4 = < c_3,c_8 >$	F_4
	$ccf_5 = < c_4,c_8 >$	F_1
	$ccf_6 = < c_5,c_8 >$	F_3
P_2	$ccf_1 = < c_3,c_7 >$	F_1
	$ccf_2 = < c_4,c_7 >$	F_3
	$ccf_3 = < c_5,c_7 >$	F_1
	$ccf_4 = < c_3,c_8 >$	F_2
	$ccf_5 = < c_4,c_8 >$	F_1
	$ccf_6 = < c_5,c_8 >$	F_1
Change of Prioritization for Goals		
P_3	$ccf_1 = < c_3,c_7 >$	F_4
	$ccf_2 = < c_4,c_7 >$	F_4
	$ccf_3 = < c_5,c_7 >$	F_2
	$ccf_4 = < c_3,c_8 >$	F_4
	$ccf_5 = < c_4,c_8 >$	F_1
	$ccf_6 = < c_5,c_8 >$	F_3
P_4	$ccf_1 = < c_3,c_7 >$	F_4
	$ccf_2 = < c_4,c_7 >$	F_4
	$ccf_3 = < c_5,c_7 >$	F_2
	$ccf_4 = < c_3,c_8 >$	F_4
	$ccf_5 = < c_4,c_8 >$	F_1
	$ccf_6 = < c_5,c_8 >$	F_3
Change of Prioritization for Contexts		
P_5	$ccf_1 = < c_3,c_7 >$	F_3
	$ccf_2 = < c_4,c_7 >$	F_4
	$ccf_3 = < c_5,c_7 >$	F_4
	$ccf_4 = < c_3,c_8 >$	F_4
	$ccf_5 = < c_4,c_8 >$	F_3
	$ccf_6 = < c_5,c_8 >$	F_3
P_6	$ccf_1 = < c_3,c_7 >$	F_4
	$ccf_2 = < c_4,c_7 >$	F_4
	$ccf_3 = < c_5,c_7 >$	F_3
	$ccf_4 = < c_3,c_8 >$	F_4
	$ccf_5 = < c_4,c_8 >$	F_1
	$ccf_6 = < c_5,c_8 >$	F_3

F_4 as feasible configurations. Table 5.2 depicts that configuration F_3 satisfies ccf_1, ccf_5, and ccf_6, whereas configuration F_4 satisfies ccf_2, ccf_3, and ccf_4. In return, by changing the prioritization of contexts to $c_6 > c_2$, the ILP solver suggested F_1, F_3, and F_4 as feasible configurations. In this case, configuration F_1 satisfies ccf_5, configuration F_3 satisfies ccf_3

and ccf_6, and configuration F_4 satisfies ccf_1, ccf_2, and ccf_4.

Figure 5.1: An example of CCF for GridStix DAS: ccf_1 (`Emergency` and `High`)

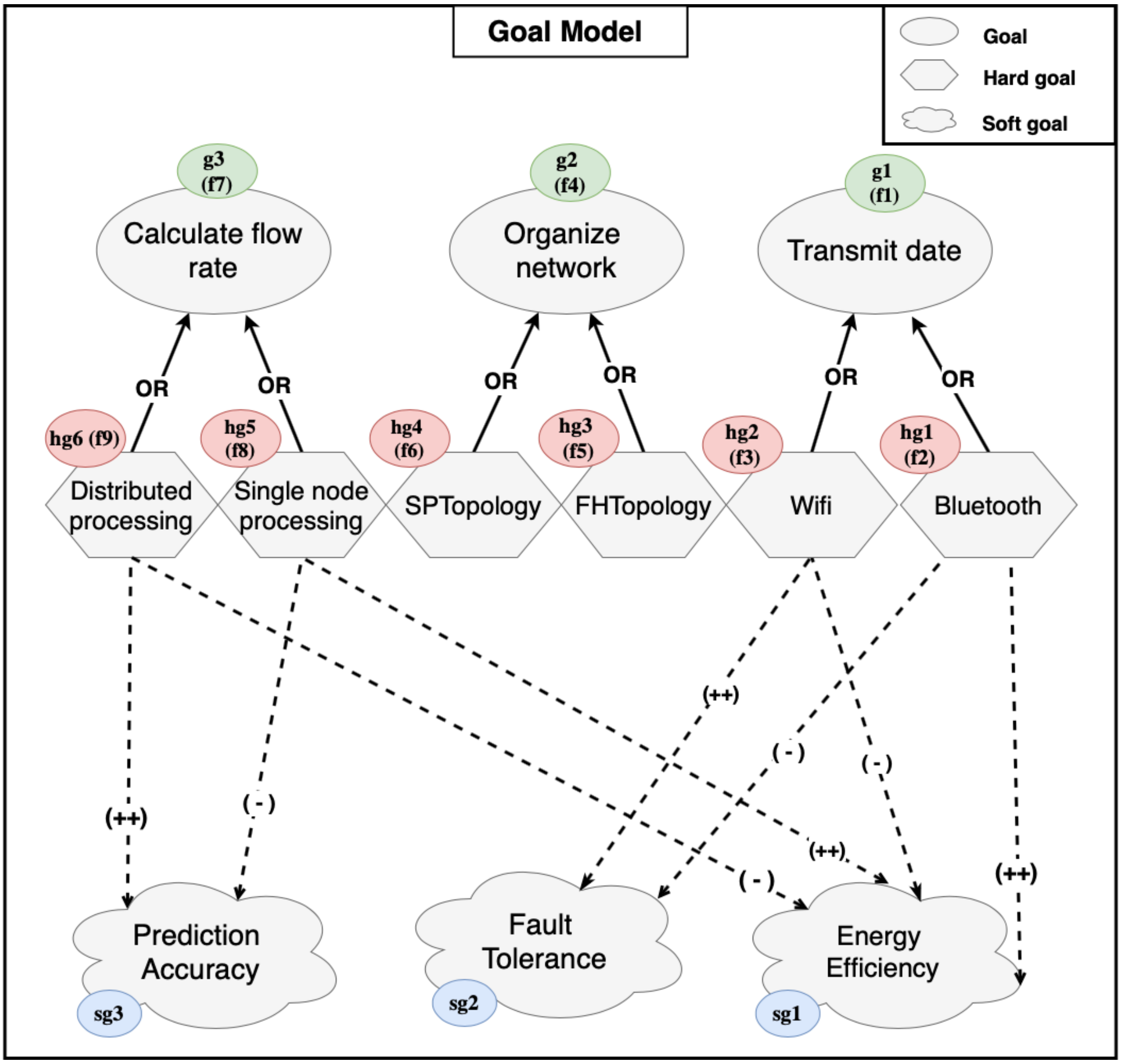


5.2.2 Context feature change

We also designed simulations related to changes of CCFs. Each CCF must be based on the relationship between **context feature** and its respective **context group**. In the running example, we selected only one **context feature** from each **context group** due to their `XOR`-group relationship. As a result, we obtained a total of six CCFs, *i.e.*, $3 * 2$ possible context feature changes (ccf_1–ccf_6), as shown in Table 5.3. Therefore, we simulated six scenarios, which were based on the different CCFs and performed the configuration process in order to find feasible configurations that meet all constraints. All scenarios correspond to the same prioritization of contexts, goals, and soft goals presented in Section 4.2.3 (contexts: $c_2 > c_6$; goals: $g_2 > g_1 > g_3$; soft goals: $sg_1 > sg_2 > sg_3$). Additionally, only the `require` and `exclude` relationships of CCF under evaluation should be kept. In other words, the software engineer must only consider the relationships between **context features** and system' features that correspond with a certain CCF.

The ILP solver suggested, for instance, that feasible configuration F_1 (see Table 5.4) meets ccf_1 (see Figure 5.1). It considered the following `require` and `exclude` relationships: *(i)* context feature `Emergency` with features **Bluetooth** and `Single Node`

Figure 5.2: Goal model corresponding to CCF ccf_1 (`Emergency` and `High`)



`Processing`; *(ii)* context feature **High** with features **Wifi** and **Distributed Processing**; *(iii)* context feature **Emergency** with features **Wifi** and **Distributed Processing**; and *(iv)* context feature **High** with features **Bluetooth** and **Single Node Processing**. This reasoning is used for all CCFs in order to find the feasible configurations that satisfy them.

Likewise, the software engineer must only consider the relationships between hard goals and soft goals correspond with a certain CCF. Figure 5.2 shows relationships that were considered, as follows: *(i)* hard goal **Bluetooth** with soft goals **Energy Efficiency** and **FaultTolerance**; *(ii)* hard goal **Wifi** with soft goals **Energy Efficiency** and **FaultTolerance**; *(iii)* hard goal **Single node processing** with soft goals **Energy Efficiency** and **Prediction Accurancy**; *(iv)* hard goal **Distributed processing** with soft goals **Energy Efficiency** and **Prediction Accurancy**;

Table 5.3: Its presents the `Require` and `exclude` relationships for the valid $ccfs$

CCF	Relationship between context feature and system' feature		Feasible Configuration
	Require	Exclude	
$ccf_1 =$ $< c_3, c_7 >$	$< c_3, f_2 >$	$< c_3, f_3 >$	F_1
	$< c_3, f_8 >$	$< c_3, f_9 >$	
	$< c_7, f_3 >$	$< c_7, f_2 >$	
	$< c_7, f_9 >$	$< c_7, f_8 >$	
$ccf_2 =$ $< c_4, c_7 >$	$< c_4, f_9 >$	$< c_4, f_8 >$	F_1
	$< c_7, f_3 >$	$< c_7, f_2 >$	
	$< c_7, f_9 >$	$< c_7, f_8 >$	
$ccf_3 =$ $< c_5, c_7 >$	$< c_5, f_3 >$	$< c_5, f_2 >$	F_2
	$< c_5, f_5 >$	$< c_5, f_6 >$	
	$< c_7, f_3 >$	$< c_7, f_2 >$	
	$< c_7, f_9 >$	$< c_7, f_8 >$	
$ccf_4 =$ $< c_3, c_8 >$	$< c_3, f_2 >$	$< c_3, f_3 >$	F_1
	$< c_3, f_8 >$	$< c_3, f_9 >$	
	$< c_8, f_5 >$	$< c_8, f_3 >$	
	$< c_8, f_8 >$	$< c_8, f_9 >$	
$ccf_5 =$ $< c_4, c_8 >$	$< c_4, f_9 >$	$< c_4, f_8 >$	F_3
	$< c_8, f_5 >$	$< c_8, f_3 >$	
	$< c_8, f_8 >$	$< c_8, f_9 >$	
$ccf_6 =$ $< c_5, c_8 >$	$< c_5, f_3 >$	$< c_5, f_2 >$	F_1
	$< c_5, f_5 >$	$< c_5, f_6 >$	
	$< c_8, f_5 >$	$< c_8, f_3 >$	
	$< c_8, f_8 >$	$< c_8, f_9 >$	

Table 5.4: Feasible configurations suggested by solver ILP

Configuration	System' features
F_1	$f_0, f_1, f_2, \neg f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$
F_2	$f_0, f_1, f_2, \neg f_3, f_4, f_5, \neg f_6, f_7, \neg f_8, f_9, f_{10}$
F_3	$f_0, f_1, \neg f_2, f_3, f_4, f_5, \neg f_6, f_7, f_8, \neg f_9, f_{10}$

In this analysis, the ILP solver suggested three different feasible configurations (F_1 – F_3), as shown in Table 5.4. Each feasible configuration, derived from *GridStix* DAS, is composed by a set of system' features. Table 5.5 presents the system' features with their respective utility values. Those three feasible configurations correspond the CCFs ccf_1, ccf_3, and ccf_5, respectively.

For the CCF ccf_1, the ILP solver suggested F_1 as feasible configuration. In this configuration, features f_2, f_5, and f_8 were selected. By observing the utility values, they

Table 5.5: Utility values of the system' features

	Features					
Configuration	f_2	f_3	f_5	f_6	f_8	f_9
F_1 **for** ccf_1	1.169	0.277	0	0	1.080	-0.043
F_2 **for** ccf_3	0.003	0.849	1.130	-0.130	0.496	0.372
F_3 **for** ccf_5	1.336	-0.150	0	0	0.746	0.289

indicate that features f_3 and f_9 have the lowest values $\mathbb{C}(f_i)$. However, although the features f_5 and f_6 have presented equal values $\mathbb{C}(f_i)$, the solver suggested that feature f_5 should be selected. It reveals that the solver suggests the first one among all features with equal values. The same occurred when configuration F_3 was considered as feasible for the CCF ccf_5, which corresponds to activation of features f_2, f_5, and f_8. For the CCF ccf_3, the configuration F_2 was suggested as feasible. In this case, features f_3, f_5, and f_8 were selected, since they presented greater utility values than feature f_2, f_6, and f_9.

5.2.3 Definition of adaptation model

From a specific CCF, it is possible to define dynamic adaptation models. These models show how a DAS can evolve from one CCF to another changing its respective feasible configuration. Figure 5.3 shows an example of a DAS adaptation model composed of three feasible configurations, which can be loaded by six different CCFs (ccf_1–ccf_6).

In this example, we chose arbitrarily the configuration F_1 as initial due to its recurrence in the majority of simulations. However, the software engineer can choose another one. It is possible to implement eight adaptations by considering the CCF changes. They can be detected on the runtime environment, as follows: *(i)* when the CCFs ccf_1, ccf_2, ccf_4, and ccf_6 are detected, the configuration F_1 is loaded; *(ii)* the configuration F_2 is loaded, when CCF ccf_3 is detected; and *(iii)* when CCF ccf_5 is detected, configuration F_3 is loaded. Therefore, this process should be made for all CCF changes and their respective feasible configurations.

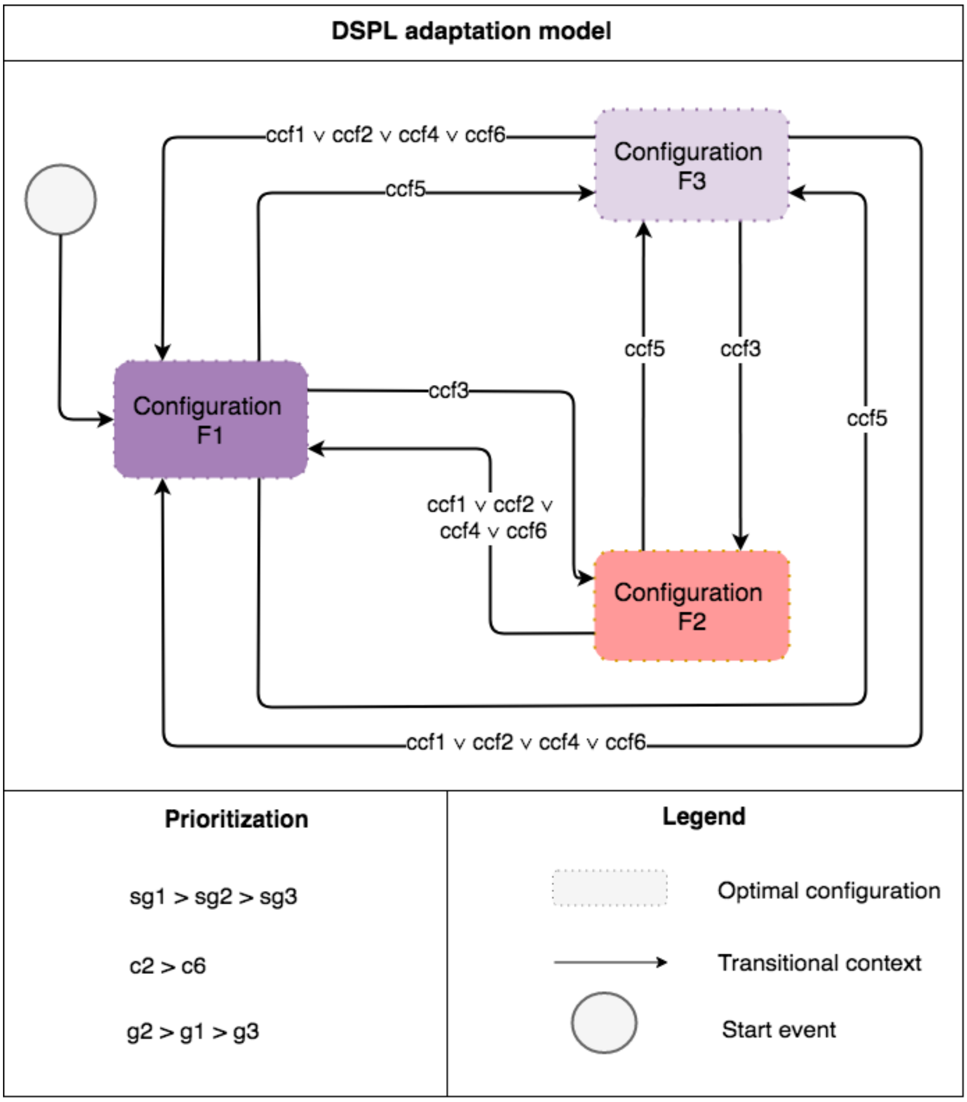
Each adaptation model encompasses a specific prioritization of *goals*, *soft goals*, and *contexts*. Then, an adaptation model can be considered as a product in DSPL engineering. In addition, the variability decisions are made at runtime by meeting the CCF changes. Thus, it provides support for the entire range of adaptations handled during application engineering.

Indeed, adaptation models offer the potential to facilitate the communication between software engineers and developers. Bencomo *et al.* [7] proposed the development of similar adaptation models to represent the transitions among contextual changes. These assets facilitate the development of DSPL applications and the understanding of how they can behave from a certain context change.

5.3 EVALUATION OF THE TOFFA-DAS APPROACH

We used a set of criteria to characterize eleven existing approaches, which were employed in the configuration selection process of DAS by considering the trade-off between con-

Figure 5.3: An example of adaptation model for a DAS application. It is defined after context feature change analysis. Each configuration represents a set of features suggested as valid and feasible during the configuration selection process.



textual information and NFRs (Section 5.1). Based on such characterization, depicted in Table 4.1, we selected only one approach for the evaluation, named *ConG4DaS*. Such an approach presented by Guedes *et al.* [107] was well-ranked since it *(i)* consider the trade-off between contexts and NFRs; *(ii)* uses the *Utility-based* planning as a strategy; and *(iii)* takes into account the combination of both modeling characteristics, *satisfaction level* and *prioritization*.

Although, *ConG4DaS* has demonstrated the likely similarity with our approach, it does not consider constraints among contexts. In addition, it does not handle contexts and NFRs in an independent way. In this sense, we decided to compare it with our approach in order to identify which one provide configurations with higher satisfaction levels of *soft goals*. In the next sections, we present the exploratory study that was organized following the guide proposed by Wohlin *et al.* [119].

5.3.1 Exploratory study definition

NFRs (*soft goals*) are strictly related to the information provided by the system's features (*hard goals*) and typically set constraints for them [120]. Since both approaches consider *soft goal* priority in their configuration process of DAS, only the satisfaction level between *hard goals* and *soft goals* was used as a criterion for the assessment presented in this study.

Thus, the study aimed at **analyzing** the ToffA-DAS and ConG4DaS approaches **for the purpose of** evaluating the resulting configurations obtained from the configuration process **with respect to** the overall satisfaction level between *hard goals* and *soft goals* **from the point of view of** Software Engineers and Researchers **in the context of** two DAS. Based on the study's goal, we defined the following research question for this assessment:

> *RQ.* **Do the configurations generated by the ToffA-DAS approach provide higher satisfaction levels of soft goals than those generated by the ConG4DaS approach?**

We employed in this study the same raw data presented by Guedes *et.al.* [107]. Thus, to make a fair comparison between ConG4DaS and ToffA-DAS, we applied in this evaluation the same metrics used by them, which are based on the negative and positive contributions that influence the satisfaction level of a *soft goal*. The metrics used are presented as follows:

- **Pos** - This metric calculates the number of positive contributions to the *soft goals* with the highest priority;
- **Neg** - This metric calculates the number of negative contributions to the *soft goals* with the highest priority; and
- **Diff** - This metric calculates the difference between the number of positive and negative contributions to the *soft goals* with the highest priority (Pos-Neg). It aims to identify whether the release presented more positive or negative contributions to the highest priority of *soft goal*.

5.3.2 Exploratory study planning

This section discusses the planning and the procedures to be followed in order to perform the exploratory study. For this study, we selected two DAS presented in the literature named *Mobile game* [2] and *Smart Home* [121], respectively. Such DAS applications were also used in the evaluation presented by Guedes *et al.* [107], which was the basis for identifying the scenarios and metrics to be employed in the evaluation of this paper (see artifacts of the exploratory study in appendix B). The following subsections present the procedures used and the hypothesis defined.

5.3.2.1 Quantitative analysis mechanisms - The exploratory study followed the activities presented in Figure 4.1: *(i)* identification of domain knowledge; *(ii)* modeling of both DAS; *(iii)* execution of the model checking to verify the eCFM models; *(iv)* definition of the prioritization of *goals*, *soft goals*, and *contexts*; *(v)* measurement of the impact of features over *goals*, *soft goals*, and *contexts*; and *(vi)* definition of the feasible configurations. In the latter activity, we collected the metrics **Pos**, **Neg**, and **Diff**.

Aiming to perform the simulations, we executed the configuration selection process presented in Section 4.3. For each simulation, software engineers must only consider relationships between the systems feature and context corresponding to a specific combination of context features (CCF). Our study is based on the same CCFs and *prioritization* defined by Guedes *et al.* [107] (see Table 5.6 and Table 5.7). The metrics of the evaluation are described in terms of mapping link between *hard goals* and *soft goals*: satisfied (++) = 1, weakly satisfied (+) = 0.5, undecided (?) = 0, weakly denied (-) = -0.5, and denied (- -) = -1. We collected the results of the metrics based on the number of positive and negative satisfaction levels of *soft goals* over *hard goals* for all configurations obtained in execution of the both approaches, ConG4DaS and ToffA-DAS.

For quantitative data, the analysis included descriptive statistics, such as *mean values* and *boxplot* aiming to explore the gathered data. Regarding the hypotheses defined for the exploratory study, the non-parametric *Wilcoxon Signed-rank Test* was used [122, 123]. This test was chosen because the study employs two related samples and it yields difference scores that may be ranked in order of absolute size. Indeed, it determines which of the measures in pair is the greatest and ranks the differences. It also gives more weight to a pair which shows a large difference between the two conditions than to a pair that shows a small difference. In addition, it shows the sign of the difference between any pair and ranks the differences in the order of absolute size [124].

5.3.2.2 Hypothesis - Null Hypotheses. The null hypotheses state that there is no difference between ToffA-DAS and ConG4Das in terms of Pos, Neg and Diff. In accordance with the *Wilcoxon Test*, the sum of the positive ranks is equal to the sum of the negative ranks. The corresponding null hypotheses are presented as follows:

- $H_{01} : Pos_{toffa} = Pos_{con}$
- $H_{02} : Neg_{toffa} = Neg_{con}$
- $H_{03} : Diff_{toffa} = Diff_{con}$

Alternative Hypotheses. The alternative hypotheses state that there is a difference between ToffA-DAS and ConG4Das in terms of Pos, Neg, and Diff. In accordance with the *Wilcoxon Test*, the sum of the positive ranks is different of the sum of the negative ranks. The corresponding alternative hypotheses are presented as follows:

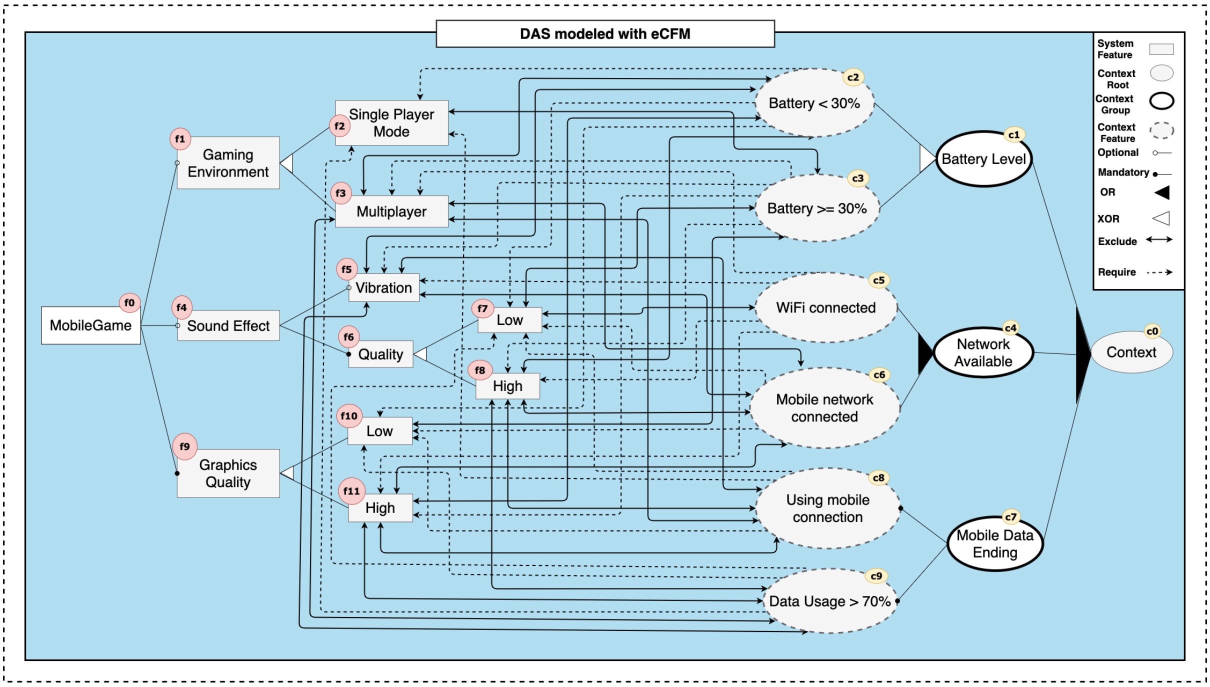
- $H_{11} : Pos_{toffa} \neq Pos_{con}$
- $H_{12} : Neg_{toffa} \neq Neg_{con}$
- $H_{13} : Diff_{toffa} \neq Diff_{con}$

5.3.3 Analysis and interpretation

This section provides an in-depth analysis of the gathered data. We present the results in terms of satisfaction level of *soft goals* for the configuration process of both ToffA-DAS and ConG4DaS approaches. Moreover, it discusses hypothesis testing.

5.3.3.1 Mobile Game DAS - From the models and information presented by Pascual *et al.* [2], we designed the corresponding eCFM and goal model. The eCFM is composed of twelve features and ten contexts. The goal model one is composed of four *goals*, seven *hard goals*, and two *soft goals*. Figures 5.4 and 5.5 present the eCFM and goal model of the Mobile Game, respectively.

Figure 5.4: eCFM for Mobile Game DAS



Once the models were finished, we simulated eight scenarios corresponding to CCFs ccf_1, ccf_3, ccf_7, ccf_8, ccf_9, ccf_{11}, ccf_{15}, and ccf_{16}. For all of them, we kept the same priority for *goals*, *i.e.*, all elements have priority equal to one and considered for *soft goals* and *contexts*, the same priority presented by Guedes *et al.* [107]. Table 5.6 shows the resulting feasible configurations from these simulations. For instance, the ILP solver suggested that the variable features f_3, f_5, f_7 and f_{10} satisfy ccf_7. For this scenario, the prioritization of *soft goals* is equal to one and the prioritization of *contexts* is $c_7 > c_1$.

Observing the results, we identified that both approaches presented only one equivalent configuration, which corresponds to ccf_7. In the scenarios corresponding to CCFs ccf_8, ccf_9, ccf_{11}, ccf_{15}, and ccf_{16}, for instance, the ILP solver in ToffA-DAS, suggested features f_2, f_5, f_7, and f_{10} to be inserted in the feasible configuration. In the same sce-

Figure 5.5: Goal Model for Mobile Game DAS

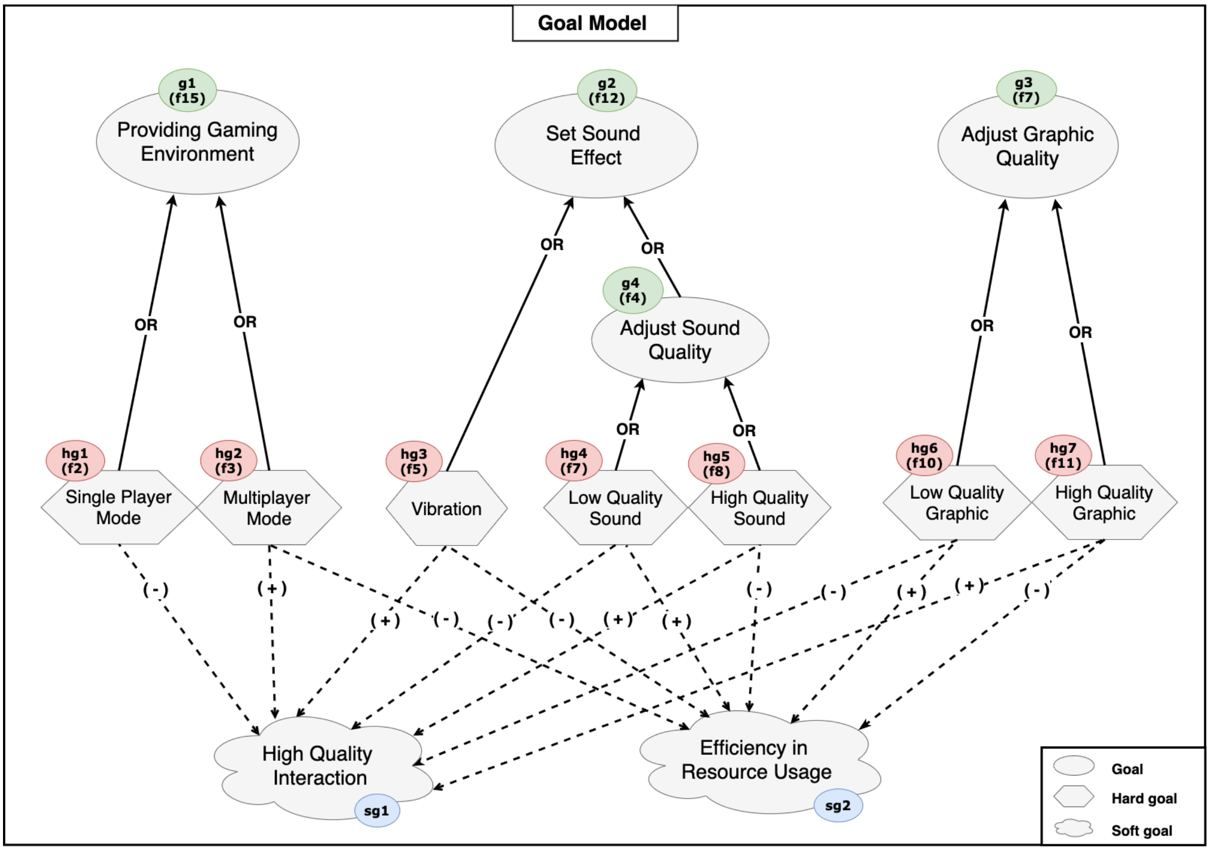


Table 5.6: Scenarios for Mobile Game DAS by considering the valid $CCFs$ and prioritization of soft goals presented in the first and fourth columns of the table. After executing the simulations, we collected the metrics Pos, Neg, and Diff for each configuration suggested in second and third columns.

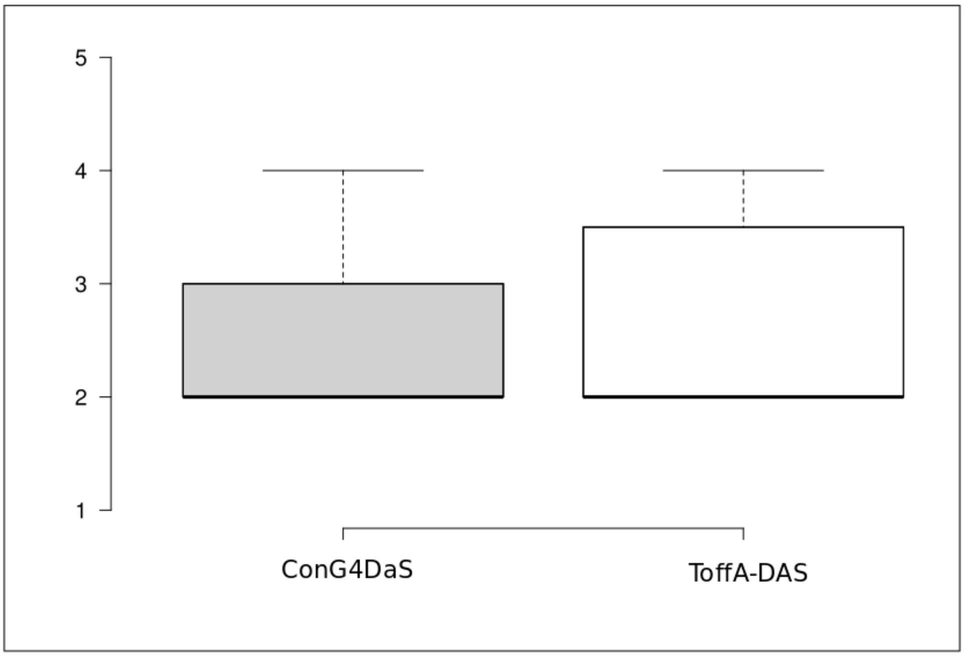
Results about configurations (Variable features)			Results about metrics						
CCFs	ConG4DaS	ToffA-DAS	Soft goal Priority	ConG4DaS			ToffA-DAS		
				Pos	Neg	Diff	Pos	Neg	Diff
$ccf_1 = \{none\}$	f_2,f_5,f_8,f_{11}	f_3,f_5,f_7,f_{11}	$sg_1 > sg_2$	3	1	2	3	1	2
$ccf_3 = \{c_5, c_6\}$	f_2,f_5,f_8,f_{11}	f_3,f_5,f_8,f_{11}	$sg_1 > sg_2$	3	1	2	4	0	0
$ccf_7 = \{c_5, c_6, c_8\}$	f_3,f_5,f_7,f_{10}	f_3,f_5,f_7,f_{10}	$sg_1 = sg_2 = 1$	4	4	0	4	4	0
$ccf_8 = \{c_5, c_6, c_8, c_9\}$	f_2,f_7,f_{10}	f_2,f_5,f_7,f_{10}	$sg_2 > sg_1$	2	0	2	2	1	1
$ccf_9 = \{c_2\}$	f_2,f_7,f_{10}	f_2,f_5,f_7,f_{10}	$sg_2 > sg_1$	2	0	2	2	1	1
$ccf_{11} = \{c_2, c_5, c_6\}$	f_3,f_7,f_{10}	f_2,f_5,f_7,f_{10}	$sg_2 > sg_1$	2	1	1	2	1	1
$ccf_{15} = \{c_2, c_5, c_6\}$	f_3,f_7,f_{10}	f_2,f_5,f_7,f_{10}	$sg_2 > sg_1$	2	1	1	2	1	1
$ccf_{16} = \{c_2, c_3, c_5, c_6, c_8, c_9\}$	f_2,f_7,f_{10}	f_2,f_5,f_7,f_{10}	$sg_2 > sg_1$	2	0	2	2	1	1

narios, the ConG4DaS approach presented different results, as follows: *(i)* features f_2, f_7, and f_{10} were suggested for ccf_8, ccf_9, ccf_{16}; and *(ii)* features f_3, f_7, and f_{10} were suggested for ccf_{11}, ccf_{15}. We concluded that is due to the way in which our approach deals with the priority of *soft goals* and *contexts*. When considering the contribution value for features over *soft goals* and *contexts* individually, it changes the *utility values* in comparison with ConG4DaS results.

We also measured the number of positive and negative satisfaction levels of *soft goals* over *hard goals* for all configurations obtained in each approach. Table 5.6 shows the results concerning the **Pos**, **Neg**, and **Diff** measures. We applied the *Wilcoxon Test* to assess the null hypothesis presented in Section 6.3.2.2. When using such a test, pairs of data that have a score difference equal to zero are removed from the analysis. Then, the number of pairs N to be considered is equal to the total number of pairs minus any pairs whose difference is zero.

We removed from the analysis, the pair POS_{toffa} and POS_{toffa+} that have a score difference equal to zero. As a result, we obtained $T^{+} = 0$ and $T^{-} = -8$. However, the sample with $N = 1$ is not enough to return the critical value of $W_{critical}$ with the level of significance *p-value* ≤ 0.05. In this way, the hypothesis null H_{01} cannot be rejected. It means that there is no statistically significant difference between POS_{toffa} and POS_{con} for the Mobile Game DAS. Figure 5.6 depicts the box plot concerning the **Pos** metric. In general, both approaches, ConG4DaS and ToffA-DAS presented the **Pos** *median* value equal to 2. However, the variance of the data set to the ToffA-DAS was higher than in ConG4DaS. It possible to see, from Table 5.6, that ToffA-DAS had a greater number of positive contributions to the *soft goals* with the highest priority.

Figure 5.6: Pos Metric - Mobile Game



Regarding the **Neg** metric, we removed the pairs Neg_{toffa} and Neg_{con} that have a score difference equal to zero. As a result, we obtained $T^{+} = 6.5$ and $T^{-} = -19.5$. However, the sample with $N = 4$ is not enough to return the critical value $W_{critical}$ with the level of significance *p-value* ≤ 0.05. As a result, the hypothesis null H_{02} cannot be rejected. It means that there is no statistically significant difference between the Neg_{toffa}

Figure 5.7: Neg Metric - Mobile Game

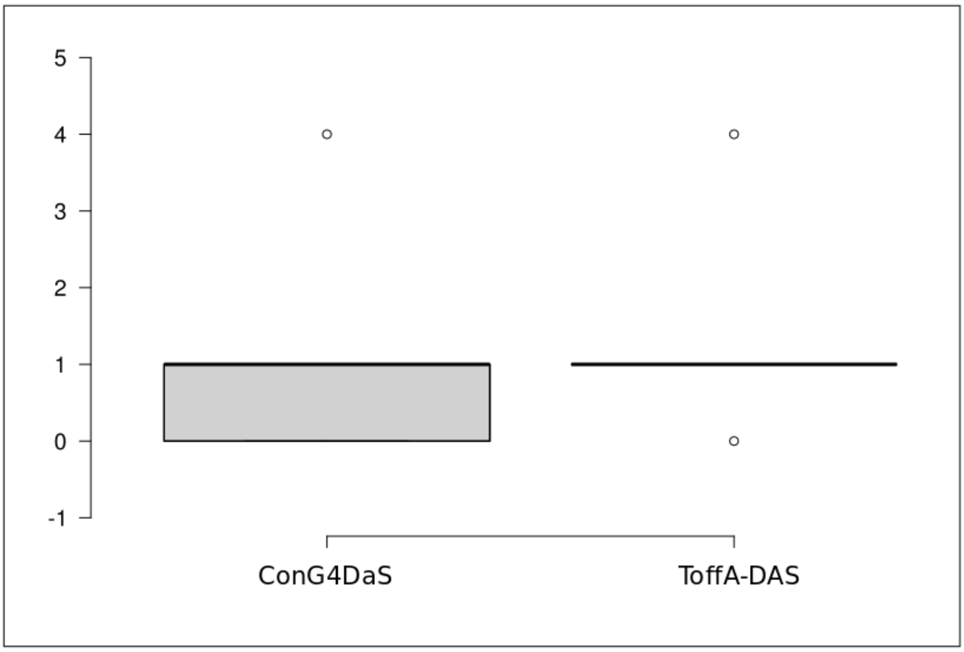
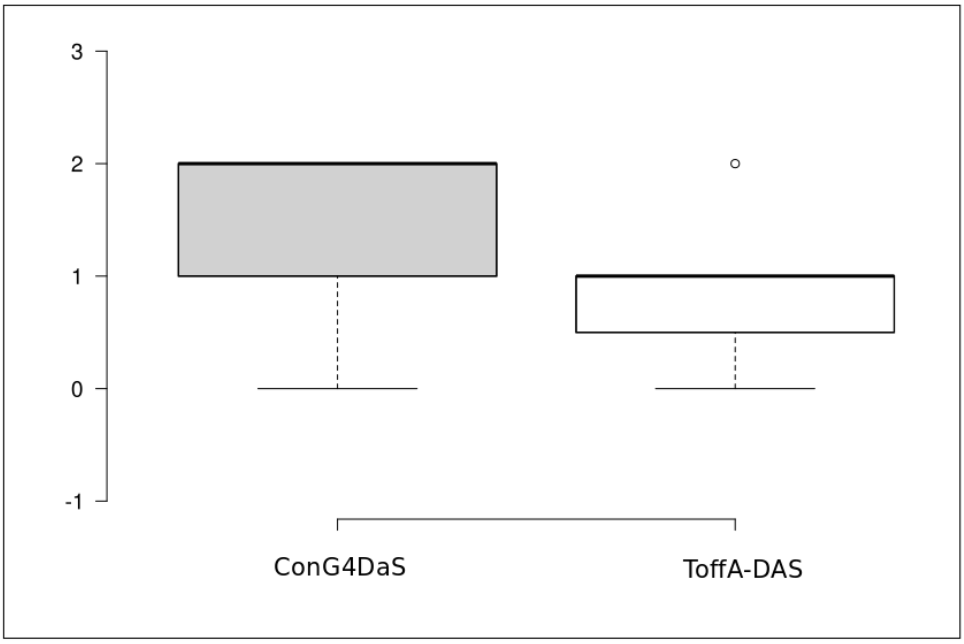


Figure 5.8: Diff Metric - Mobile Game



and Neg_{con} for the Mobile Game DAS. Figure 5.7 depicts the box plot concerning the **Neg** metric. For this measure in ToffA-DAS, the *median* value was equal to 1 and there was no variance of the data set. It may also be observed two outliers, which came from scenarios ccf_7 and ccf_3. It indicates that such scenarios had a greater number of negative contributions to the *soft goals* with the highest priority. In contrast, in the

ConG4DaS, we can see an outlier that came from scenario ccf_7. In addition, this approach presented a greater variance in the data set. We could associate this sharp variance to the scenarios, such as ccf_1, ccf_3, ccf_{11}, and ccf_{15}, that presented a greater number of negative contributions to the *soft goals* with highest priority.

Finally, for the **Diff** metric, after removing the pairs $Diff_{toffa}$ and $Diff_{con}$ that have a score difference equal to zero, we obtained $T^+ = 26$ and $T^- = 0$. The sample presented a value of $N = 4$, then it is not enough to return the critical value of $W_{critical}$ with the level of significance *p-value* ≤ 0.05. Therefore, the hypothesis null H_{03} cannot be rejected, meaning that there is no statistically significant difference between the $Diff_{toffa}$ and $Diff_{con}$ for the Mobile Game DAS. Figure 5.8 depicts the box plot concerning the **Diff** metric. For this metric, the *median* value in ToffA-DAS was equal to 1, whereas in GonG4DaS such value was equal to 2. In general, the variation of the data set to the second approach was higher than in the first one. It may also be observed an outlier in ToffA-DAS, which indicates a greater **Diff** value in the scenario ccf_1. In summary, the ToffA-DAS approach presented more positive contributions to the *soft goals* with the highest priority than the ConG4DaS approach, considering the scenarios defined for Mobile Game DSPL.

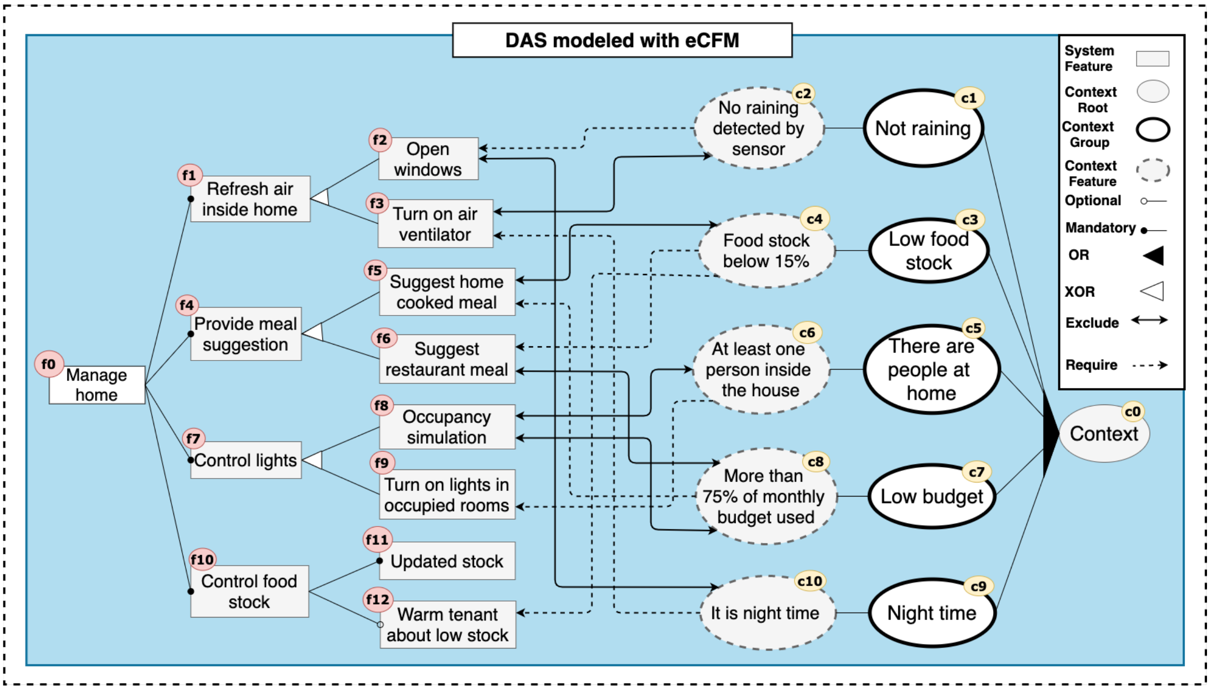
5.3.3.2 Smart Home DSPL - From the models and information presented by Pimentel *et.al.* [121], we designed the corresponding eCFM and goal model. The eCFM is composed of thirteen features and eleven contexts. The goal model one is composed of three *goals*, six *hard goals*, and three *soft goals*. Figures 5.9 and 5.10 present the eCFM and goal model of the Smart Home, respectively.

Once the models were finished, we simulated thirty-two scenarios corresponding to CCFs from ccf_1 to ccf_{32}. For all of them, we kept the same priority for *goals* and *contexts* (priority equal to one), besides considering for *soft goals*, the same priority presented by Guedes *et al.* [107]. Table 5.7 shows the resulting feasible configurations from these simulations. For instance, the ILP solver in ToffA-DAS suggested that the variable features are inserted into the configuration, as follows:

- Features f_3, f_5, and f_9 satisfy ccf_1, ccf_2, ccf_3, ccf_4, ccf_5, ccf_6, ccf_7, ccf_{16}, ccf_{18}, and ccf_{22};
- Features f_2, f_5, and f_9 satisfy ccf_{17}, ccf_{19}, ccf_{20}, ccf_{21}, ccf_{23}, and ccf_{24};
- Features f_3, f_6, f_9, and f_{12} satisfy ccf_9, ccf_{10}, ccf_{12}, ccf_{13}, ccf_{14}, ccf_{26}, and ccf_{30};
- Features f_2, f_6, f_9, and f_{12} satisfy ccf_{11}, ccf_{25}, ccf_{27}, ccf_{28}, ccf_{29}, ccf_{31}, and ccf_{32}.

Observing the results, we identified that both approaches presented twenty one equivalent configuration, which corresponds to CCFs ccf_3, ccf_4, ccf_5, ccf_6, ccf_7, ccf_{11}, ccf_{13}, ccf_{14}, ccf_{17}, ccf_{19}, ccf_{20}, ccf_{21}, ccf_{22}, ccf_{23}, ccf_{24}, ccf_{27}, ccf_{28}, ccf_{29}, ccf_{30}, ccf_{31}, and ccf_{32}. On the remainder scenarios the ConG4DaS approach suggested changes with regard to features f_2, f_5, and f_9. For instance, it is suggested that features f_3, f_6, f_8, and f_{12} to be inserted into the feasible configuration corresponding to scenario ccf_9. It is due

Figure 5.9: eCFM for Smart Home DAS

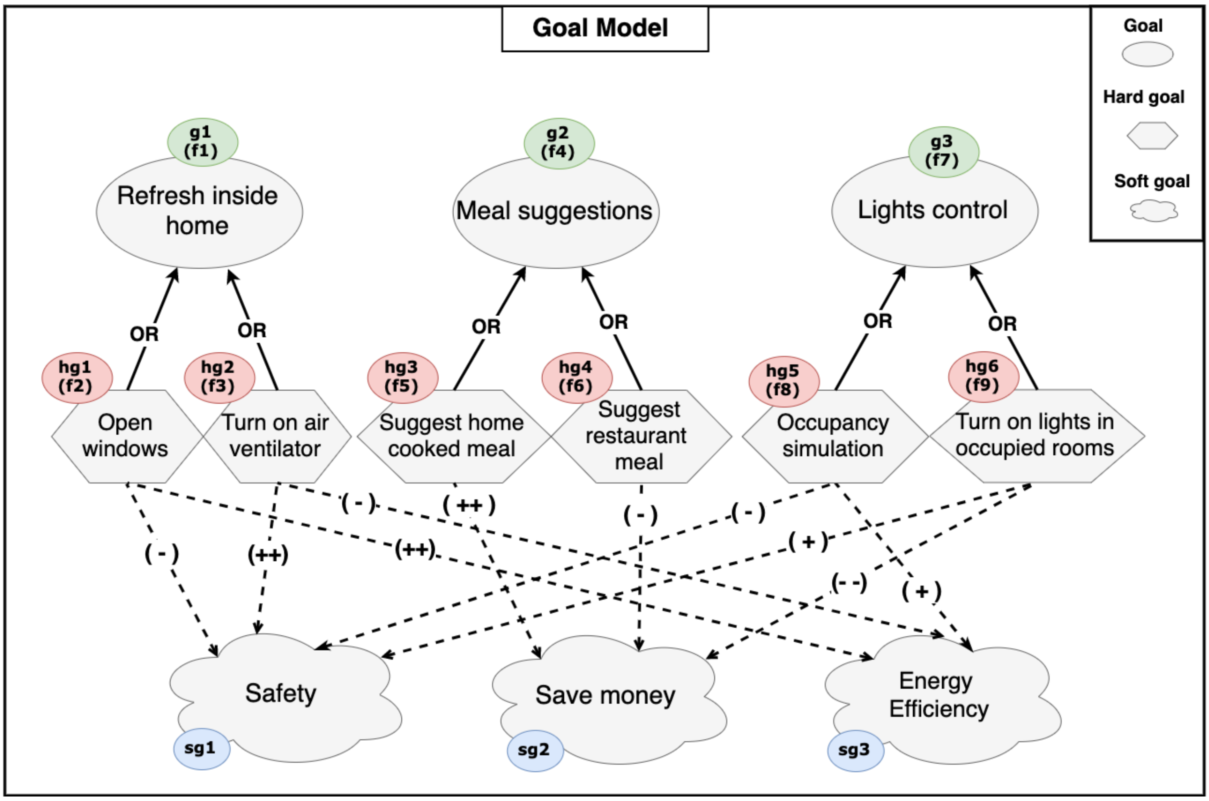


to the way in which our approach deals with the priority of *soft goals* and *contexts*. When considering the contribution value for features over *soft goals* and *contexts* individually, it changes the *utility values* in comparison with ConG4DaS results.

We also measured the number of positive and negative satisfaction levels of *soft goals* over *hard goals* for all configurations obtained in each approach. Table 5.7 shows the results concerning the **Pos**, **Neg**, and **Diff** measures. We applied the *Wilcoxon Test* to assess the null hypothesis presented in Section 6.3.2.2. When using such a test, pairs of data that have a score difference equal to zero are removed from the analysis. Then, the number of pairs N to be considered is equal to the total number of pairs minus any pairs whose difference is zero.

We removed from the analysis, the pairs POS_{toffa} and POS_{toffa+} that have a score difference equal to zero. As a result, we obtained $T^{+} = 229.5$ and $T^{-} = -127.5$. The sample presented a value of $N = 14$, the critical value $W_{critical}$ at *p-value* ≤ 0.05 equal to 21, and $W_{start} = 127.5$. Since $W_{start} > W_{critical}$, the null hypothesis H_{03} is rejected. It means that there is statistical significance difference between POS_{toffa} and POS_{con} for the Smart Home DSPL. Figure 5.11 depicts the box plot concerning the **Pos** metric. For this metric, the *median* value in ConG4DaS release was equal to 2, whereas in ToffA-DAS such value was equal to 1.5. Both approaches presented a similar variance of the data set. It means that the configurations resulting from the execution of Cong4DaS and ToffA-DAS were similar for most of the scenarios. In addition, some scenarios presented a **Pos** value equal to 3 in ConG4DaS as well as ToffA-DAS. It is possible to see, from Table 5.7, that ConG4DaS approach had a greater number of positive contributions to

Figure 5.10: Goal Model for Smart Home DAS



the *soft goals* with the highest priority.

Regarding the **Neg** metric, we also removed the pairs Neg_{toffa} and Neg_{con} that have a score difference equal to zero and obtained $T^{+} = 125$ and $T^{-} = -232$. The sample presented a value of $N = 14$, the critical value $W_{critical}$ at *p-value* ≤ 0.05 equal to 21, and $W_{start} = 125$. Since $W_{start} > W_{critical}$, the null hypothesis H_{03} is rejected. It means that there is statistical significance difference between Neg_{toffa} and Neg_{con} for the Smart Home DSPL. Figure 5.12 depicts the box plot concerning the **Neg** metric. For this metric, the *median* value was equal to 1 for both approaches. In general, the variation of the data set to the ToffA-DAS was higher than in the ConG4DaS. It may also be observed two outliers in ToffA-DAS, which indicates a greater **Neg** value in the scenarios ccf_{15}, ccf_{17}, ccf_{21}, ccf_{25}, and ccf_{29} and lower **Neg** value in the scenarios ccf_{19}, ccf_{20}, ccf_{25}, and ccf_{29}. In general, it indicates that the ConG4DaS approach had a greater number of negative contributions to the *soft goals* with the highest priority.

Finally, for the **Diff** metric, after removing the scenarios where the difference between $Diff_{toffa}$ and $Diff_{con}$ is zero, we obtained $T^{+} = 232$ and $T^{-} = -125$. The sample presented a value of $N = 14$, the critical value $W_{critical}$ at *p-value* ≤ 0.05 equal to 21, and $W_{start} = 125$. In this way, the hypothesis null H_{03} is rejected. It means that there is statistical significance difference between $Diff_{toffa}$ and $Diff_{con}$ for the Smart Home DSPL. Figure 5.13 depicts the box plot concerning the **Diff** metric. For this metric, the

Table 5.7: Scenarios for Smart Home DAS by considering the valid $CCFs$ and prioritization of soft goals presented in the first and fourth columns of the table. After executing the simulations, we collected the metrics Pos, Neg, and Diff for each configuration suggested in second and third columns.

Results about configurations (Variable features)			**Results about metrics**						
CCFs	**ConG4DaS**	**ToffA-DAS**	**Soft goal Priority**	**ConG4DaS**			**ToffA-DAS**		
				Pos	**Neg**	**Diff**	**Pos**	**Neg**	**Diff**
$ccf_1 = \{none\}$	f_3,f_5,f_8	f_3,f_5,f_9	$sg_2 > sg_1 > sg_3$	2	0	2	1	1	0
$ccf_2 = \{c_{10}\}$	f_3,f_5,f_8	f_3,f_5,f_9	$sg_2 > sg_1 > sg_3$	2	0	2	1	1	1
$ccf_3 = \{c_8\}$	f_3,f_5,f_9	f_3,f_5,f_9	$sg_1 > sg_3 > sg_2$	2	1	1	2	1	1
$ccf_4 = \{c_8, c_{10}\}$	f_3,f_5,f_9	f_3,f_5,f_9	$sg_1 > sg_3 > sg_2$	2	1	1	2	1	1
$ccf_5 = \{c_6\}$	f_3,f_5,f_9	f_3,f_5,f_9	$sg_2 > sg_1 > sg_3$	1	1	0	1	1	0
$ccf_6 = \{c_6, c_{10}\}$	f_3,f_5,f_9	f_3,f_5,f_9	$sg_2 > sg_1 > sg_3$	1	1	0	1	1	0
$ccf_7 = \{c_9, c_8\}$	f_3,f_5,f_9	f_3,f_5,f_9	$sg_1 > sg_3 > sg_2$	2	1	1	2	1	1
$ccf_8 = \{c_6, c_8, c_{10}\}$	f_3,f_5,f_9	f_3,f_5,f_9	$sg_1 > sg_3 > sg_2$	2	1	1	2	1	1
$ccf_9 = \{c_4\}$	f_3,f_6,f_8,f_{12}	f_3,f_6,f_9,f_{12}	$sg_2 > sg_1 > sg_3$	2	0	2	1	1	0
$ccf_{10} = \{c_4, c_{10}\}$	f_3,f_6,f_8,f_{12}	f_3,f_6,f_9,f_{12}	$sg_2 > sg_1 > sg_3$	2	0	2	1	1	0
$ccf_{11} = \{c_4, c_8\}$	f_3,f_6,f_9,f_{12}	f_2,f_6,f_9,f_{12}	$sg_1 > sg_3 > sg_2$	1	2	-1	2	1	1
$ccf_{12} = \{c_4, c_8, c_{10}\}$	f_3,f_6,f_9,f_{12}	f_3,f_5,f_9,f_{12}	$sg_1 > sg_3 > sg_2$	1	2	-1	2	1	1
$ccf_{13} = \{c_4, c_6\}$	f_3,f_6,f_9,f_{12}	f_3,f_6,f_9,f_{12}	$sg_2 > sg_1 > sg_3$	1	1	0	1	1	0
$ccf_{14} = \{c_4, c_6, c_{10}\}$	f_3,f_6,f_9,f_{12}	f_3,f_6,f_9,f_{12}	$sg_2 > sg_1 > sg_3$	1	1	0	1	1	0
$ccf_{15} = \{c_4, c_6, c_8\}$	f_3,f_6,f_9,f_{12}	f_2,f_6,f_9	$sg_1 > sg_3 > sg_2$	1	2	-1	1	2	-1
$ccf_{16} = \{c_4, c_6, c_8, c_{10}\}$	f_3,f_6,f_9,f_{12}	f_3,f_5,f_9	$sg_1 > sg_3 > sg_2$	1	2	-1	2	1	1
$ccf_{17} = \{c_1\}$	f_2,f_5,f_8	f_2,f_5,f_9	$sg_2 > sg_1 > sg_3$	1	1	0	0	2	-2
$ccf_{18} = \{c_1, c_{10}\}$	f_3,f_5,f_8	f_3,f_5,f_9	$sg_2 > sg_1 > sg_3$	2	0	2	1	1	0
$ccf_{19} = \{c_1, c_8\}$	f_2,f_5,f_9	f_2,f_5,f_9	$sg_1 > sg_3 > sg_2$	3	0	3	3	0	3
$ccf_{20} = \{c_1, c_9, c_{10}\}$	f_2,f_5,f_9	f_2,f_5,f_9	$sg_1 > sg_3 > sg_2$	3	0	3	3	0	3
$ccf_{21} = \{c_1, c_6\}$	f_2,f_5,f_9	f_2,f_5,f_9	$sg_2 > sg_1 > sg_3$	0	2	-2	0	2	-2
$ccf_{22} = \{c_1, c_6, c_{10}\}$	f_3,f_5,f_9	f_3,f_5,f_9	$sg_2 > sg_1 > sg_3$	1	1	0	1	1	0
$ccf_{23} = \{c_1, c_6, c_8\}$	f_2,f_5,f_9	f_2,f_5,f_9	$sg_1 > sg_3 > sg_2$	3	0	3	3	0	3
$ccf_{24} = \{c_1, c_6, c_8, c_{10}\}$	f_2,f_5,f_9	f_2,f_5,f_9	$sg_1 > sg_3 > sg_2$	3	0	3	3	0	3
$ccf_{25} = \{c_1, c_4\}$	f_2,f_6,f_8,f_{12}	f_2,f_6,f_9,f_{12}	$sg_2 > sg_1 > sg_3$	1	0	1	0	2	-2
$ccf_{26} = \{c_1, c_4, c_{10}\}$	f_3,f_5,f_8,f_{12}	f_3,f_6,f_9,f_{12}	$sg_2 > sg_1 > sg_3$	2	0	2	1	1	0
$ccf_{27} = \{c_1, c_4, c_8\}$	f_2,f_6,f_9,f_{12}	f_2,f_6,f_9,f_{12}	$sg_1 > sg_3 > sg_2$	2	1	1	2	1	1
$ccf_{28} = \{c_1, c_4, c_8, c_{10}\}$	f_2,f_6,f_9,f_{12}	f_2,f_6,f_9,f_{12}	$sg_1 > sg_3 > sg_2$	2	1	1	2	1	1
$ccf_{29} = \{c_1, c_4, c_6\}$	f_2,f_6,f_9,f_{12}	f_2,f_6,f_9,f_{12}	$sg_2 > sg_1 > sg_3$	0	2	-2	0	2	-2
$ccf_{30} = \{c_1, c_4, c_6, c_{10}\}$	f_3,f_6,f_9,f_{12}	f_2,f_6,f_9,f_{12}	$sg_2 > sg_1 > sg_3$	1	1	0	1	1	0
$ccf_{31} = \{c_1, c_4, c_6, c_8\}$	f_2,f_6,f_9,f_{12}	f_2,f_6,f_9,f_{12}	$sg_1 > sg_3 > sg_2$	2	1	1	2	1	1
$ccf_{32} = \{c_1, c_4, c_6, c_8, c_{10}\}$	f_2,f_6,f_9,f_{12}	f_3,f_5,f_9	$sg_1 > sg_3 > sg_2$	2	1	1	2	1	1

median value was equal to 1 for ConG4DaS and equal to 0.5 for ToffA-DAS. It may also be observed two outliers in the ToffA-DAS approach. One outlier indicates a greater **Diff** value in the scenarios ccf_{19}, ccf_{20}, ccf_{23}, and ccf_{24}, whereas the second one indicates lower **Diff** value in the scenarios ccf_{17}, ccf_{21}, ccf_{25}, and ccf_{29}. In general, the variation of the data set to the first approach was higher than in the second one. In summary, the ToffA-DAS approach presented more positive contributions to the *soft goals* with the highest priority than ToffA-DAS+ release, considering the scenarios defined for Smart Home DAS.

Figure 5.11: Pos Metric - Smart Home

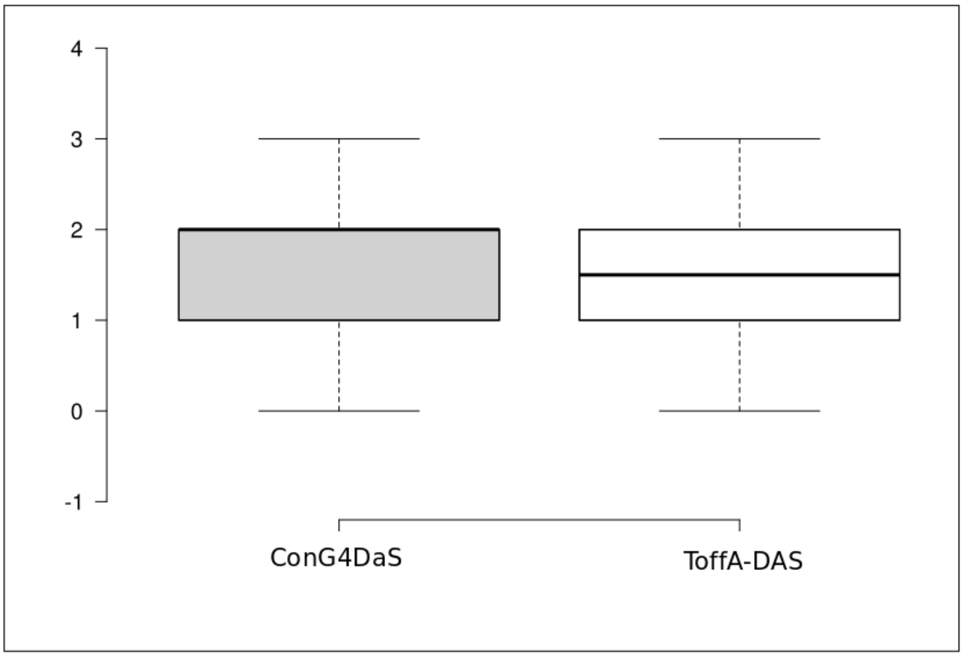
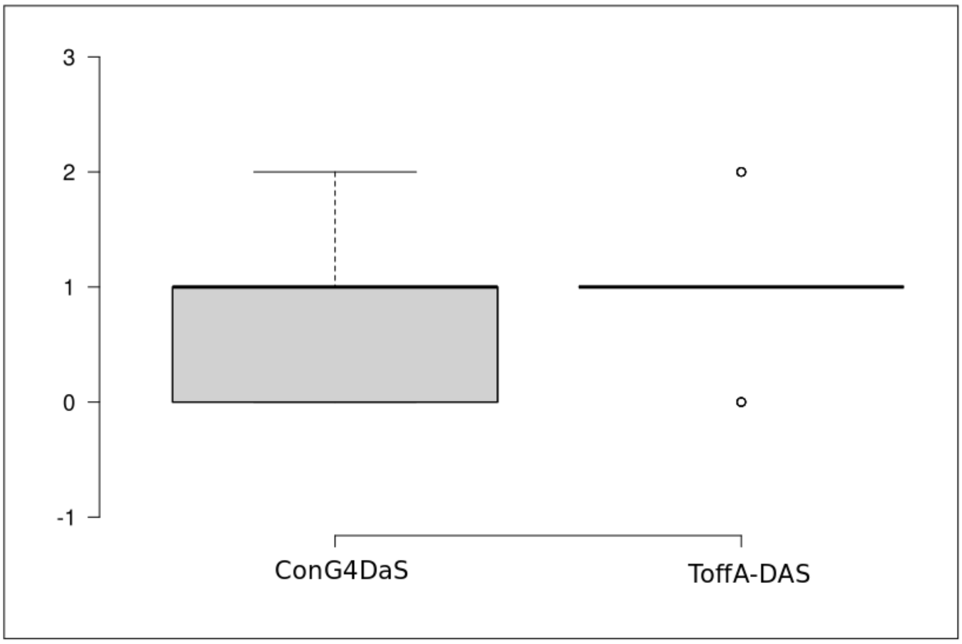


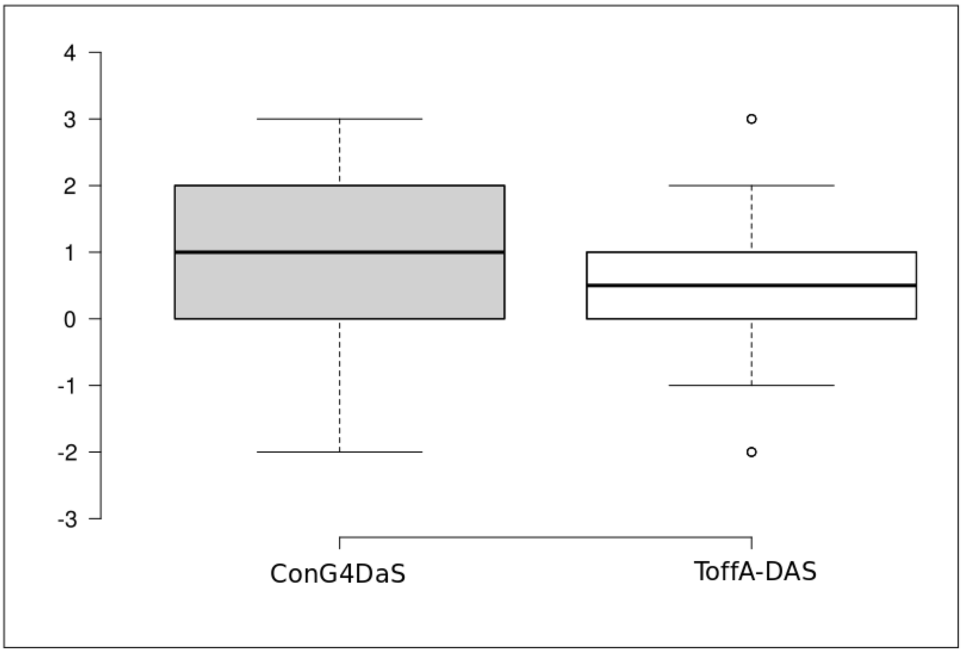
Figure 5.12: Neg Metric - Smart Home



5.4 DISCUSSION

In this section, we discuss the results found in this exploratory study and present threats to the validity of the results.

Figure 5.13: Diff Metric - Smart Home



5.4.1 Exploratory study

In the exploratory study, we assessed the ConG4DaS and ToffA-DAS approaches using both DAS, Mobile Game, and Smart Home. In Mobile Game, we observed that the configurations generated by ConG4DaS and ToffA-DAS were different for all CCFs. In Smart Home, the configurations resulting from simulations with both approaches were different for twelve from thirty-two CCFs. Furthermore, the set of configurations varied for each approach. When the ConG4DaS was applied, we obtained four different configurations in Mobile Game and six in Smart Home. By using ToffA-DAS, the set of configurations was also equal to four in Mobile Game and three in Smart Home.

Regarding the metrics, in all cases of the Mobile Game, the null hypothesis was not rejected due to the sample is not enough to return the critical value with the level of significance *p-value* $\leq$ 0.05. Conversely, in all cases of the Smart Home, the null hypothesis was rejected and, thus, the values of **Pos**, **Neg**, and **Diff** were different for the results generated by both approaches, ConG4DaS and ToffA-DAS.

We observed that the most frequent features also varied. In ToffA-DAS, the feature f_5 was presented in all configurations generated for Mobile Game, whereas in ConG4DaS, such a feature was presented only in three of the configurations generated. For Smart Home, f_9 was the feature appearing in all configurations generated by ToffA-DAS. By using ConG4DaS, in turn, feature f_8 was present in eight configurations. This resulted in a greater number of negative contributions to the *soft goals* with the highest priority in the ConG4DaS approach.

The evidence gathered in the performed evaluation shown that the set of configurations generated by the ToffA-DAS approach is different from those ones generated by the

ConG4DaS approach for most of the scenarios in both DAS, Mobile Game and Smart Home. Furthermore, the configurations generated by the ToffA-DAS execution provide higher satisfaction levels of *soft goals* than those generated by the ConD4DaS execution.

Such results were due to way to model contexts and assign the prioritization for modeling elements of both approaches. As opposed to ToffA-DAS, the ConG4Das approach does not deal with constraints among contexts and the contribution values for modeling elements. It considers quality contexts that are inserted in the goal model in order to affect the required satisfaction level of *soft goals*. It means that a given context is ranked according to the priority of NFRs. In our approach, however, contexts and NFRs are handled in an independent way by using both, goal model and eCFM. The ToffA-DAS approach measures the degree to which the variable features satisfy the *soft goals* and evaluates the impact of variable features over *contexts*, *goals*, and *soft goals*. In this way, it is possible to measure the contribution value for those modeling elements, which in turn can be modified according to stakeholder's preferences by resulting in different configurations.

In general, we consider ToffA-DAS as a comprehensive approach, since that it embraces *(i)* a variability modeling technique (eCFM); *(ii)* model verification by model checking, and *(iii)* prioritization of *soft goals*, *goals*, and *contexts*. Thus, our approach may support software engineering in the creation of correct models that can suit as a baseline to develop DAS. Indeed, such models will be composed of the best configurations to be adapted at runtime.

5.4.2 Threats to validity

In this study, we identified some threats to validity, which are described as follows:

Internal validity threats concern factors that can influence our observations. We have identified two internal validity threats. The first one is the mapping of all modeling elements. Unlike our approach that uses eCFM and goal model, the ConG4DaS approach is based on intentional *i**-orthogonal model and context model. For this reason, it was necessary to map all information among these models in both DAS, Mobile Game, and Smart Home. To mitigate this threat, we applied the pair review to assure that the models used in the experiment were correctly defined. The second validity threat is regarding the metrics calculation that was conducted manually. Even this step having been made cautiously, some mistakes could have happened during this process. To address this threat, we performed a pair review of the data set resulting from the metric measurements.

External validity threats concern the generalization of our findings and points required for experiment replications. Our study considers only eight scenarios for Mobile Game and thirty-two scenarios for Smart Home. This number of scenarios does not have statistical significance for most of the analysis of data set resulting from the metrics calculation and can be seen as a threat to external validity. However, we employed in this study the same raw data and metrics presented by Guedes *et. al.* [107] in order to compare both approaches, ConG4DaS and ToffA-DAS. Such scenarios were based on all possible valid CCFs, besides the negative and positive contributions that influence the satisfaction level of a *soft goal*. Thus, the findings of the analysis can be used as a baseline for other studies dealing DAS configuration selection process. All the data used to run this study are available[1] for replication and further details.

Construct validity threats concern the relationship between theory and observation. The ConG4DaS approach is based on seven levels (make, help, some+, unknown, break, hurt, and some-), whereas our approach uses five levels (satisfied, weakly satisfied, undecided, weakly denied and denied). However, the metrics must have a clear interpretation of the data set resulting from their measurements for both approaches. Thus, it was necessary to map all the negative and positive contributions that influence the satisfaction level of a *soft goal* by considering the different levels of ConG4DaS and ToffA-DAS. To mitigate this threat, we used the raw data of the ConG4Das and provide a clear interpretation under the data set. We also assessed the same metrics used by the authors of the ConG4DaS approach [107]. Moreover, the exploratory study protocol was developed in detail and reviewed by researchers in order to mitigate the threat to the construct validity of the exploratory study.

Conclusion validity threats concern the relationship between treatment and outcome. Thus, the exploratory study design must make sure that there was a statistical relationship between ConG4DaS and ToffA-DAS. For this reason, the results of the study were described using descriptive statistics, such as *median values* and *boxplot* to deal with numerical processing and presentation of the data set. It is an adequate method to describe the analysis and interpretation of the data type collected. Regarding the hypotheses defined for the exploratory study, we used the non-parametric *Wilcoxon Signed-rank Test* [124] because the study employs two related samples and it yields difference scores that may be ranked in order of absolute size. Such a statistical test is suitable not only for large samples but also with small samples.

5.5 CHAPTER SUMMARY

We developed the ToffA-DAS approach (Chapter 4) aiming to identify feasible configurations. With respect to identify feasible configurations, ToffA-DAS deals with the configuration selection process embracing the interactions between contexts and NFRs. Such an approach uses utility function as a strategy to express the priorities of users over services provided by DSPL applications. Those priorities are represented as weights aiming to direct the choice of an feasible solution.

We performed simulations with the *GridStix* DAS to gather initial evidence about the feasibility of using the ToffA-DAS approach from the point of view of *(i)* conduction of trade-off analysis by considering changes in the prioritization of elements that comprise eCFM and goal models such as *goals*, *soft goals*, and contexts, and *(ii)* definition of adaptation models, from feasible configurations found in the CCF change analysis.

We also conducted an exploratory study when ToffA-DAS and ConD4DaS approaches were compared with each other. All simulations presented consistent results and in accordance with the real-world scenarios and satisfied the estimated utility values and linear constraints. In addition, they meet the variability dimensions and different measurements of prioritization and satisfaction levels assigned to soft goals.

[1] https://sites.google.com/view/dspl-life-cycle/home

By observing the resulting data from evaluation, we concluded that there is no significant difference between ToffA-DAS and ConG4DaS approaches regarding the satisfaction of soft goals. However, ToffA-DAS embraces modeling, and prioritization of *goals*, *soft goals*, and contexts, thereby promoting a more comprehensive approach for the DAS configuration process.

The next chapter presents, an empirical study performed to improve the approach proposed in this thesis. We investigated the application of another optimization method for dealing with the configuration selection process such as multi-objective evolutionary algorithm.

Chapter 6

To lead the orchestra, you have to turn your back on the crowd. —Max Lucado

EVOLUTION OF THE TOFFA-DAS APPROACH

In Chapter 4, we proposed the ToffA-DAS approach to identify at design time, how the trade-off between contextual information and NFRs affect product configuration in DSPL engineering. ToffA-DAS embraces *domain analysis*, *modeling*, *prioritization*, *contribution*, and *optimization*. It also uses *Utility-based* planning as a strategy to assist software engineers in the trade-off analysis. Using this strategy, it is possible to apply an optimization method to simulate and evaluate a solution among possible configurations in accordance with model [3].

We defined a single-objective optimization problem and used the ILP technique to solve it. Then, we reported in Chapter 5 that using such a technique is quite efficient for identifying feasible configurations based on simulations. However, the multi-objective optimization provides a complete view of the trade-offs between their objectives functions that are attained by feasible solutions. Thus, we decided to formulate the objective functions for *contexts*, *goals*, and *soft goals* separately and exchanged the optimization method by applying a Genetic Algorithm (GA) [91]. GAs are a widely used optimization method to solve multi-objective problems. These algorithms manage a set of candidate solutions to an optimization problem that is combined and modified iteratively to obtain better solutions. Such a process stimulates the natural selection of the more adapted individuals that are selected and generate an improved offspring of solutions [125].

Additionally, we implemented a tool encompassing not only a GA but also the SAT solver technology to obtain information about feature model satisfiability. After evolving the ToffA-DAS approach, we conducted an exploratory study to compare both releases. The chapter consists of main sections as follows:

Section 6.1 discusses the related studies and compare them to our approach;

Section 6.2 describes how to combine ToffA-DAS with a GA and SAT solver technology to support the configuration selection process of DAS;

Section 6.3 presents the exploratory studys design and planning, the analysis, and the interpretation of the results;

Section 6.4 discusses the results of the exploratory study and threats to validity;

Section 6.5 reports the learned lessons identified after carrying out the empirical studies; and

Section 6.6 draws concluding remarks.

6.1 RELATED WORK

Variability management is an important activity that describes different configurations of the system [6]. This activity requires a consistent and scalable approach focused on supporting software engineering at design time in order to check the capacity of the system to meet self-adaptive operations. In this section, we discuss the existing approaches that are employed for supporting the selection of the most suitable system variants. When dealing with the configuration selection process to meet desired quality goals in DAS, most of the existing studies do not focus on the interactions between contextual information and NFRs and do not use a strategy to support the trade-off analysis [25]. Sousa *et al.* [15] present an overview of challenges with regard to quality evaluations of DAS. Such challenges were pointed out in the existing literature. The authors also report a set of research opportunities in this topic, such as the definition of thresholds for quality measures and the development of approaches for prioritization of NFRs according to DAS operations and domains.

In general, related studies presented by Hallsteinsen *et al.* [9], Ali *et al.* [110], Esfahani *et al.* [105], Greenwood *et al.* [106], Guedes *et al.* [107], Nascimento *et al.* [108], Sanchez *et al.* [109], Goldsby *et. al.* [74], Parra *et. al.* [80], Sawyer *et. al.* [81], Pascual *et. al.* [2], Ali *et. al.* [110], Gamez *et. al.* [111], and Welsh *et. al.* [112] propose approaches that model the interactions between contexts and NFRs. Additionally, the studies presented by Franco *et al.* [102], Edwards *et al.* [103], Hallsteinsen *et al.* [9], Esfahani *et al.* [105], Greenwood *et al.* [106], Guedes *et al.* [107], Nascimento *et al.* [108], and Sanchez *et al.* [109] use utility function to approximate the fulfillment of stakeholder's preferences in different situations. Among these studies, only Franco *et al.* [102], Edwards *et al.* [103], and Pacuar *et al.* [104] do not consider the trade-off between contextual information and NFRs in decision making. Conversely, only Pascual *et. al.* [2] use GA as an optimization method to support the configuration selection process. However, their approach is based on making decision at runtime.

Franco *et al.* [102] propose an approach to improve the planning phase of DAS by anticipating the reliability of each adaptation on NFRs. Such an approach uses quantitative prediction of NFRs to assure that a particular adaptation strategy will meet the required quality goals. Differently from our approach, they use an architectural description language to assess each strategy generated, *i.e.*, mathematical models that predict the failure behavior of DAS for each selected adaptation strategy.

Edwards *et al.* [103] propose an approach called DeSiRE to measure the extent to which NFRs are violated or satisfied by exploring the possible trade-off among them in the decision making. However, their work does not provide enough information to assist the interpretation of how managing the trade-off analysis. In contrast, Pacuar *et al.* [104] present initial results of how their approach can assist the system to enhance its behavior based on learning at runtime. Such an approach seeks the improvement of decision making by allowing the identification of utility preferences and their effects on the satisfaction of NFRs in each scenario.

Additionally, Pascual *et al.* [2] propose an approach to support the dynamic reconfiguration of mobile applications by considering the interactions between contexts and NFRs. However, it does not use *Utility-based* planning as a strategy to formalize the knowledge obtained. Their approach uses a multi-objective function that specifies objectives to be optimized, such as battery consumption, usability, and memory footprint. This is done by defining the contribution of these objectives for each feature in the model. Therefore, it explores the usage of GAs to generate feasible configurations that fit the current context by meeting the NFRs at runtime.

Our approach is focused on supporting the software engineers in the configuration selection process of DAS at design time. We designed ToffA-DAS to identify the feasible and valid configurations. Based on the artifacts generated from its execution, software engineers can define dynamic adaptation models that meet specific scenarios. These models represent the transitions among contextual changes that will fulfill the satisfaction of relevant NFRs and also suit as a baseline to implement the system to be adapted at runtime. ToffA-DAS uses the ILP to solve the optimization problem, which was defined based on a single-objective. However, the adoption of this technique may reduce the solution space treated. Therefore, in this paper, we decided to evolve the ToffA-DAS approach by applying a GA. Such an optimization method has a random search capability that promotes simulations closer to reality and deals with multi-objectives optimization problems [91]. We also improved the feature model analysis through the use of the SAT solver technology to obtain information about satisfiability.

6.2 THE TOFFA-DAS+ APPROACH

The ToffA-DAS approach aims to support trade-off analysis and can be reduced to an optimization model composed by a *utility function*. It uses the *Utility-based* planning [31] as a strategy to assist software engineers with trade-off analysis. Additionally, it uses a solver based on the ILP technique [89] to run the configuration process. As a result, it is possible to identify feasible configurations that meet all constraints. We argue that the specification of DAS can be done by using ToffA-DAS since it facilitates software engineers achieving consensus with stakeholders and understanding their preferences and needs. However, we realized that improvements could be achieved in the modeling and optimization steps. The green boxes in Figure 6.1 highlight the changes to be detailed in this study. We explain the differences between the previous release of ToffA-DAS and its current release through a running example *GridStix* [81], which was presented in Chapter 4.

Figure 6.1: ToffA-DAS Approach with SAT solver and GA

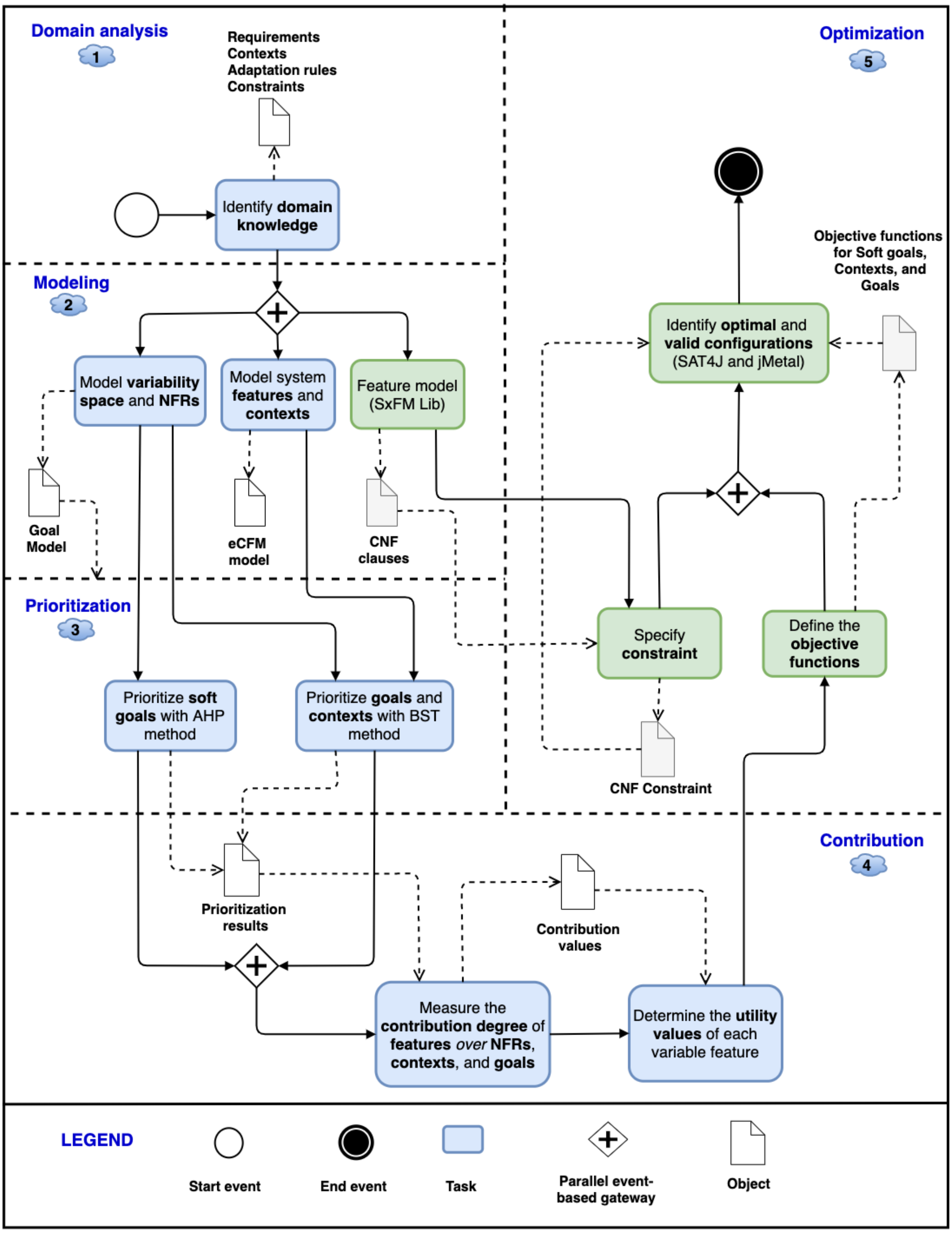


Figure 6.2: Tools and solutions that compose our approach

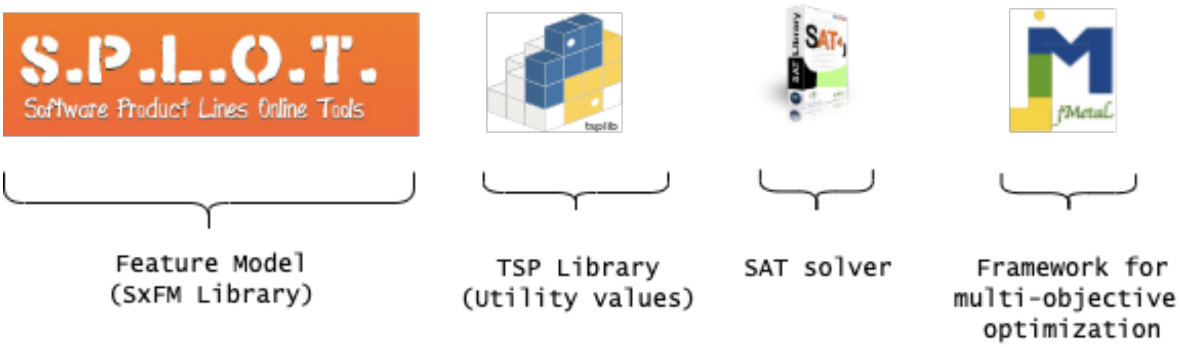
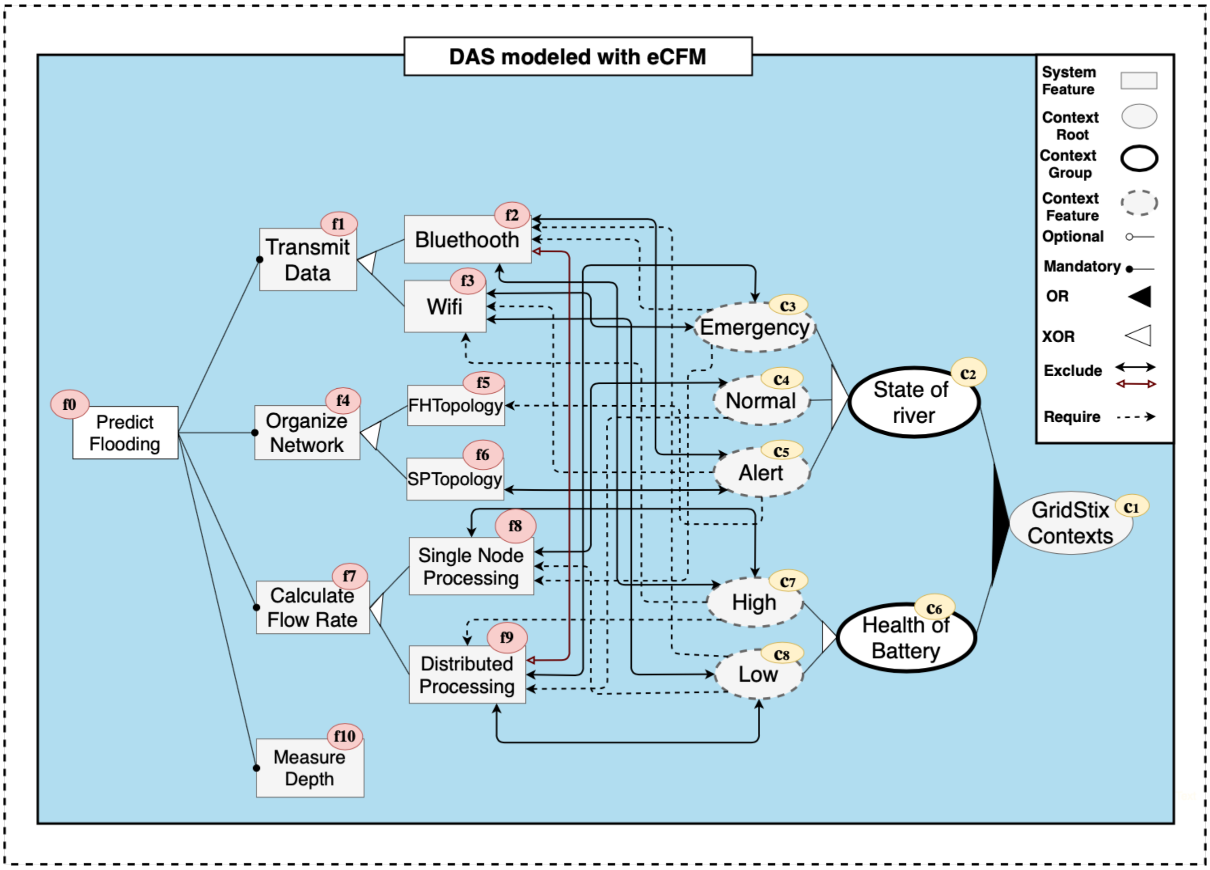


Figure 6.3: *GridStix* DAS modeled with eCFM. It represents all features, contexts features, context groups, context root, and their respective relationships.



In the SPL and DSPL fields, software engineers use feature models as design artifacts aiming to capture the similarity and variability between the possible configurations of products in a particular domain [77]. Usually, software engineers are only interested in a set of products composed of a valid combination of features. A valid product refers to the configuration that satisfies all constraints in the feature model. These constraints illustrate the relationships among features, which should be combined in a way to compose

Figure 6.4: Grid Stix DAS represented in SXFM format.

```
<feature_model name="GridStix DSPL">
<meta>
<data name="description">It establishes an automated mechanism
                         to warn about flood on rivers.</data>
<data name="creator">Michelle Carvalho</data>
<data name="organization">Federal University of Bahia</data>
</meta>
<feature_tree>
:r PredictFlooding(predict_flooding)
    :m TransmitData(transmit_data)
        :g (g_transmit_data) [1,1]
            : Bluethooth(bluethooth)
            : Wifi(wifi)
    :m OrganizeNetwork(organize_network)
        :g (g_organize_network) [1,1]
            : FHTopology(fh_topology)
            : SOPTopology(sop_topology)
    :m CalculateFlowRate(calculate_flow_rate)
        :g (g_calculate_flow_rate) [1,1]
            : SingleNodeProcessing(single_node_processing)
            : DistributingProcessing(distributing_processing)
    :m MeasureDepth(measure_depth)
</feature_tree>
<constraints> </constraints>
</feature_model>
```

the valid configurations [127]. By the way, the feature model analysis is an important task that includes checking whether the model has at last one valid configuration, detecting dead features, counting valid configurations, and so on [128]. Using SAT solver technology enables software engineers to obtain information for the feature model analysis based on its properties. However, such a model needs to be transformed into propositional logic. Accordingly, we evolved our approach by changing the way to represent and analyze feature models (**STEP 2-Modeling**).

In the previous release of the ToffA-DAS approach, we used mono-objective optimization to find a feasible solution that represents the maximum of the objective function. However, we understood that is possible to obtain the best solution in an aspect of the system by meeting multi-objectives. Thus, we formulated the objective functions for *goals*, *contexts*, and *soft goals* separately. In this case, it is common that such objectives are conflicting resulting in not only one feasible solution, but a set of solutions considering all objectives simultaneously. GAs are a widely used strategy to solve multiobjective optimization problems [126]. Thus, we changed the optimization method (**STEP 5-Optimization**) to be applied in the feature selection problem in order to obtain feasible solutions based on a GA, which is an iterative learning process. This optimization method uses the adaptive ability to accumulate information for approaching itself of the feasible solutions gradually and in a multi-directional way [129]. Figure 6.2 illustrates the tools used to support the improvements of our approach that are presented as follows.

6.2.1 Modeling

In **STEP 2-Modeling**, we used the *Java parser library* to represent the features and constraints that were defined in the feature model. Such library is for Simple *XML Feature Model format* (*SXFM*), which was defined by *SPLOT* website [130]. *SPLOT* is a host to a feature model repository that adheres to the *SXFM* format. Figure 6.4 depicts the *GridStix* DAS represented in *SXFM* format. It shows the root feature (defined as a **:r**), the mandatory features (defined as **:m**), the optional features (defined as **:o**), and the group features (defined as **:g**). Once the feature model was transformed on a *SXFM* format, we implemented an XML parser to validate it, by using an SAT solver. But for this purpose, the feature model is transformed into a propositional formula and then converted into an equivalent formula that is in Conjunctive Normal Form (CNF) [131, 132].

In the CNF form, features and constraints are expressed as a conjunction of n clauses, $C_1, C_2, ..., C_n$, where a clause is a disjunction of several literals, each of which is a feature that is selected (f_i) or not ($\neg f_i$). For example, the feature model in Figure 6.3 is expressed in the following propositional formula:

$$
\begin{aligned}
&PredictFlooding(f_0) \wedge \\
&PredictFlooding(f_0) \leftrightarrow TransmitData(f_1) \wedge \\
&PredictFlooding(f_0) \leftrightarrow OrganizeNetwork(f_4) \wedge \\
&PredictFlooding(f_0) \leftrightarrow CalculateFlowRate(f_7) \wedge \\
&PredictFlooding(f_0) \leftrightarrow MeasureDepth(f_{10}) \wedge \\
&Bluetooth(f_2) \leftrightarrow \{Wifi(\neg f_3) \wedge TransmitData(f_1)\} \wedge \\
&Wifi(f_3) \leftrightarrow \{Bluetooth(\neg f_2) \wedge TransmitData(f_1)\} \wedge \\
&FHTopology(f_5) \leftrightarrow \{SPTopology(\neg f_6) \\
&\wedge OrganizeNetwork(f_4)\} \wedge \\
&SPTopology(f_6) \leftrightarrow \{FHTopology(\neg f_5) \\
&\wedge OrganizeNetwork(f_4)\} \wedge \\
&SingleNodeProcessing(f_8)\{DistributingProcessing(\neg f_9) \\
&\wedge CalculateFlowRate(f_7)\} \wedge \\
&DistributingProcessing(f_9)\{SingleNodeProcessing(\neg f_8) \\
&\wedge CalculateFlowRate(f_7)\}
\end{aligned}
\tag{6.1}
$$

This propositional logic can be written in the following CNF: $f_0 \wedge (\neg f_0 \vee f_1 \vee f_4 \vee f_7 \vee f_{10}) \wedge (f_0 \vee \neg f_1) \wedge (f_0 \vee \neg f_4) \wedge (f_0 \vee \neg f_7) \wedge (f_0 \vee \neg f_{10}) \wedge (\neg f_2 \vee \neg f_3) \wedge (\neg f_2 \vee f_1) \wedge (f_2 \vee f_3 \vee \neg f_1) \wedge (\neg f_3 \vee \neg f_2) \wedge (\neg f_3 \vee f_1) \wedge (f_3 \vee f_2 \vee \neg f_1) \wedge (\neg f_5$

$\lor \neg f_6) \land (\neg f_5 \lor f_4) \land (f_5 \lor f_6 \lor \neg f_4) \land (\neg f_6 \lor \neg f_5) \land (\neg f_6 \lor f_4) \land (f_6 \lor f_5 \lor \neg f_4) \land (\neg f_8 \lor \neg f_9) \land (\neg f_8 \lor f_7) \land (f_8 \lor f_9 \lor \neg f_7) \land (\neg f_9 \lor \neg f_8) \land (\neg f_9 \lor f_7) \land (f_9 \lor f_8 \lor \neg f_7)$, which in turn is loaded on a standard format named DIMACS [133] to be parsed by a SAT solver.

Figure 6.5: *GridStix* DAS represented with CNF clauses.

```
----------HIERARCHY AND CNF CLAUSES----------
---------- Root Feature: PredictFlooding
# Root CNF Clause: 1
---------- Parent feature: PredictFlooding
Child feature: TransmitData
# Group CNF Clause: -1,2
---------- Parent feature: PredictFlooding
Child feature: OrganizeNetwork
# Group CNF Clause: -1,3
---------- Parent feature: PredictFlooding
Child feature: CalculateFlowRate
# Group CNF Clause: -1,4
---------- Parent feature: PredictFlooding
Child feature: MeasureDepth
# Group CNF Clause: -1,5
---------- Parent feature: TransmitData
Child feature: Bluethooth
# OR CNF Clause: 2,-6
Child feature: Wifi
# OR CNF Clause: 2,-7
#-# Group CNF Clause: -2,6,7
#-# XOR CNF Clause: -6,-7
---------- Parent feature: OrganizeNetwork
Child feature: FHTopology
# OR CNF Clause: 3,-8
Child feature: SOPTopology
# OR CNF Clause: 3,-9
#-# Group CNF Clause: -3,8,9
#-# XOR CNF Clause: -8,-9
---------- Parent feature: CalculateFlowRate
Child feature: SingleNodeProcessing
# OR CNF Clause: 4,-10
Child feature: DistributingProcessing
# OR CNF Clause: 4,-11
#-# Group CNF Clause: -4,10,11
#-# XOR CNF Clause: -10,-11
```

In this study, we use Sat4j [134], which is an open-source library of SAT solvers. It enables Java developers to access cross-platform SAT-based solvers for solving optimization problems. Figure 6.6 shows an example of analysis made in the *GridStix* feature model. The solver suggested that this model is consistent and provides eight valid con-

Figure 6.6: Consistency analysis of the *GridStix* feature model using Sat4j lib.

```
------------------------------------------ CONSISTENCY ------------------------------------------
Feature model is consistent!
It should have at least one valid configuration to be consistent.
------------------------------ FEATURE INDEX | TYPE OF FEATURES ---------------------------------
VALUES (TRUE X FALSE)
# transmit_data : [true ] (Mandatory)
# wifi : [false true ] (Variable)
# calculate_flow_rate : [true ] (Mandatory)
# measure_depth : [true ] (Mandatory)
# predict_flooding : [true ] (Mandatory)
# bluethooth : [false true ] (Variable)
# fh_topology : [false true ] (Variable)
# distributing_processing : [false true ] (Variable)
# organize_network : [true ] (Mandatory)
# sop_topology : [false true ] (Variable)
# single_node_processing : [false true ] (Variable)
SOLVER STATISTICS
Total Mandatory Features.....: 5
Total Variable Features......: 6
Total Dead Features..........: 0
Running Time.................: 5
Number of SAT solver Checks..: 7
-------------------------------- NUMBER OF VALID CONFIGURATIONS ---------------------------------
Feature model has 8 possible valid configurations
Number of variables: 11
------------------------------------ ALL VALID CONFIGURATIONS -----------------------------------
1=[1=true, 2=true, 3=true, 4=true, 5=true, 6=true, -7=false, 8=true, -9=false, 10=true, -11=false]
2=[1=true, 2=true, 3=true, 4=true, 5=true, 6=true, -7=false, 8=true, -9=false, -10=false, 11=true]
3=[1=true, 2=true, 3=true, 4=true, 5=true, 6=true, -7=false, -8=false, 9=true, -10=false, 11=true]
4=[1=true, 2=true, 3=true, 4=true, 5=true, 6=true, -7=false, -8=false, 9=true, 10=true, -11=false]
5=[1=true, 2=true, 3=true, 4=true, 5=true, -6=false, 7=true, -8=false, 9=true, -10=false, 11=true]
6=[1=true, 2=true, 3=true, 4=true, 5=true, -6=false, 7=true, 8=true, -9=false, 10=true, -11=false]
7=[1=true, 2=true, 3=true, 4=true, 5=true, -6=false, 7=true, 8=true, -9=false, -10=false, 11=true]
8=[1=true, 2=true, 3=true, 4=true, 5=true, -6=false, 7=true, -8=false, 9=true, 10=true, -11=false]
```

figurations by satisfying all constraints. In addition, it is possible to identify the type of features, count them, and attribute Boolean values in the following way: the `Mandatory` feature receives Boolean value *True*, since it is always selected, whereas the variable feature (`Optional`, `Or`, and `Xor`) receives values *True* or *False*. It means that a variable feature can or not be selected in a specific configuration. Configuration 1, for instance, shows that features f_1, f_2, f_3, f_4, f_5, f_6, f_8, and f_{10} are selected (Boolean value *True*), whereas features f_7, f_9, and f_{11} are not selected (Boolean value *False*). Such information is used as an input for the algorithm that represents the problem to be addressed by GA. Based on the indexes of the variable features that the GA is informed which genes can have their values flipped during the execution of the *mutation* operation (see Listings 6.6 and 6.7).

After executing the **STEPS 1-4**, the software engineers are able to select a feasible set of features that can maximize stakeholder's preferences. Therefore, it is possible to run the configuration process and use a SAT solver to check the feature model satisfiability and a GA to find feasible configurations that meet the trade-off between contexts and NFRs.

6.2.2 Optimization

For **STEP 5-Optimization**, we defined an optimization model that suggests feasible and valid configurations by considering the integrity constraints and variability of the feature model. In addition, it ensures that solutions satisfy *contexts*, *goals*, *soft goals*. It is characterized, as follows:

- Let n be the number of features fi extracted from feature model;
- Let F be a set of features fi, when $|F| = n$;
- A set of **decision variables** x_{fi} whose value is equal to 1 if the feature fi is selected, 0 otherwise;
- Let $\mathbb{C}_{fi}$ be the contribution value of feature over *contexts*. The **first objective function** measures the decision variables summation by satisfying the *contexts* represented in eCFM (equation 6.2);

$$\max \sum_{fi}^{n} \mathbb{C}_{fi} \cdot X_{fi}, \forall f_i \in F \tag{6.2}$$

- Let $\mathbb{G}_{fi}$ be the contribution value of feature over *goals*. The **second objective function** measures the decision variables summation by satisfying the *goals* represented in goal model (equation 6.3);

$$\max \sum_{fi}^{n} \mathbb{G}_{fi} \cdot X_{fi}, \forall f_i \in F \tag{6.3}$$

- Let $\mathbb{S}_{fi}$ be the contribution value of feature over *soft goals*. The **third objective function** measures the decision variables summation by satisfying the *soft goals* represented in goal model (equation 6.4);

$$\max \sum_{fi}^{n} \mathbb{S}_{fi} \cdot X_{fi}, \forall f_i \in F \tag{6.4}$$

- Let C_{cnf} be a set of CNF clauses (translation 6.1) that is subject to the integrity constraints and variability of the feature model. The **constraint** consists of not violating any CNF clause $cnf_i \subseteq C_{cnf}$.

Additionally, we developed such an optimization model in the Java language that receives as input the expressions created referring to the algebraic form of *utility functions* and the CNF expression referring to the DIMACS form of features and constraints. We also use the TSP library [115] to numerically represent the *utility values* of each objective function. Such a library of sample instances is used by the optimization model, which in turn is executed by non-dominated sorting GA [135]. Such an algorithm is hosted at the jMetal [136] that consists of a Java-based framework for solving multi-objective optimization problems with meta-heuristics.

Figure 6.7 shows an example using the TSP library defined for *GridStix* DAS. The dimension consists of the total of the system's features (in the running example this number is equal to 11). Each section represents the system's features and *utility values* (weighs). In section *CONTEXT-SECTION*, we represented 11 relationships between feature and its respective *utility value*. Each *utility value* represents the contribution degree of feature *over* contexts (calculated in **STEP 4**). The feature `Bluetooth` (f_6 with index 5), for instance, has a *utility value* equal to 0.5, whereas the feature `SOPTopology` (f_9 with index 8) has a *utility value* equal to 0.25. This reasoning is applied for all features in sections *CONTEXT-SECTION*, *GOAL-SECTION* and *SOFTGOAL-SECTION*. Thus, a method called *readTSPFile()* was implemented to read the TSP file and *utility values* for *goals*, *contexts*, and *soft goals*.

The Algorithm on Listing 6.1 shows how to read and store all contribution values (*utility values*) of features *over* contexts, goals, and soft goals as ordered collections, which were named as *contextCoefficientsList*, *goalCoefficientsList*, and *softgoalCoefficientsList* (Lines 39, 64, and 89), respectively. Thus, the method *readTSPFile()* was implemented to read the TSP file and *utility values* for *goals*, *contexts*, and *soft goals*.

```
1  public void readTSPFile() throws IOException{
2      //TSP FILE
3    InputStream file = new FileInputStream("FeatureModels/
        GridStix/GridStix.objectiveFunctions");
4    InputStreamReader fileReader=new InputStreamReader(file);
5    BufferedReader fileReaderBuffered = new BufferedReader(
        fileReader);
6    StreamTokenizer token = new StreamTokenizer(
        fileReaderBuffered);
7
8    boolean found;
9    found = false;
10
11   // Find the string DIMENSION
12   token.nextToken();
13   while(!found) {
14     if((token.sval !=null) && ((token.sval.compareTo(
          Dimension) == 0)))
15       found = true;
16     else
17       token.nextToken();
18   }
19   token.nextToken();
20   token.nextToken();
21   numberOfVariablesTSP = (int)token.nval;
```

```java

    // Find the string CONTEXT_SECTION
    token.nextToken();
    while(!found) {
      if((token.sval !=null) && ((token.sval.compareTo(
         ContextContribution) == 0)))
        found = true;
      else
        token.nextToken();
    }
    token.nextToken();
    token.nextToken();
    numberOfContexts = (int)token.nval;
    token.nextToken();
    token.nextToken();

    //Array with an ordered collection (value = utility value
       for feature)
    //New object
    contextCoefficientsList = new ArrayList<Double>();
    for(int i = 0; i < numberOfContexts; i++) {
      token.nextToken();
      int keyContext = (int)token.nval ;
      token.nextToken();
      double valueContext = (double)token.nval;
      contextCoefficientsList.add(valueContext);
    }
    token.nextToken();
    token.nextToken();

    //Find the string GOAL_SECTION
    while(!found) {
      if((token.sval !=null) && ((token.sval.compareTo(
         GoalContribution) == 0)))
        found = true;
      else
        token.nextToken();
    }
    token.nextToken();
    token.nextToken();
    token.nextToken();
    numberOfGoals = (int)token.nval;

    //Array with an ordered collection (value = utility value
       for feature)
```

```
    //New object
    goalCoefficientsList = new ArrayList<Double>();
    for(int i = 0; i < numberOfGoals; i++) {
      token.nextToken();
      int keyGoal = (int)token.nval ;
      token.nextToken();
      double valueGoal = (double)token.nval;
      goalCoefficientsList.add(valueGoal);
    }
    token.nextToken();
    token.nextToken();

    // Find the string SOFTGOAL_SECTION
    while(!found) {
      if((token.sval !=null) && ((token.sval.compareTo(
         SoftgoalContribution) == 0)))
        found = true;
      else
        token.nextToken();
    }
    token.nextToken();
    token.nextToken();
    token.nextToken();
    numberOfSoftgoals = (int)token.nval;

    //Array with an ordered collection (value = utility value
       for feature)
    //New object
    softgoalCoefficientsList = new ArrayList<Double>();
    for(int i = 0; i < numberOfSoftgoals; i++) {
      token.nextToken();
      int keySoftgoal = (int)token.nval;
      token.nextToken();
      double valueSoftgoal = (double)token.nval;
      softgoalCoefficientsList.add(valueSoftgoal);
    }
  }
```

Listing 6.1: readTSPFile Method

Listing 6.2 (Lines 11-13) shows the lists `valueContext`, `valueGoal`, and `valueSoftGoal` used to manipulate within the method *evaluate(Solution solution)*, all *utility values* for *contexts*, *goals*, and *soft goals*. These *utility values* can be accessed through of the methods *getcontextCoefficientsList()*, *getgoalCoefficientsList()*, and *getsoftgoalCoefficientsList()* in order to formulate the objective functions as follows:

Figure 6.7: Utility values represented using TSP library.

```
NAME : Objective Functions - GridStix
COMMENT : MICHELLE CARVALHO
TYPE : DSPL
DIMENSION : 11
CONTEXT_SECTION : 11
0 0
1 0
2 0.3333333333
3 0.3333333333
4 0
5 0.5
6 0.5
7 0
8 0.25
9 0.25
10 0.25
GOAL_SECTION : 11
0 0
1 0
2 0.1666666667
3 -0.1666666667
4 0
5 0
6 0
7 0
8 0.1666666667
9 -0.1666666667
10 0
SOFTGOAL_SECTION : 11
0 0
1 0
2 0.50309674
3 -0.056174904
4 0
5 0
6 0
7 0
8 0.5802675585
9 -0.2105165366
10 0
EO
```

- We expect `Variable[]` (chromosome) to be an array of one-size binary variables. The chromosome to be defined and evaluated should have its size equal to the number of features (Line 15). It consists of a set of decision variables (genes) whose value is *true* if the feature is inserted, 0 otherwise;

- The source code in Lines 18-30 shows how to manipulate and store the decision variable values as binary numbers in a list named *solutionInteger* that will be used later when measuring the objective functions. The decision variable values is manipulated and stored as binary numbers that will be used later when measuring the objective functions. First, it is necessary to identify through the indexes, which genes will receive value 0 or 1. If the gene is inserted (*true*), then receives value equal to 1, whereas if the gene is not inserted, then receives value equal to 0 (*false*). This evaluation is based on the propositional logic that represents features and constraints (Section 6.2.1).

- The first objective function measures the decision variables summation by satisfying the contexts (Lines 36-47). Each double value (stored in list `valueContext`), is multiplied by integer value 0 or 1 (stored in the list `solutionInteger`). Such an operation is performed according to the corresponding index (Line 37) and the resulting value is stored in the list `sumObjectiveContext` (Line 43). The method called *solution.setObjective()* (Line 74) sets this value to be accessed during the execution of GA.

- The second objective function measures the decision variables summation by satisfying the soft goals (Lines 50-61). Each double *utility value* (stored in list `valueSoftGoal`), is multiplied by integer value 0 or 1 (stored in the list `solutionInteger`). Such an operation is performed according to the corresponding index (Line 51) and the resulting value is stored in the list `sumObjectiveSoftGoal` (Line 47). The method *solution.setObjective()* (Line 75) sets this value to be accessed during the execution of GA.

- The third objective function measures the decision variables summation by satisfying the goals (Lines 64-72). Each double *utility value* (stored in list *valueGoal*), is multiplied by integer value 0 or 1 (stored in the list `solutionInteger`). Such an operation is performed according to the corresponding index (Line 65) and the resulting value is stored in the list `sumObjectiveGoal` (Line 61). The method *solution.setObjective()* (Line 76) sets this value to be accessed during the execution of GA.

```
1   /**
2    * Evaluates a solution
3    * @param solution The solution to evaluate
4    * @throws JMException
5    */
6   public void evaluate(Solution solution) throws JMException {
7     if (!(solution.getType() instanceof BinarySolutionTypeNew)
          ) {
8       throw new JMException("Unexpected solution type");
9     }
10
11    List<Double>valueContext=TSP_ContributionValues.
          getcontextCoefficientsList();
12    List<Double>valueGoal=TSP_ContributionValues.
          getgoalCoefficientsList();
13    List<Double>valueSoftGoal=TSP_ContributionValues.
          getsoftgoalCoefficientsList();
14
15    Variable[] variable=solution.getDecisionVariables();
16    Binary bin=(Binary) variable[0];
17
18    solutionInteger = new ArrayList<Integer>();
```

```

    for (int i = 0; i < bin.getNumberOfBits(); i++) {
      boolean binaryValue=bin.getIth(i);
      if (binaryValue == true) {
        int geneValue = 1;
        solutionInteger.add(geneValue);
      }
      else if (binaryValue == false) {
        int geneValue = 0;
        solutionInteger.add(geneValue);
      }
    }

    ContributionContext = new ArrayList<Double>();
    sumObjectiveContext = 0;

        //Objective function (1) => Context
    for(int y = 0; y < valueContext.size(); y++) {
      contributionValue = solutionInteger.get(y)*valueContext.
         get(y);
      ContributionContext.add(contributionValue);
    }

        //Sum total
    for (int k = 0; k < ContributionContext.size(); k++) {
      sumObjectiveContext += ContributionContext.get(k);
    }

    ContributionSoftGoal = new ArrayList<Double>();
    sumObjectiveSoftGoal=0;

    // Objective function (2) => SoftGoals
    for(int y = 0; y < valueSoftGoal.size(); y++) {
      contributionValue = solutionInteger.get(y)*valueSoftGoal
         .get(y);
      ContributionSoftGoal.add(contributionValue);
    }

    //Sum total
    for (int k = 0; k < ContributionSoftGoal.size(); k++) {
      sumObjectiveSoftGoal += ContributionSoftGoal.get(k);
    }

    ContributionGoal = new ArrayList<Double>();
    sumObjectiveGoal = 0;
```

```

    // Objective function (3) => Goals
    for(int y = 0; y < valueGoal.size(); y++) {
      contributionValue = solutionInteger.get(y)*valueGoal.get
         (y);
      ContributionGoal.add(contributionValue);
    }

    //Sum total
    for (int k = 0; k < ContributionGoal.size(); k++) {
      sumObjectiveGoal += ContributionGoal.get(k);
    }

    solution.setObjective(0, sumObjectiveContext);
    solution.setObjective(1, sumObjectiveSoftGoal);
    solution.setObjective(2, sumObjectiveGoal);
  }// evaluate
```

Listing 6.2: Evaluation method (GridStixProblem class

The number of objective functions, constraints, and chromosomes to be evaluated at a time are defined within the method *GridStixProblem()* in Listing 6.3. This method creates a new instance of the problem by defining which type of solution to be manipulated through GA.

```
  /**
   * Constructor
   * Creates a new instance of the GridStix problem.
   * @param solutionType The solution type must "Binary"
   */
  public GridStixProblem(String solutionType, int nFeatures)
     throws ClassNotFoundException {
    numberOfVariables_ = 1;
    numberOfObjectives_ = 3;
    numberOfConstraints_ = 1;
    problemName_ = "GridStixProblem";
    solutionType_ = new BinarySolutionTypeNew(this);

    //List of feature indexes
    List<Integer> featuresIndex = ManageFeatures.
       getAllFeaturesIndex();

        //Reads the number of features (decision variables) to
           define the chromosome length
```

```
    length_ = new int[numberOfVariables_];
    length_[0] = featuresIndex.size();

    if (solutionType.compareTo("Binary") == 0)
      solutionType_ = new BinarySolutionTypeNew(this);
    else {
      System.out.println("GridStixProblem: solution type " + solutionType + " invalid");
      System.exit(-1);
    }
  } // GridStix
```

Listing 6.3: Method GridStixProblem()

The method *evaluateConstraints()* in Listing 6.4 aims to evaluate the constraint defined to the problem concerned. It consists of not violating any CNF clause that is subject to the variability and integrity of the model. Whether the chromosome under evaluation violates any CNF clause, the method *solution.setNumberofViolatedConstraint()* sets a value equal to one in order to eliminate it (Line 20). On the other hand, if it does not violate any CNF clause, the method *solution.setNumberofViolatedConstraint()* sets a value equal to zero for ensuring that this chromosome to be manipulated by GA (Line 23). The number of violated constraints is calculated within the method *NumberOfViolatedConstraints()* in Listing 6.5 and stored in the variable `NumberOfViolatedFeatures` (Line 16 on Listing 6.4).

In addition, we implemented a method to evaluate the constraint defined to the problem concerned. It consists of not violating any CNF clause that is subject to the variability and integrity of the model. Whether the chromosome under evaluation violates any CNF clause, the method *solution.setNumberofViolatedConstraint()* sets a value equal to one in order to eliminate it (Line 20). On the other hand, if it does not violate any CNF clause, such a method sets a value equal to zero for ensuring that this chromosome to be manipulated by GA.

```
1   /**
2    * Evaluates the constraint overhead of a solution
3    * @param solution
4    * @param solution The solution
5    * @throws JMException
6    */
7   public void evaluateConstraints(Solution solution) throws JMException {
8     if (!(solution.getType() instanceof BinarySolutionTypeNew)) {
```

```
      throw new JMException("Unexpected solution type");
    }

    Variable[] variable = solution.getDecisionVariables();
    Binary bin = (Binary) variable[0];

    int NumberOfViolatedFeatures = 0;
    NumberOfViolatedFeatures = NumberOfViolatedConstraints(bin
       );

    if (NumberOfViolatedFeatures > 0) {
      solution.setOverallConstraintViolation(
         NumberOfViolatedFeatures);
      solution.setNumberOfViolatedConstraint(1);
    } else {
      solution.setOverallConstraintViolation(0);
      solution.setNumberOfViolatedConstraint(0);
    }
  }
```

Listing 6.4: Method to evaluate constraints

In order to calculate the number of violated constraints, all CNF clauses are stored in a list named `clauseConstraintsComplete` (Lines 4-40 on Listing 6.5). Then the chromosome is evaluated by checking if its binary values that correspond to the decision variables, satisfy or not all clauses (as presented in Lines 42-62).

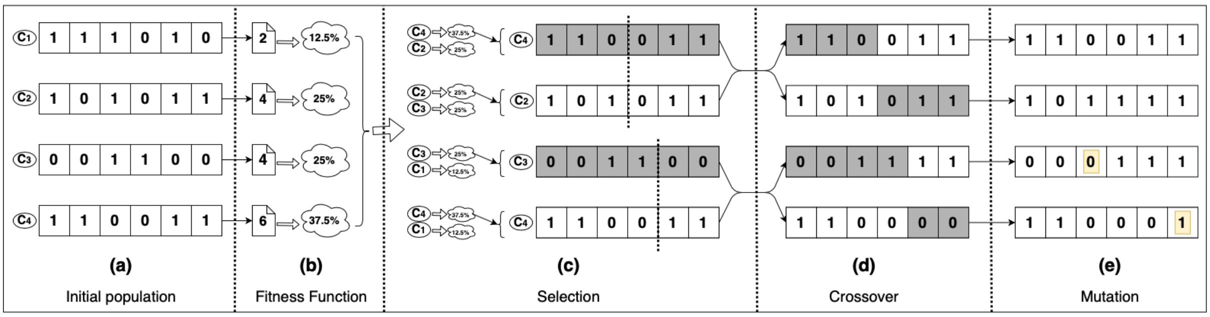
```
1    public int NumberOfViolatedConstraints(Binary bin) {
2
3      //Features and relationships
4      List<IVecInt> CNFclauses = SxFmFTReasoningWithSAT.
          getCNFList();
5      //Constraints
6      List<IVecInt> extra = SxFmReasoningWithSAT.
          getExtraConstraintsList();
7      //New object
8      clauseConstraintsComplete = new ArrayList<IVecInt>();
9
10     for(IVecInt extraConstraints : extra) {
11       if(clauseConstraintsComplete.isEmpty()) {
12       clauseConstraintsComplete.add(extraConstraints);
13       }else {
14         int count = 0;
```

```
          for(IVecInt clauseNew : clauseConstraintsComplete) {
            if(extraConstraints.equals(clauseNew)) {
              count++;
            }
          }
          if(count == 0) {
            clauseConstraintsComplete.add(extraConstraints);
          }
        }
      }

        for(IVecInt clause : CNFclauses) {
            if(clauseConstraintsComplete.isEmpty()) {
            clauseConstraintsComplete.add(clause);
            }else {
              int count = 0;
              for(IVecInt clauseNew : clauseConstraintsComplete)
                  {
                if(clause.equals(clauseNew)) {
                  count++;
                }
              }
              if(count == 0) {
                clauseConstraintsComplete.add(clause);
              }
            }
      }

      int violatedConstraint = 0;
      for (int i = 0; i < bin.getNumberOfBits(); i++) {
          for (IVecInt clause : clauseConstraintsComplete) {
            boolean isSat = false;
            for (int c = 0; c < clause.size(); c++) {
              int individualclause = clause.get(c);
              int clauseIndex = (individualclause < 0) ? -
                 individualclause : individualclause;
              boolean signal = individualclause > 0;

              if (bin.getIth(clauseIndex - 1) == signal) {
                isSat = true;
                break;
              }
            }
            if (!isSat) {
              violatedConstraint++;
```

```
            }
          }
        }
      return violatedConstraint;
    }
```

Listing 6.5: Violated constraints (GridStixProblem Class)

Figure 6.8: The GA execution adapted from Eiben *et al.* [5]



In **(a)**, we have an initial population of four individuals. They are scored by the objective function (*fitness function*) in **(b)**; the top individual scores a 2 and the bottom scores a 6. It works out that the bottom individual has a 37.5% chance of being chosen on each selection. In **(c)**, selection has given us two pairs of individuals, and the crossover points (dotted lines) have been chosen. Notice that the individual c_4 mates twice. In **(d)**, we see the new offspring, generated by the crossover of their parent's genes. Finally, in **(e)**, the mutation has changed the two bits surrounded by boxes. This gives us the population for the next generation.

The ToffA-DAS+ approach employs the multi-objective algorithm NSGA-II to optimize the selection of features that will conform to the stakeholders preferences and constraints. We chose this GA due to its occurrence in related work that deals with the feature selection problem [2, 137, 138, 139]. Figure 6.8 summarizes the GA to be executed. The basic idea is to randomly generate a set of initial configurations **(a)** and evolve the population through generations aiming to find the best ones, which provide a higher *fitness* **(b)**. Once it is possible to identify the number of valid configurations using the SAT solver technology, we decided to apply this value as an input to define the number of individuals in the initial population and the next generations.

During the execution of GA, configurations are generated randomly by applying three operators [135]. The first operator is the *binary tournament selection* **(c)**. It chooses the best k configurations from the population, uniformly at random and returns the best item from those chosen by following the elitism principle. Thus, the best configuration is returned as a solution to the optimization problem since it provides the highest *fitness*.

This policy always keeps a certain number of best solutions when each new solution is generated [95].

The second operation is reproduction. To exploit the potential of the current chromosome, we use a *single-point crossover* **(d)** to generate new solutions that will retain good features from the previous generations. The crossover follows the natural rule of mating when good parts of each parent solution will combine to make an even better offspring. The method used in the single-point crossover consists of choosing a bit position at random and combining all the bit values below this position from one parent, with all the remaining bit values from the other [95, 129].

The crossover operator exploits current feature potentials, however, whether the selected population does not contain all the encoded information needed to find an optimal or even a good solution, no amount of feature mixing can produce a satisfactory solution. For this reason, the third operation included is a *mutation* [129].

```
/**
 * Perform the mutation operation
 * @param probability Mutation probability
 * @param solution The solution to mutate
 * @throws JMException
 */
public void doMutation(double probability, Solution solution
   ) throws JMException {

  try {
    //This list stores all indexes of variable features
    List<Integer> variableIndex = ManageFeatures.
       getVariableFeaturesIndex();

    if ((solution.getType().getClass() ==
       BinarySolutionTypeNew.class) ||
        (solution.getType().getClass() ==
           BinaryRealSolutionType.class)) {
      for (int i = 0; i < solution.getDecisionVariables().
         length; i++) {

        //Flip only not "fixed" features (variable
           features)
        for(Integer j : variableIndex) {
          if (PseudoRandom.randDouble() < probability) {
            ((Binary) solution.getDecisionVariables()[i]).
               bits_.flip(j);
          }
        }
      }
```

```java

        for (int i = 0; i < solution.getDecisionVariables().length; i++) {
          ((Binary) solution.getDecisionVariables()[i]).decode();
        }
      }
      else { // Integer representation
        for (int i = 0; i < solution.getDecisionVariables().length; i++)
          if (PseudoRandom.randDouble() < probability) {
            int value = PseudoRandom.randInt(
                (int)solution.getDecisionVariables()[i].getLowerBound(),
                (int)solution.getDecisionVariables()[i].getUpperBound());
            solution.getDecisionVariables()[i].setValue(value);
          }
      }
    } catch (ClassCastException e1) {
      Configuration.logger_.severe("BitFlipMutation.doMutation: " +
          "ClassCastException error" + e1.getMessage());
      Class cls = String.class;
      String name = cls.getName();
      throw new JMException("Exception in " + name + ".doMutation()");
    }
  }
```

Listing 6.6: Bit-flip mutation Operator

We adopted the *bit-flip mutation* **(e)** as an operator to flip the bit at the appropriate position, which corresponds to the indexes of variables features. Listing 6.6 presents the algorithm defined to execute such an operation in our optimization problem, whereas Listing 6.7 shows how we manage the variable features aiming to manipulate them during the mutation operation. The action is simultaneously flip a 1 to a 0 and a 0 to a 1 by considering the position of the variable features. It means that all `mandatory` features have their bits fixed with a value equals to 1 and only variable features may have its bit flipped.

```java
public class ManageFeatures {
```

```
  //This list stores the CNF clauses of all features
  private static List<Integer> variableFeaturesIndex = new
     ArrayList<Integer>();

  public void manageVariableFeatures() throws IOException{

    //All features
    Map<String,Integer> featuresIndexNameNew =
       SxFmFTReasoningWithSAT.getAllFeaturesIndexName();

    //Variable features
    List<FeatureDataList> getAllvariableFeatureNew =
       ParserSxFmMain.getAllVariableFeature();

    for (Map.Entry<String,Integer> features :
       featuresIndexNameNew.entrySet()) {
      for(FeatureDataList variable : getAllvariableFeatureNew)
          {
        String featureKey = features.getKey();
        int featureValue = features.getValue();

        if(variable.getVariableFeatureName().equalsIgnoreCase(
           features.getKey())) {
          variableFeaturesMain.put(featureKey, featureValue);
          variableFeaturesIndex.add(featureValue);
        }
      }
    }
  }
}
```

Listing 6.7: ManageFeatures Class

For the current example, the set of features $F = \{\ f_0,\ f_1,\ f_2,\ f_3,\ f_4,\ f_5,\ \neg f_6,\ f_7,\ f_8,\ \neg f_9,\ f_{10},\ \neg f_{11}\}$ was suggested as the feasible and valid configuration resulting from GA execution. Such a solution recommended by the GA solver meets a scenario that represents a specific prioritization of *contexts*, *goals*, and *soft goals* that was mentioned in Chapter 5.

6.3 EVALUATION OF THE TOFFA-DAS+ APPROACH

After evolving the ToffA-DAS approach by applying the algorithm NSGA-II as an optimization method for dealing with configuration selection conflicts, we decided to compare it with the previous release that applies the ILP technique as an optimization method. In the following subsections, we present the exploratory study that was organized following the guide proposed by Wohlin *et al.* [122].

6.3.1 Exploratory study definition

NFRs (*soft goals*) are strictly related to the information provided by the system's features (*hard goals*) and typically set constraints for them [120]. Since both releases of the ToffA-DAS approach consider *soft goal* priority in their configuration process of DAS, the satisfaction level between *hard goals* and *soft goals* was used as a criterion for the assessment presented in this chapter.

Thus, the study aimed at **analyzing** both releases of our approach, ToffA-DAS and ToffA-DAS+ **for the purpose of** evaluating the resulting configurations obtained from the configuration process **with respect to** the overall satisfaction level between *hard goals* and *soft goals* **from the point of view of** Software Engineers and Researchers **in the context of** two DAS. Based on the study's goal, we defined the following research question for this assessment:

RQ. **Do the configurations generated by the ToffA-DAS approach provide higher satisfaction levels of soft goals than those generated by the ToffA-DAS+ execution?**

In order to make a fair comparison between ToffA-DAS and ToffA-DAS+, we used in this evaluation three metrics. Such metrics are based on the negative and positive contributions that influences the satisfaction level of a *soft goal*. The metrics used are presented as follows:

- **Pos** - This metric calculates the number of positive contributions to the *soft goals*;
- **Neg** - This metric calculates the number of negative contributions to the *soft goals*; and
- **Diff** - This metric calculates the difference between the number of positive and negative contributions to the *soft goals* (Pos-Neg). It aims to identify whether the release presented more positive or negative contributions of *soft goal*.

6.3.2 Exploratory study planning

This section discusses the planning and the procedures to be followed in order to perform the exploratory study. For this study, we selected two DAS presented in the literature named *Mobile game* [2] and *Smart Home* [121], respectively. These DASs were also used in the evaluation presented by Guedes *et al.* [107], which was the basis for identifying the metrics to be employed in the evaluation of this paper (see artifacts of the exploratory study in appendix B). The following subsections present the procedures used and the hypothesis defined.

6.3.2.1 Quantitative analysis mechanisms - The exploratory study followed the activities presented in Figure 6.1: *(i)* identification of domain knowledge; *(ii)* modeling of

DAS; *(iii)* definition of the prioritization of *goals*, *soft goals*, and *contexts*; *(iv)* measurement of the impact of features over *goals*, *soft goals*, and *contexts*; *(vi)* trade-off analysis, which consists of simulating changes in the prioritization of *goals*, *soft goals*, and *contexts*; and *(vi)* identification of the feasible and valid configurations during the simulations. In this latter, we collected the metrics **Pos**, **Neg**, and **Diff**.

Aiming to perform the simulations, we executed the configuration selection process presented in Section 6.2. For each simulation, software engineers must only consider relationships between the systems feature and context corresponding to a specific CCF. Each CCF must be based on the relationship between **context feature** and its respective **context group**. Likewise, the software engineer must only consider the relationships between *hard goals* and *soft goals* corresponding to a certain CCF. From a specific CCF, it is possible to define dynamic adaptation models. These models show how DAS can evolve from one CCF to another changing its respective feasible configuration.

In this study, we defined the CCFs presented in Table 6.1 and Table 6.2. In addition, the metrics of the evaluation are described in terms of a mapping link between *hard goals* and *soft goals*: satisfied (++) = 1, weakly satisfied (+) = 0.5, undecided (?) = 0, weakly denied (-) = -0.5, and denied (- -) = -1. We collected the results of the metrics based on the number of positive and negative satisfaction levels of *soft goals* over *hard goals* for all configurations obtained in the execution of both releases of the approach, ToffA-DAS and ToffA-DAS+.

We considered the number of individuals of the initial population and the next generations equal to the number of valid configurations identified by the SAT solver. In addition, we used the number of evaluations as a stop criterion for the algorithm, which is equal to three times the population size. After executing the algorithm, we analyzed the valid and more adapted configurations (among all suggested by the approach) according to *fitness*. In this study, we did not take into account the execution time since our objective is not to compare both algorithms in terms of performance.

For quantitative data, the analysis included descriptive statistics, such as *median values* and *box plot* aiming to explore the gathered data. Regarding the hypotheses defined for the exploratory study, the non-parametric *Wilcoxon Signed-rank Test* was used [122, 123]. This test was chosen because the study employs two related samples and it yields difference scores that may be ranked in order of absolute size. Indeed, it determines which of the measures in pair is the greatest and ranks the differences. In this sense, it gives more weight to a pair which shows a large difference between the two conditions than to a pair that shows a small difference. In addition, it shows the sign of the difference between any pair and ranks the differences in the order of absolute size [124].

6.3.2.2 Hypothesis - Null Hypotheses. The null hypotheses state that there is no difference between ToffA-DAS and ToffA-DAS+ in terms of **Pos**, **Neg** and **Diff**. The corresponding null hypotheses are presented as follows:

- $H_{01} : Pos_{toffa} = Pos_{toffa+}$
- $H_{02} : Neg_{toffa} = Neg_{toffa+}$

- $H_{03} : Diff_{toffa} = Diff_{toffa+}$

Alternative Hypotheses. The alternative hypotheses state that there is a difference between ToffA-DAS and ToffA-DAS+ in terms of **Pos**, **Neg** and **Diff**. The corresponding alternative hypotheses are presented as follows:

- $H_{11} : Pos_{toffa} \neq Pos_{toffa+}$
- $H_{12} : Neg_{toffa} \neq Neg_{toffa+}$
- $H_{13} : Diff_{toffa} \neq Diff_{toffa+}$

6.3.3 Analysis and interpretation

This section provides an in-depth analysis of the gathered data. We discuss the results in terms of satisfaction level of *soft goals* for the configuration process of both ToffA-DAS and ToffA-DAS+ releases. Moreover, it discusses the hypothesis testing.

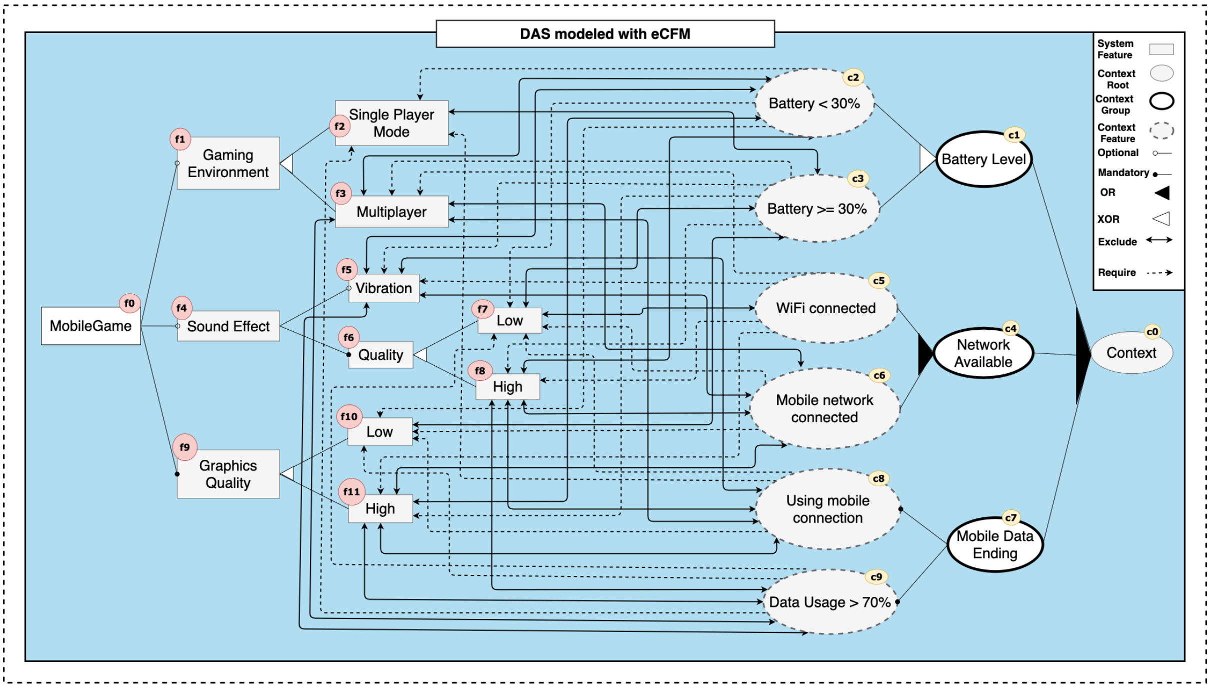
6.3.3.1 DAS in the mobile domain - From the models and information presented by Pascual *et al.* [2], we designed the corresponding eCFM and goal model. The eCFM is composed of eleven features and nine contexts. The goal model one is composed of four *goals*, seven *hard goals*, and two *soft goals*. Figures 6.9 and 6.10 present the eCFM and goal model of the *Mobile Game*, respectively. Once the models were finished, we simulated eight scenarios corresponding to CCFs ccf_1, ccf_3, ccf_7, ccf_8, ccf_9, ccf_{11}, ccf_{15}, and ccf_{16}. For all of them, we kept the same priority for contexts and *goals* (*i.e.*, priority equal to one), besides considering for *soft goals*, the priority presented in Table 6.1. For instance, the ILP solver in ToffA-DAS suggested that the variable features f_3, f_5, f_8 and f_{11} satisfy ccf_3. For this scenario, the prioritization of *soft goals* is $sg_1 > sg_2$.

Table 6.1: Scenarios for Mobile Game DAS by considering the valid CCFs and prioritization of *soft goals* presented in the first and fourth columns of the table. After executing the simulations, we collected the metrics **Pos**, **Neg**, and **Diff** for each configuration suggested in second and third columns. The scenario related to ccf_1 does not consider any relationship (`require` or `exclude`) between the system's feature and **context feature**.

Results about configurations (Variable features)			Results about metrics						
CCFs	ToffA-DAS	ToffA-DAS+	Soft goal Priority	ToffA-DAS			ToffA-DAS+		
				Pos	Neg	Diff	Pos	Neg	Diff
$ccf_1 = \{none\}$	f_3,f_5,f_7,f_{10}	f_2,f_7,f_{11}	$sg_1 > sg_2$	4	4	0	2	3	-1
$ccf_3 = \{c_5, c_6\}$	f_3,f_5,f_8,f_{11}	f_7,f_{10}	$sg_1 > sg_2$	4	4	0	2	2	0
$ccf_7 = \{c_5, c_6, c_8\}$	f_3,f_5,f_7,f_{10}	f_3,f_7,f_{11}	$sg_1 = sg_2 = 1$	4	4	0	3	3	0
$ccf_8 = \{c_5, c_6, c_8, c_9\}$	f_2,f_5,f_7,f_{10}	f_3, f_5, f_8,f_{10}	$sg_2 > sg_1$	3	4	-1	4	4	0
$ccf_9 = \{c_2\}$	f_2,f_5,f_7,f_{10}	f_8,f_{11}	$sg_2 > sg_1$	4	4	-1	2	2	0
$ccf_{11} = \{c_2, c_5, c_6\}$	f_2,f_5,f_7,f_{10}	f_8,f_{11}	$sg_2 > sg_1$	3	4	-1	1	2	-1
$ccf_{15} = \{c_2, c_5, c_6\}$	f_2,f_5,f_7,f_{10}	f_2,f_8,f_{11}	$sg_2 > sg_1$	3	4	-1	2	3	-1
$ccf_{16} = \{c_2, c_3, c_5, c_6, c_8, c_9\}$	f_2,f_5,f_7,f_{10}	f_8,f_{11}	$sg_2 > sg_1$	3	4	-1	2	2	0

Observing the results, we notice that Toffa-DAS and ToffA-DAS+ suggested different configurations for the CCFs. In the scenario corresponding to ccf_1, for instance, ToffA-DAS suggested features f_3, f_5, f_7, and f_{10} to be inserted in the feasible configuration.

Figure 6.9: Mobile Game - eCFM



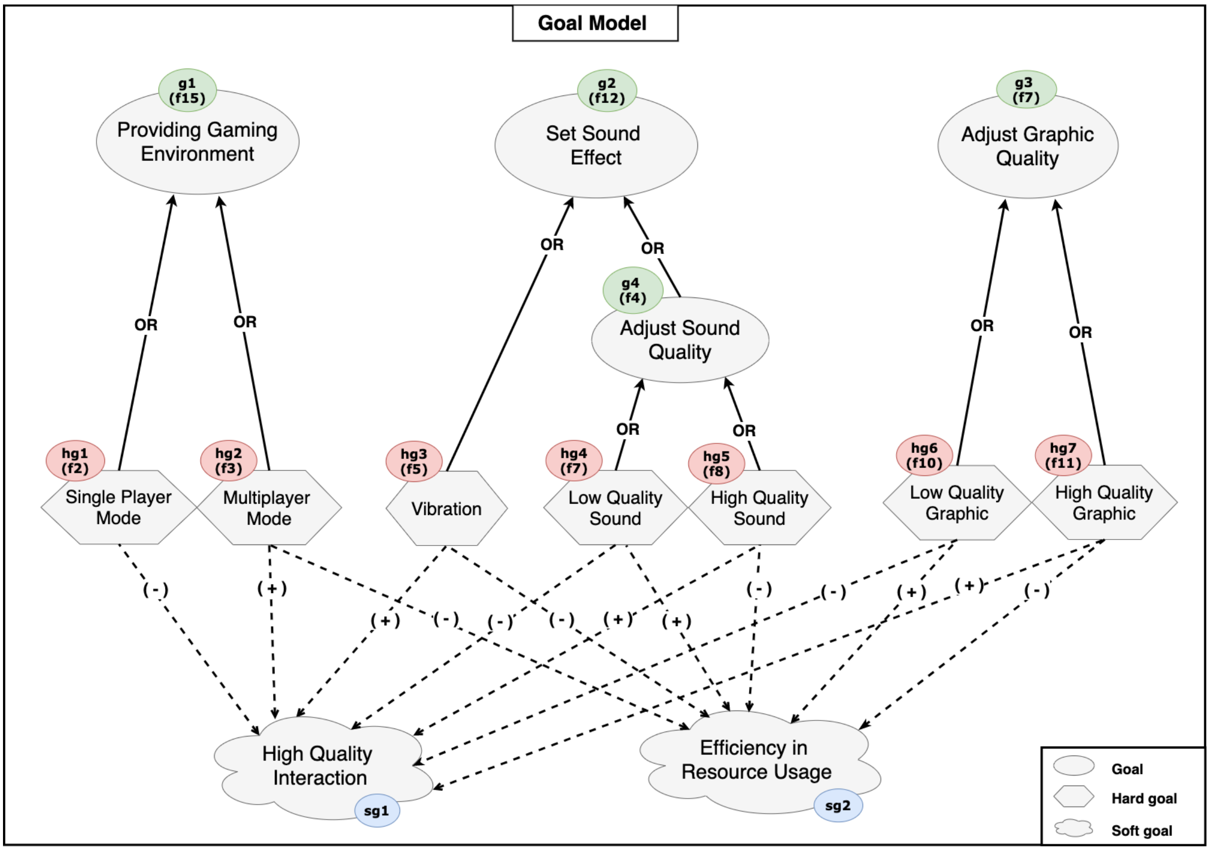
For the same scenario, ToffA-DAS+ suggested other set of features, such as f_2, f_7 and f_{11}. The same is true for other CCFs. In the remaining scenarios, there is at least one equivalent feature in the configuration pairs resulting from the execution of both releases.

In general, we observed that the most frequent feature in ToffA-DAS was f_5 appearing in all configurations. By using ToffA-DAS+, such a feature was present only in scenario ccf_8. Addtionally, we can see that ToffA-DAS suggested the feature f_{10} in 99% of the configurations (see scenarios ccf_1, ccf_7, ccf_8, ccf_9, ccf_{11}, ccf_{15}, and ccf_{16}), whereas ToffA-DAS+ suggested f_{11} in 75% of the configurations (see scenarios ccf_1, ccf_7, ccf_8, ccf_9, ccf_{11}, ccf_{15}, and ccf_{16}). Indeed, the most frequent features in ToffA-DAS were f_5, f_7, and f_{10}. It explains why the configurations resulting from execution of ToffA-DAS presented a greater number of negative contributions of *soft goals* in comparison with ToffA-DAS+.

We also measured the number of positive and negative satisfaction levels of *soft goals* over *hard goals* for all configurations obtained in each release. Table 6.1 shows the results concerning the **Pos**, **Neg**, and **Diff** measures. We applied the *Wilcoxon Test* to assess the null hypothesis presented in Section 6.3.2.2. When using such a test, we must calculate the sum of the positive ranks (T^+) and the sum of the negative ranks (T^-), besides observing whether the pairs of data have a score difference different to zero. When a pair of data has a score difference equal to zero, it is removed from the analysis. Then, the number of pairs N to be considered is equal to the total number of pairs minus any pairs whose difference is zero.

We removed from the analysis, the pairs POS_{toffa} and POS_{toffa+} that have a score difference equal to zero. As a result, we obtained $T^+ = 33.5$ and $T^- = -2.5$. With the

Figure 6.10: Mobile Game - Goal Model



sample with $N = 8$, the critical value $W_{critical}$ at *p-value* ≤ 0.05 is 3 and $W_{start} = 2.5$. Since $W_{start} < W_{critical}$, the null hypothesis H_{01} is rejected. It means that there is statistical significance difference between POS_{toffa} and POS_{toffa+} for the Mobile Game DAS. Figure 6.11 depicts the box plot concerning the **Pos** metric. For this measure, the *median* value in ToffA-DAS was equal to 3.5, whereas in ToffA-DAS+ such value was equal to 2. In general, the number of positive contributions of *soft goals* in the first release ranged between 3 and 4. In the second release this variance of the data set was between 1 and 4. In addition, ToffA-DAS+ presented two outliers, which are related to a greater **Pos** value in the scenario ccf_8 and lower **Pos** value in the scenario ccf_{11}, respectively.

Regarding the **Neg** metric, we also removed the pairs Neg_{toffa} and Neg_{toffa+} that have a score difference equal to zero. The sum of the negative and positive ranks presented values $T^+ = 35$, $T^- = 0$, respectively. When $N = 7$, the critical value $W_{critical}$ at $p - value \leq 0.05$ is 2 and the test statistic W_{start} equal to 0. Since $W_{start} < W_{critical}$, the hypothesis null H_{02} is rejected. It means that there is statistical significance difference between Neg_{toffa} and Neg_{toffa+} for the Mobile Game DAS. Figure 6.12 depicts the box plot concerning the **Neg** metric. For this measure in ToffA-DAS release, the *median* value was equal to 4 and there was no variance of the data set. It indicates that such scenarios had a similar number of negative contributions to *soft goals*. In contrast, in the

Figure 6.11: Pos Metric - Mobile Game

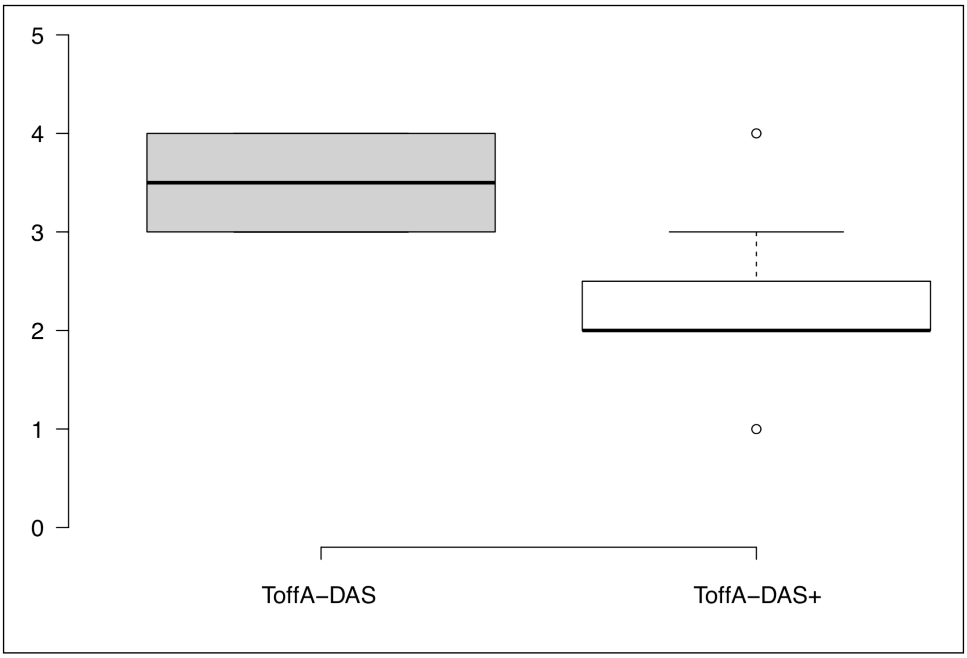
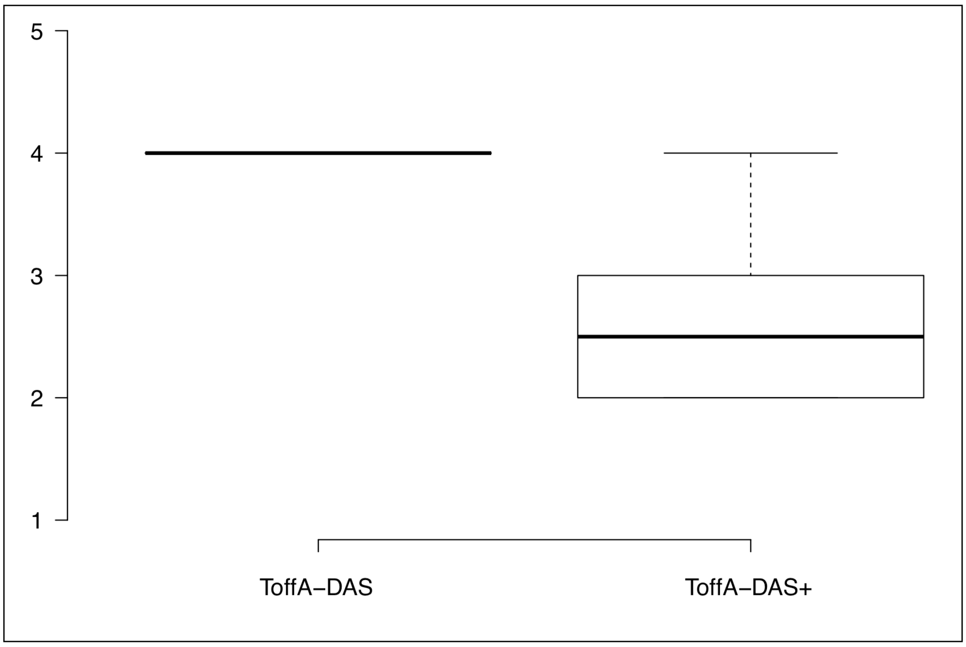


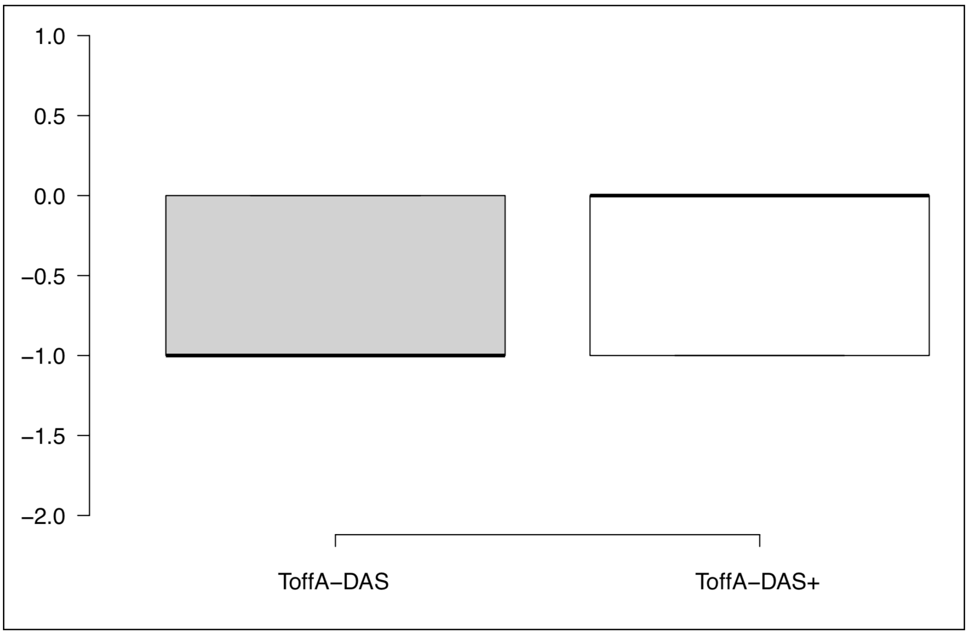
Figure 6.12: Neg Metric - Mobile Game



ToffA-DAS+ release, there was a greater variance of the data set, besides of a *median* value equal to 2.5. Only the scenario ccf_8 presented a number of negative contributions to the *soft goals* equal to 4.

Finally, for the **Diff** metric, after removing the pairs $Diff_{toffa}$ and $Diff_{toffa+}$ that have a score difference equal to zero, we obtained $T^+ = 6.5$ and $T^- = 0$. With the

Figure 6.13: Diff Metric - Mobile Game



sample with $N = 4$, it is not enough to return the critical value of $W_{critical}$ with the level of significance *p-value* ≤ 0.05. Thus, the hypothesis null H_{03} cannot be rejected, meaning that there is no statistically significant difference between the $Diff_{toffa}$ and $Diff_{toffa+}$ for the Mobile Game DAS. Figure 6.13 depicts the box plot concerning the **Diff** metric. For this metric, the *median* value in ToffA-DAS was equal to -1, whereas in ToffA-DAS+ such value was equal to 0. In general, the variation of the data set to both releases was similar.

In summary, ToffA-DAS presented a greater number of positive contributions to the *soft goals* than the ToffA-DAS+. The first release also presented a greater number of negative contributions to the *soft goals* than second one. Based on the analysis, we can conclude that ToffA-DAS+ provided a more balanced result to the satisfaction level of *soft goals* considering the scenarios defined for Mobile Game DAS.

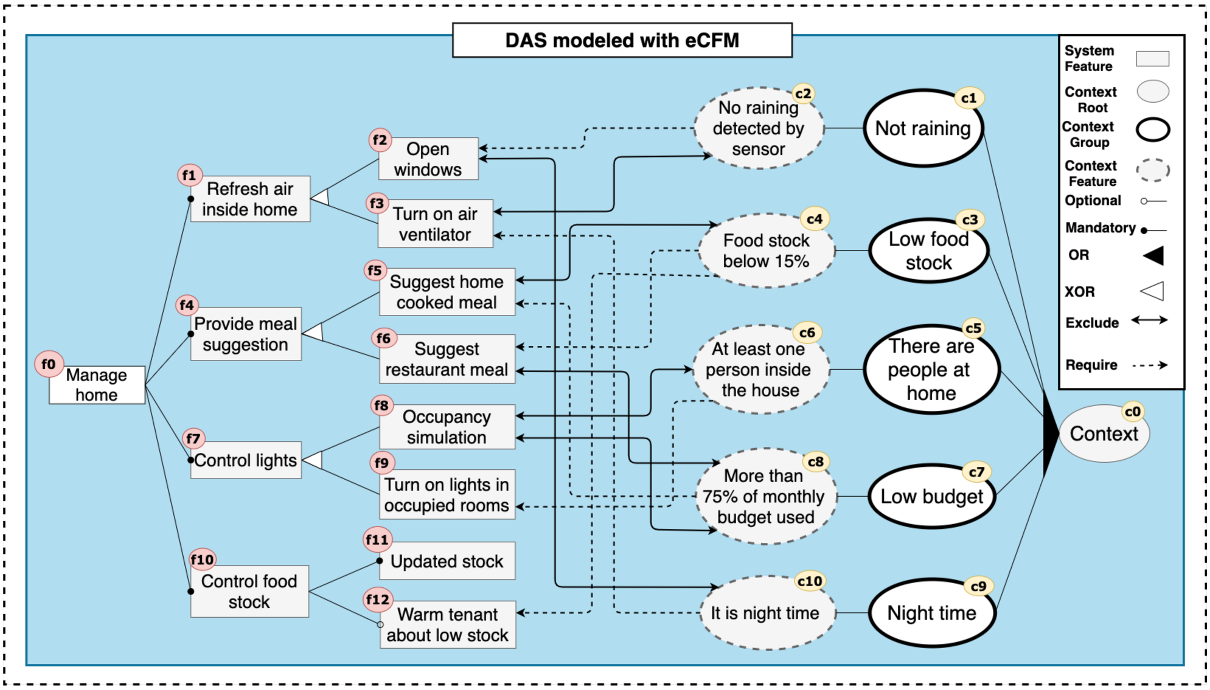
6.3.3.2 DAS in the smart home domain - From the models and information presented by Pimentel *et al.* [121], we designed the corresponding eCFM and goal model. The former is composed of thirteen features and eleven contexts. The second one is composed of three *goals*, six *hard goals*, and three *soft goals*. Figures 6.14 and 6.15 present the eCFM and goal model of the *Smart Home*, respectively. We simulated thirty-two scenarios corresponding to CCFs from ccf_1 to ccf_{32}. For all of them, we kept the same priority for contexts and *goals* (*i.e.*, priority equal to one), besides considering for *soft goals*, the priority presented in Table 6.2. The table also shows the resulting feasible configurations from simulations. For instance, the ILP solver in ToffA-DAS suggested that the variable features are inserted into the configuration, as follows:

- Features f_3, f_5, and f_9 satisfy ccf_1, ccf_2, ccf_3, ccf_4, ccf_5, ccf_6, ccf_7, ccf_8, ccf_{16},

ccf_{18} and ccf_{22};

- Features f_3, f_6, f_9, and f_{12} satisfy ccf_9, ccf_{10}, ccf_{12}, ccf_{13}, ccf_{14}, ccf_{26}, and ccf_{30};
- Features f_2, f_5, and f_9 satisfy ccf_{17}, ccf_{19}, ccf_{20}, ccf_{21}, ccf_{23}, and ccf_{24};
- Features f_2, f_6, and f_9 satisfy ccf_{15} and ccf_{17};
- Features f_2, f_6, f_9, and f_{12} satisfy ccf_{11}, ccf_{25}, ccf_{27}, ccf_{28}, ccf_{29}, ccf_{31}, and ccf_{32}.

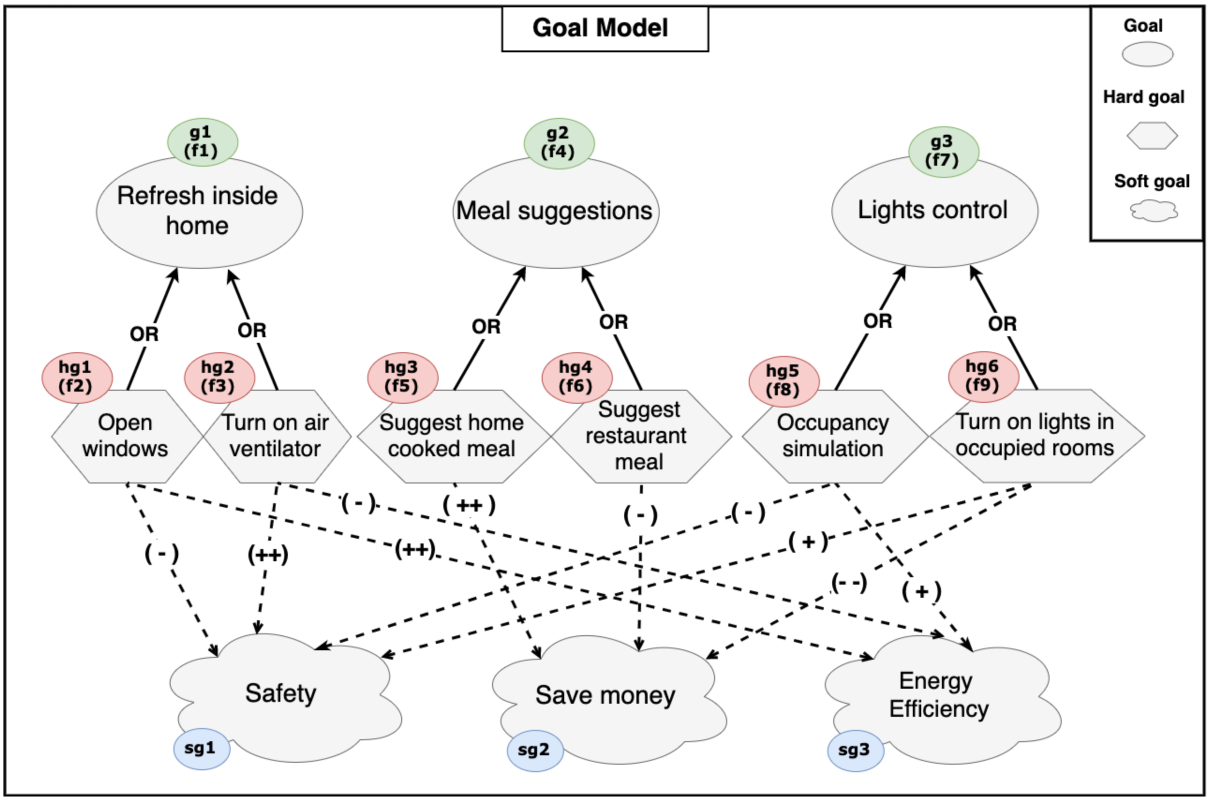
Figure 6.14: Smart Home - eCFM



ToffA-DAS returned four different configurations for the CCfs that were taken into consideration at the simulations. Conversely, in ToffA-DAS+ the number of configurations increased to eleven ones. The set of such configurations are presented as follow:

- Features f_2, f_6, and f_8 satisfy ccf_1, ccf_6, ccf_8, ccf_{20}, ccf_{24}, and ccf_{28};
- Features f_2, f_6, ccf_8, and f_{12} satisfy ccf_2, ccf_4, ccf_5, ccf_7, ccf_9, ccf_{29}, and ccf_{30} ;
- Features f_2, f_6, f_9, and f_{12} satisfy ccf_3;
- Features f_2, f_5, f_8, and f_{12}, satisfy ccf_{10};
- Features f_3, f_5, and f_8 satisfy ccf_6, ccf_{11}, ccf_{15}, ccf_{25}, and ccf_{31};
- Features f_3, f_6, and f_9 satisfy ccf_{12} and ccf_{27};

Figure 6.15: Smart Home - Goal Model



- Features f_2, f_5, and f_8 satisfy ccf_{13}, ccf_{14}, and ccf_{16};
- Features $f3$, f_6, f_8, and f_{12} satisfy ccf_{10} and ccf_{16};
- Features f_3, f_6, and f_8 satisfy ccf_{18} and ccf_{26};
- Features f_2, f_5, and f_9 satisfy ccf_{17};
- Features f_3, f_5, and f_9 satisfy ccf_{32}.
- Features f_3, f_5, f_8, and f_{12} satisfy ccf_{21}.

The ILP solver in ToffA-DAS suggested that the variable features f_3, f_5, and f_9 satisfy ccf_1, whereas the GA in ToffA-DAS+ suggested the features f_2, f_6, and f_8. For this scenario, the prioritization of *soft goals* is $sg_2 > sg_1 > sg_3$. In general, we observed that the most frequent feature in ToffA-DAS was f_9 appearing in all configurations. By using ToffA-DAS+, in turn, the most frequent feature was f_8, which was present in twenty eight configurations. This explains why some configurations resulting from the execution of the ToffA-DAS presented a greater number of negative contributions of *soft goals* in comparison with the ToffA-DAS+. In scenarios ccf_1, ccf_5, ccf_9, ccf_{13}, ccf_{18}, ccf_{22}, ccf_{26}, and ccf_{30} the *utility value* for features f_2 and f_3 was equal to 0.6. It means that any of

Table 6.2: Scenarios for Smart Home DAS by considering the valid CCFs and prioritization of *soft goals* presented in the first and fourth columns of the table. After executing the simulations, we collected the metrics **Pos**, **Neg**, and **Diff** for each configuration suggested in second and third columns. The scenario related to ccf_1 does not consider any relationship (`require` or `exclude`) between the system's feature and **context feature**.

Results about configurations (Variable features)			Results about metrics						
CCFs	ToffA-DAS	ToffA-DAS+	Soft goal Priority	ToffA-DAS			ToffA-DAS+		
				Pos	Neg	Diff	Pos	Neg	Diff
$ccf_1 = \{none\}$	f_3,f_5,f_9	f_2,f_6,f_8	$sg_2 > sg_1 > sg_3$	5	3	2	3	3	0
$ccf_2 = \{c_{10}\}$	f_3,f_5,f_9	f_2,f_6,f_8	$sg_2 > sg_1 > sg_3$	5	3	2	3	3	0
$ccf_3 = \{c_8\}$	f_3,f_5,f_9	f_2,f_6,f_9,f_{12}	$sg_1 > sg_3 > sg_2$	5	3	2	3	4	-1
$ccf_4 = \{c_8, c_{10}\}$	f_3,f_5,f_9	f_2,f_6,f_8	$sg_1 > sg_3 > sg_2$	5	3	2	3	3	0
$ccf_5 = \{c_6\}$	f_3,f_5,f_9	f_2,f_6,f_8	$sg_2 > sg_1 > sg_3$	5	3	2	3	3	0
$ccf_6 = \{c_6, c_{10}\}$	f_3,f_5,f_9	f_2,f_6,f_8	$sg_2 > sg_1 > sg_3$	5	3	2	3	3	0
$ccf_7 = \{c_9, c_8\}$	f_3,f_5,f_9	f_2,f_6,f_8	$sg_1 > sg_3 > sg_2$	5	3	2	3	3	0
$ccf_8 = \{c_6, c_8, c_{10}\}$	f_3,f_5,f_9	f_2,f_6,f_8	$sg_1 > sg_3 > sg_2$	5	3	2	3	3	0
$ccf_9 = \{c_4\}$	f_3,f_6,f_9,f_{12}	f_2,f_6,f_8	$sg_2 > sg_1 > sg_3$	3	4	-1	3	3	0
$ccf_{10} = \{c_4, c_{10}\}$	f_3,f_6,f_9,f_{12}	f_2,f_5,f_8,f_{12}	$sg_2 > sg_1 > sg_3$	3	4	-1	5	2	3
$ccf_{11} = \{c_4, c_8\}$	f_2,f_6,f_9,f_{12}	f_3,f_5,f_8	$sg_1 > sg_3 > sg_2$	3	4	-1	5	2	3
$ccf_{12} = \{c_4, c_8, c_{10}\}$	f_3,f_5,f_9,f_{12}	f_3,f_6,f_9	$sg_1 > sg_3 > sg_2$	5	3	2	3	4	-1
$ccf_{13} = \{c_4, c_6\}$	f_3,f_6,f_9,f_{12}	f_2,f_5,f_8	$sg_2 > sg_1 > sg_3$	3	4	-1	5	2	3
$ccf_{14} = \{c_4, c_6, c_{10}\}$	f_3,f_6,f_9,f_{12}	f_2,f_5,f_8	$sg_2 > sg_1 > sg_3$	3	4	-1	5	2	3
$ccf_{15} = \{c_4, c_6, c_8\}$	f_2,f_6,f_9	f_3,f_5,f_8	$sg_1 > sg_3 > sg_2$	3	4	-1	5	2	3
$ccf_{16} = \{c_4, c_6, c_8, c_{10}\}$	f_3,f_5,f_9	f_2,f_5,f_8	$sg_1 > sg_3 > sg_2$	5	3	2	5	2	3
$ccf_{17} = \{c_1\}$	f_2,f_6,f_9	f_3,f_6,f_8,f_{12}	$sg_2 > sg_1 > sg_3$	3	4	-1	3	3	0
$ccf_{18} = \{c_1, c_{10}\}$	f_3,f_5,f_9	f_3,f_6,f_8	$sg_2 > sg_1 > sg_3$	5	3	2	3	3	0
$ccf_{19} = \{c_1, c_8\}$	f_2,f_5,f_9	f_3,f_6,f_8,f_{12}	$sg_1 > sg_3 > sg_2$	5	3	2	3	3	0
$ccf_{20} = \{c_1, c_9, c_{10}\}$	f_2,f_5,f_9	f_2,f_6,f_8	$sg_1 > sg_3 > sg_2$	5	3	2	3	3	0
$ccf_{21} = \{c_1, c_6\}$	f_2,f_5,f_9	f_3,f_5,f_8	$sg_2 > sg_1 > sg_3$	5	3	2	5	2	3
$ccf_{22} = \{c_1, c_6, c_{10}\}$	f_3,f_5,f_9	f_3,f_6,f_8,f_{12}	$sg_2 > sg_1 > sg_3$	5	3	2	3	3	0
$ccf_{23} = \{c_1, c_6, c_8\}$	f_2,f_5,f_9	f_3,f_6,f_8,f_{12}	$sg_1 > sg_3 > sg_2$	5	3	2	3	3	0
$ccf_{24} = \{c_1, c_6, c_8, c_{10}\}$	f_2,f_5,f_9	f_2,f_6,f_8	$sg_1 > sg_3 > sg_2$	5	3	2	3	3	0
$ccf_{25} = \{c_1, c_4\}$	f_2,f_6,f_9,f_{12}	f_3,f_5,f_8	$sg_2 > sg_1 > sg_3$	3	4	-1	5	2	3
$ccf_{26} = \{c_1, c_4, c_{10}\}$	f_3,f_6,f_9,f_{12}	f_3,f_6,f_8	$sg_2 > sg_1 > sg_3$	3	4	-1	3	3	0
$ccf_{27} = \{c_1, c_4, c_8\}$	f_2,f_6,f_9,f_{12}	f_3,f_6,f_9	$sg_1 > sg_3 > sg_2$	3	4	-1	3	4	-1
$ccf_{28} = \{c_1, c_4, c_8, c_{10}\}$	f_2,f_6,f_9,f_{12}	f_2,f_6,f_8	$sg_1 > sg_3 > sg_2$	3	4	-1	3	3	0
$ccf_{29} = \{c_1, c_4, c_6\}$	f_2,f_6,f_9,f_{12}	f_2,f_6,f_8	$sg_2 > sg_1 > sg_3$	3	4	-1	3	3	0
$ccf_{30} = \{c_1, c_4, c_6, c_{10}\}$	f_3,f_6,f_9,f_{12}	f_2,f_6,f_8	$sg_2 > sg_1 > sg_3$	3	4	-1	3	3	0
$ccf_{31} = \{c_1, c_4, c_6, c_8\}$	f_2,f_6,f_9,f_{12}	f_3,f_5,f_8	$sg_1 > sg_3 > sg_2$	3	4	-1	5	2	3
$ccf_{32} = \{c_1, c_4, c_6, c_8, c_{10}\}$	f_2,f_6,f_9,f_{12}	f_3,f_5,f_9	$sg_1 > sg_3 > sg_2$	3	4	-1	4	3	1

them could be selected by ToffA-DAS and ToffA-DAS+. However, ToffA-DAS suggested the first feature that was detected during analysis, whereas ToffA-DAS+ suggested the feature that influences the configuration to provide a more balanced satisfaction level of *soft goals*.

We also measured the number of positive and negative satisfaction levels of *soft goals* over *hard goals* for all configurations obtained in each release. Table 6.2 shows the results concerning the **Pos**, **Neg**, and **Diff** metrics. In addition, we applied the *Wilcoxon Test* to assess the null hypothesis presented in Section 6.3.2.2. When removing from the analysis the pairs POS_{toffa} and POS_{toffa+} that have a score difference equal to zero, we obtained $T^+ = 315$ and $T^- = -177$. The sample presented a value of $N = 24$,

Figure 6.16: Pos Metric - Smart Home

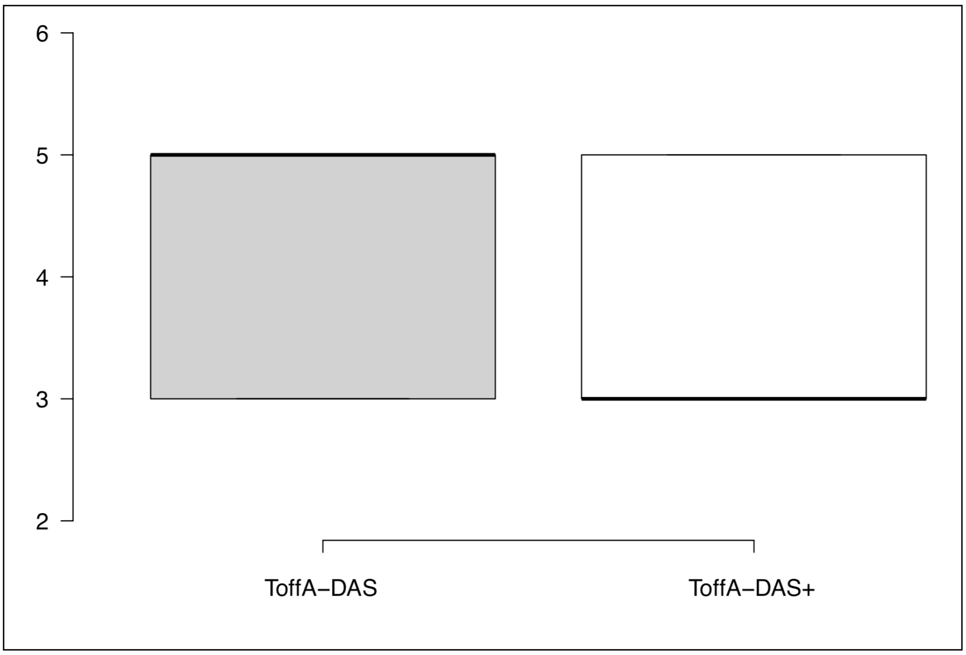
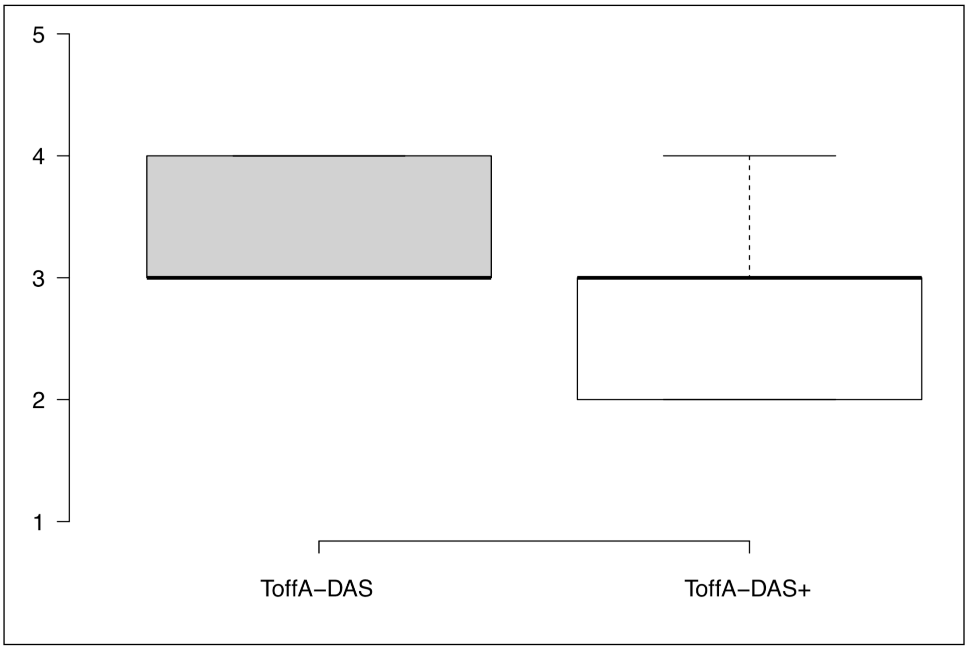
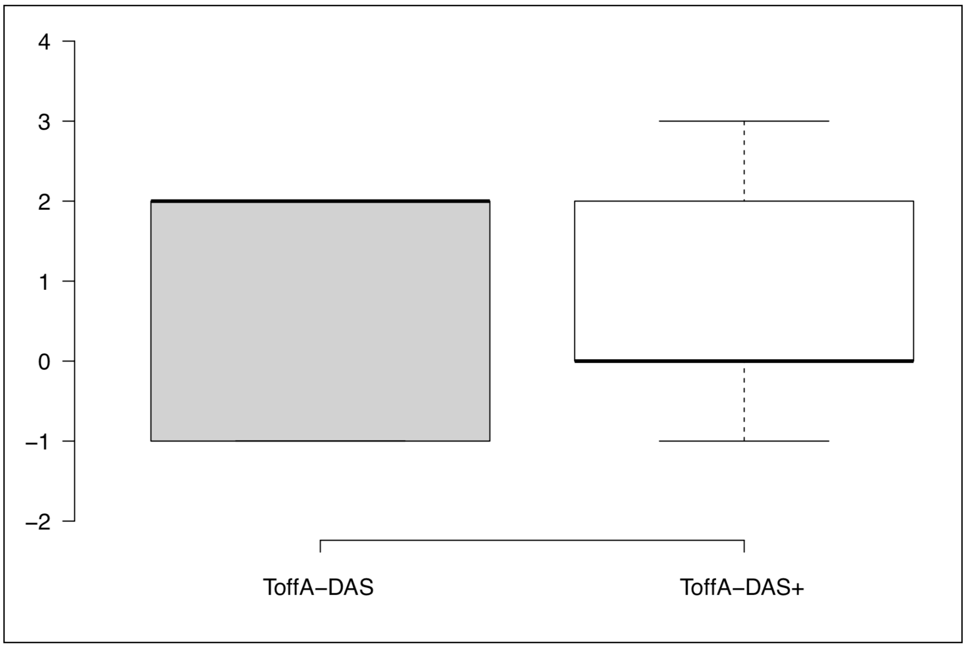


Figure 6.17: Neg Metric - Smart Home



the critical value $W_{critical}$ at $p - value \leq 0.05$ equal to 81, and the test statistic W_{start} equal to 177. Since $W_{start} > W_{critical}$, the hypothesis null H_{01} is rejected. It means that there is statistical significance difference between POS_{toffa} and POS_{toffa+} for the *Smart Home* DAS. Figure 6.16 depicts the box plot concerning the **Pos** metric. For this metric, the *median* value in ToffA-DAS was equal to 5, whereas in ToffA-DAS+ such

Figure 6.18: Diff Metric - Smart Home



value was equal to 3. Both releases presented a similar variance of the data set, *i.e.* the number of positive contributions of *soft goals* to configurations resulting from both releases ranged between 3 and 5. However, ToffA-DAS presented a **Pos** value equal to 5 in 51% of the configurations, whereas ToffA-DAS+ presented such value only in 28% of the configurations.

Regarding the **Neg** metric, we also removed the pairs Neg_{toffa} and Neg_{toffa+} that have a score difference equal to zero. The sum of the negative and positive ranks presented values $T^{+} = 383$, $T^{-} = -40$, respectively. When $N = 18$, the critical value $W_{critical}$ at $p - value \leq 0.05$ is equal to 40 and the test statistic W_{start} is equal to 40. Since $W_{start} = W_{critical}$, the hypothesis null H_{02} is not rejected. It means that there is no statistical significance difference between Neg_{toffa} and Neg_{toffa+} for the *Smart Home* DAS. Figure 6.17 depicts the box plot concerning the **Neg** metric. For this metric, the *median* value was equal to 3 in both releases. In general, the variation of the data set to ToffA-DAS+ was higher than in ToffA-DAS. However, ToffA-DAS presented a greater number of negative contributions of *soft goals*.

Finally, for the **Diff** metric, after removing the pairs $Diff_{toffa}$ and $Diff_{toffa+}$ that have a score difference equal to zero, we obtained $T^{+} = 279$, $T^{-} = -248.5$. For the observed value of $N = 31$, the critical value $W_{critical}$ at $p - value \leq 0.05$ is equal to 147 and the test statistic W_{start} is equal to 248. Since $W_{start} > W_{critical}$, the hypothesis null H_{03} is rejected. It means that there is statistical significance difference between $Diff_{toffa}$ and $Diff_{toffa}$ for the *Smart Home* DAS. Figure 6.18 depicts the box plot concerning the **Diff** metric. For this metric, the *median* value was equal to 2 in ToffA-DAS and equal to 0 in ToffA-DAS+. In general, the variation of the data set in ToffA-DAS was lower than in the ToffA-DAS+.

In summary, the metric measures in ToffA-DAS show a high number of negative and positive contributions of *soft goals* in comparison with ToffA-DAS+. Based on the analysis, we can conclude that ToffA-DAS+ provided a more balanced result to the satisfaction level of *soft goals* considering the scenarios defined for *Smart Home* DAS.

6.4 DISCUSSION

In this section, we discuss the results found in this exploratory study. Next, we present threats to the validity of the results.

6.4.1 Exploratory study

In the exploratory study, we assessed the ToffA-DAS and ToffA-DAS+ releases using both DAS, *Mobile Game* and *Smart Home*. The size of their goal model was similar, but the size of their eCFM is quite different. The *Mobile Game*'s eCFM has twelve features and ten contexts, whereas the *Smart Home*'s eCFM is composed of thirteen features and eleven contexts. In order to perform the simulations, we considered eight valid CCFs for *Mobile Game* and thirty-two valid CCFs for *Smart Home*.

Regarding the metrics in *Mobile Game*, only the null hypothesis for **Diff** measures was not rejected. In this case, the sample was not enough to return the critical value with the level of significance of $p-value \leq 0.05$. The same is true for the **Neg** measures in *Smart Home* DAS. For the remaining metrics in both DAS, the null hypotheses were rejected. It means that there is sufficient evidence to suggest a difference between ToffA-DAS and ToffA-DAS+ in terms of such measurements considering the resulting configurations.

Furthermore, the set of configurations varied in each release of the approach. When the ToffA-DAS was applied, we obtained four different configurations in *Mobile Game* and five in *Smart Home*. By using ToffA-DAS+, the set of configurations rise to six in *Mobile Game* and twelve in *Smart Home*. Such a result is due to the way in which the optimization methods deal with the feature selection. In addition, the prioritization of *soft goals* affected the feasible configuration selection in both releases. The prioritization of *goals* was equal to one in all scenarios, then it did not directly influence the choice of variable features. The prioritization of contexts also was equal to one in all scenarios, however, the impact degree representing how each feature can satisfy a **context feature** was considered. This impact degree is based on the `require` and `exclude` relationships.

In ToffA-DAS, we defined only one objective function measuring the decision variables summation by satisfying the *contexts*, *goals*, and *soft goals*. Conversely, we considered three objective functions in ToffA-DAS+ to measure the decision variables summation by satisfying such elements, separately. Observing the *fitness* of each configuration suggested, we noticed that the way of how the objective functions were defined in each release is not the reason for different results. It means that in ToffA-DAS+, we can also consider only one objective function as was made in ToffA-DAS. The strategies used to evaluate the constraints in both also did not affect the results. However, in ToffA-DAS+ such a satisfiability analysis was simplified by employing the SAT solver technology. Thus, we concluded that the manner in which the optimization problem was defined both releases is not a factor to produce different results. The main reason for the different results suggested by ToffA-DAS and ToffA-DAS+ is the manner how each optimization method deals with the feature selection. The ILP technique searches for a feasible solution from a single decision variable set. Conversely, GA strives for a feasible solution from a population of decision variable sets.

Normally, a GA searches for more adapted individuals, however, it can often be opted to select less adapted individuals in order to keep the diversity in the population. It explains why in some scenarios, ToffA-DAS+ suggested configurations with a lower number of positive contributions for satisfaction level of *soft goals*. When considering two `alternative` features with the same contribution value, for instance, the ILP algorithm in ToffA-DAS selects the first feature that is detected and suggests the configuration containing it. In contrast, the GA takes into account the feature, which enables ToffA-DAS+ to suggest the configuration that keeps the diversity of the population. Therefore, ToffA-DAS+ suggests configurations that provide a more balanced satisfaction level for *soft goals*. Additionally, it suggests a greater number of solutions benefiting the definition of adaptation models. Such adaptation models will be more diversified to meet different CCFs.

Although the configurations generated by the ToffA-DAS execution provide higher number of positive contribution of *soft goals* than those generated by the ToffA-DAS+ execution, the evidence gathered in the performed evaluation showed that the set of configurations generated by the second one is more diversified. In addition, it provides more feasible solutions to meet the requirements specification and needs of their stakeholders.

6.4.2 Threats to validity

In our study, we identified some threats to validity, which are described as follows:

Internal validity threats concern factors that can influence our observations. We have identified two internal validity threats. The first one is related to the instrumentation of the optimization algorithm. To mitigate this threat, we used the solver Gurobi[29] in ToffA-DAS release and the framework jMetal [136] in ToffA-DAS+ release. Both give support to find feasible configurations that meet all constraints defined in the optimization models. The second validity threat is regarding the metrics calculation that was conducted manually. Even this step having been made cautiously, some mistakes could have happened during this process. To address this threat, we performed a pair review of the data set resulting from the metric measurements.

External validity threats concern the generalization of our findings and points required for experiment replications. Our study considers only eight scenarios for Mobile Game and thirty-two scenarios for Smart Home. This number of scenarios generates small samples of data set resulting from the metrics calculation and can be seen as a threat to external validity. However, we employed in this study the same raw data and metrics presented by Guedes *et al.* [107] to compare both releases, ToffA-DAS and ToffA-DAS+. Such scenarios were based on all possible valid CCFs, besides the negative and positive contributions that influence the satisfaction level of a *soft goal*. Thus, the findings of the analysis can be used as a baseline for other studies dealing DAS configuration process.

In this sense, all the data used to run this study are available[1] for replication and further details.

Construct validity threats concern the relationship between theory and observation. We have identified two construct validity threats. The first one is the optimization method used. Unlike ToffA-DAS release that uses a solver based on the ILP technique, the ToffA-DAS+ release is based on a GA in combination with an SAT solver. For this reason, it was necessary to evaluate the constraints identified in the feature models in a different way. To mitigate this threat, we applied the pair review to assure that the optimization models defined in the experiment were correctly implemented aiming to generate the results in an equivalent manner. The second validity threat is regarding the interpretation of the data set resulting from the metric measurements for both releases. Thus, we used the raw data of ToffA-DAS and metrics used by authors [107]. Moreover, the exploratory study protocol was developed in detail and reviewed by researchers in order to mitigate the threat to the construct validity of the exploratory study.

Conclusion validity threats concern the relationship between treatment and outcome. Thus, the exploratory study design must sure that there was a statistical relationship between ToffA-DAS and ToffA-DAS+. For this reason, the results of the study were described using descriptive statistics, such as *median values* and *box plot* to deal with numerical processing and presentation of the data set. It is an adequate method to describe the analysis and interpretation of the data type collected. Regarding the hypotheses defined for the exploratory study, we used the non-parametric *Wilcoxon Signed-rank Test* [124] because the study employs two related samples and it yields difference scores that may be ranked in order of absolute size. Such a statistical test is suitable not only for large samples but also with small samples.

6.5 LESSONS LEARNED

Variability management is an important activity that describes different configurations of the system. This activity requires a consistent and scalable approach to explore, define, represent, implement, and evolve DAS. Based on simulations, we evidenced that our approach can be used for such purpose, *i.e.* it aims to explore reuse and support the specification of adaptation models for both dimensions *structural variability* and *context variability*.

We performed simulations with the *GridStix* DAS, as presented in Chapter 5. Such simulations aimed to verify the feasibility of using the ToffA-DAS approach and encourage the developers to use it in the configuration selection process of DAS. It also was possible to identify how to develop a generic optimization model, which can be used for different domains and system applications.

As a result of the simulations, we concluded that the ToffA-DAS approach is useful to perform trade-off analysis, generate feasible configurations, and identify possible adaptations that can occur at runtime. It meets the *structural variability* and *context variability*, besides the different measurements of *prioritization*, *contribution*, and *satisfaction levels* assigned to *goals*, *soft goals*, and *contexts*. However, the way of building the eCFM and

[1] https://sites.google.com/view/dspl-life-cycle/home

goal model may impact heavily on the result suggested by solver. Considering this factor, we provide the following points for consideration:

1. Inconsistency in CCFs may arise when the dependencies between features and **context features** are not represented in the correct way. The same is true with dependencies between *goals* and *hard goals*. In this scenario, the software engineer and stakeholders should check the models aiming to identify design faults. After that, they can agree on developing DAS applications according to the specialized feature model.

2. During the simulations, the optimization model identified some design faults. Initially, for example, we inserted `require` and `exclude` relationships between parent features and **context features**. At the same time, we inserted them between leaf features of each control system and **context features**. It resulted in faults, which we corrected to continue the simulations. Therefore, we recommend software engineers avoid modeling and development of DAS applications with such adaptation rules in order to prevent failures at runtime.

3. A problem associated to use of utility function as an optimization strategy is a difficulty to define such function that precisely represents the stakeholder's preferences. It aims to the heuristic representation of a desirable configuration. However, to find such a configuration, it is necessary to measure a utility value for each system's feature, which equals the weighted sum of all values assigned to the modeling elements. Then, the solver suggests an feasible configuration among possible configurations that maximize this utility value.

4. An adaptation to a given CCF corresponds to a products feasible configuration. Therefore, during application engineering, the DAS applications should be built according to variations in the requirement prioritization and artifacts defined in domain engineering. Next, it is necessary that CCFs be predetermined for all possible dynamic adaptations, in order to define different adaptation models. Thus, the software engineer can choose one of them to be eventually developed.

5. A GA starts with an initial population of individuals generated at random and its parameters include population size (*e.g.*, the number of possible valid configurations), crossover probability, mutation probability, and stopping condition. For this study, we applied the number of evaluations as a stop criterion for the algorithm. Such a value is equal to three times the population size, which was empirically established according to availability of the computer resources. When performing the simulations, we noticed that the results aforementioned could be better for ToffA-DAS+ whether the stopping condition was defined as being a greater number of iterations since the *fitness* measure of the better individual tends to continuously increase in each new generation. This would ensure that the NSGA-II algorithm searched for an optimal or near-optimal solution that meets a greater number of positive contributions to the *soft goals*.

6.6 CHAPTER SUMMARY

DAS is a software system capable of adapting at runtime based on changes in the surrounding environment. Aiming to design and develop this kind of system, the software engineer must handle both, system and context variability. This makes the development of DAS a challenging task. In this sense, software engineers have used the DSPL engineering processes and optimization algorithms to support their activities.

In this scenario, aiming to support the software engineer in the DAS design and development, we proposed the ToffA-DAS approach. It embraces *domain analysis*, *modeling*, *prioritization*, *contribution*, and *optimization*. In this chapter, we evolved our approach, now called ToffA-DAS+, to use the multi-objective algorithm NSGA-II and the SAT solver technology. With these improvements, our approach is capable of employing, separately, the objective functions for contexts, *goals*, and *soft goals* to evaluate the different solutions, besides providing feasible and valid configurations during the configuration selection process.

We conducted an exploratory study comparing both releases, ToffA-DAS, and ToffA-DAS+. Such a study was based on two DAS and simulations in accordance with different CCFs. As a result, we collected evidence that ToffA-DAS suggests configurations with a high number of positive contributions for *soft goals*, as well as a high number of negative contributions. Conversely, ToffA-DAS+ provides a more balanced result for the satisfaction level of *soft goals*, besides a greater number of solutions. It benefits the definition of adaptation models in which will be more diversified to meet different CCFs.

The next chapter presents the concluding remarks and future work.

PART IV

CONCLUSIONS

Chapter 7

The eye sees only what the mind is prepared to comprehend. — Robertson Davies, Tempest-Tost

CONCLUDING REMARKS AND FUTURE WORK

DAS is a software system capable of adapt at runtime based on changes in the surrounding environment. Aiming to design and develop this kind of system, the software engineer must satisfy the system's features, contexts, and NFRs. However, it is not a trivial task and must be made not only at runtime but also at design time to check the capacity of the system to meet self-adaptive operations. In this sense, software engineers have used the DSPL engineering processes and optimization methods to support such an activity.

In this scenario, aiming to support the software engineers in the design of DAS, we proposed the ToffA-DAS approach. It embraces *domain analysis*, *modeling*, *prioritization*, *contribution*, and *optimization*. Thus, ToffA-DAS is a comprehensive approach that supports the generation of the valid and feasible configurations that address the interactions between contextual information and NFRs. Next we present the contributions made by this thesis and directions to future work.

7.1 THESIS CONTRIBUTIONS

In this thesis, we are pursuing a threefold goal in DSPL engineering field by proposing an approach that *(i)* manages both dimensions, *structural variability* and *context variability*; *(ii)* facilitates the understanding of how DAS applications can behave from a certain context change, and *(iii)* enables to conduct trade-off analysis in order to find the valid and feasible configurations that meet the constraints and the interactions between contextual information and NFRs. To fulfill our goals, we made the following contributions:

1. *eCFM Technique (Chapter 3).* We extended the CFM technique aiming to model the context constraints since in the real environment there are contexts that cannot occur at the same time. We performed a survey to compare both techniques CFM and eCFM, from the viewpoint of *expressiveness* to model the context constraints and *easiness of use*. Indeed, the analysis was focused on the comprehensibility of contextual variability modeling. As a result, the eCFM was considered a technique with a greater expressiveness power to represent adaptation rules between contexts

and system features, besides the easiness of use and organization with the grouping of contexts. Therefore, we argue that the software engineers may take into account the use of eCFM technique to model DAS. Based on it, we planned the second part of this work.

2. *ToffA-DAS Approach (Chapter 4).* Once defined how to represent both dimensions *structural variability* and *context variability*, it was proposed an approach called ToffA-DAS to support the configuration selection process in DAS projects. Such an approach deals with the configuration selection process embracing the interactions between contextual information and NFRs. In addition, it uses *Utility-based* planning as a strategy to express the priorities of users over services provided by a DAS application. Those priorities are represented as weights aiming to direct the choice of feasible solutions.

3. *Feasibility of using the ToffA-DAS approach (Chapter 5).* We performed two studies in order to evaluate our proposal. Firstly, we performed a study based on simulations using the *GridStix* DAS to gather initial evidence about the feasibility of using the ToffA-DAS approach from the point of view of *(i)* conduction of trade-off analysis by considering changes in the prioritization of elements that comprise eCFM and goal models such as *goals*, *soft goals*, and contexts, and *(ii)* definition of adaptation models, from feasible configurations found in the CCF change analysis. All simulations presented consistent results and in accordance with the real-world scenarios and satisfied the estimated utility values and linear constraints. They met the variability dimensions and different satisfaction levels assigned to *soft goals*, besides measurements of prioritization assigned to *contexts*, *goals*, and *soft goals*. Secondly, we conducted an exploratory study when ToffA-DAS and ConD4DaS approaches were compared with each other. It was focused on evaluating how the configurations obtained from both approaches affect the overall satisfaction level of *soft goals*. By observing the resulting data from evaluation, we concluded that the configurations generated by the ToffA-DAS execution provide higher satisfaction levels of *soft goals* than those generated by the ConD4DaS execution. ToffA-DAS promotes a more comprehensive approach to the DAS configuration selection process, since it embraces a variability modeling technique (eCFM), prioritization of *soft goals*, *goals*, and *contexts*, besides the calculation of the contribution values for those elements.

4. *ToffA-DAS+ Approach (Chapter 6).* We evolved our approach, now called ToffA-DAS+, to use the GA NSGA-II and an SAT solver. Next, we conducted an exploratory study comparing both releases, ToffA-DAS, and ToffA-DAS+. Such a study was performed using two DAS applications and several simulations in accordance with different CCFs. As a result, we collected evidence that ToffA-DAS suggests configurations with a greater number of positive contributions, as well as, a greater number of negative contributions for *soft goals*. Conversely, ToffA-DAS+ provides a more balanced result for the satisfaction level of *soft goals*, besides a

greater number of solutions. It benefits the definition of adaptation model since it will be more diversified to meet different CCFs.

7.2 LIMITATIONS AND DIRECTIONS FOR FUTURE WORK

In this section, we provide an extra discussion on the directions for future work for the ToffA-DAS+ approach. In addition, we also present potential gaps identified after carrying out the experimental studies.

- **Multi-objective algorithms.** Firstly, we plan to compare different multi-objective algorithms with diverse configurations to identify the best option to be used with our approach. Although the assessment results have shown that using NSGA-II is quite efficient for identifying feasible configurations, studies existing [2, 139] in the literature reported evaluations when algorithms such as IBEA and PAES were compared with NSGA-II in the DSPL engineering field and demonstrated good results.

- **Tool support.** In our current approach only STEP 4 (optimization) is automated. However, it is not an easy task to measure the satisfaction level, prioritization, and contribution to a huge number of modeling elements such as features, contexts, and NFRs. For systems that present many modeling elements and to overcome scalability issues, as future work we intend to develop a tool that encompasses all steps of the ToffA-DAS+ approach.

- **Robot domain.** Additionally, we plan to investigate the advantages and drawbacks of employing the ToffA-DAS+ approach across real DAS applications, for example, in the industrial robotics. Robots have a variety of forms, purposes, and functions that can operate to accomplish several missions. In this sense, it must be composed of a specific combination of functionalities that strongly depend on several elements such as robot mechanical structure, tasks to be performed, and environmental conditions. These elements give rise to a multitude of product configurations that require suitable variability mechanisms and management methodologies [140].

- **Trade-off analysis at runtime.** Our approach aims to identify at design time a set of possible adaptations and information that can affect the product configuration. However, the system quality evaluation must also be made at runtime to check the capacity of the system to meet self-adaptive operations [15]. For future work, we are interested in investigating the feasibility of using the approach for trade-off analysis during the execution of the DAS application. It aims to conduct adaptations at runtime by meeting the desirable variants and default configurations defined in the adaptation model.

- **Technical aspects.** The definition of the degree to which the variable features satisfy the *soft goals* is often subjective and makes the configuration selection process more difficult. It depends on the software engineer knowledge in the application domain and may not be feasible to satisfy certain change requests at runtime [17]. Further efforts should be made, in regarding the employ the ToffA-DAS+ approach at the running application to establish confidence regarding assign suitable satisfaction levels for NFRs. Therefore, more studies are needed in distinct domains considering, especially, the point of view of the DAS developers.

- **Uncertainty.** Although we employ the trade-off analysis to find a set of valid and feasible configurations that meet different scenarios, the DAS applications are subjected to the uncertainty that was not explicitly designed. Such uncertainty is caused by parameters whose values change at the running applications and can make NFRs unsatisfied. The parameters are given information related to sensor failure or occlusion which can occur at runtime [141, 142]. Importantly, we may propose to evolve the ToffA-DAS+ approach aiming to address uncertainty and define a strategy to handle the possible adaptations that were not foreseen at design time.

BIBLIOGRAPHY

1 NORTHROP, L. M. Sei's software product line tenets. *IEEE*, v. 19, n. 4, p. 32–40, 2002.

2 PASCUAL, G. G. et al. Applying multiobjective evolutionary algorithms to dynamic software product lines for reconfiguring mobile applications. *Journal of Systems and Software*, Elsevier, 2015.

3 BASHARI, M.; BAGHERI, E.; DU, W. Dynamic software product line engineering: a reference framework. *International Journal of Software Engineering and Knowledge Engineering*, World Scientific, v. 27, n. 02, p. 191–234, 2017.

4 CAPILLA, R. et al. An overview of dynamic software product line architectures and techniques: Observations from research and industry. *Journal of Systems and Software*, Elsevier, 2014.

5 EIBEN, A. E.; SMITH, J. E. *Introduction to evolutionary computing*. [S.l.]: Springer, 2015.

6 HINCHEY, P. S. M.; SCHMID, K. Building dynamic software product lines. *IEEE Computer*, 2012.

7 BENCOMO, N. et al. Dynamically adaptive systems are product lines too: Using model-driven techniques to capture dynamic variability of adaptive systems. In: *Second International Workshop DSPL*. [S.l.: s.n.], 2008. p. 23–32.

8 BENCOMO, N.; HALLSTEINSEN, S. O.; ALMEIDA, E. S. de. A view of the dynamic software product line landscape. *IEEE Computer*, 2012.

9 HALLSTEINSEN, S. et al. Using product line techniques to build adaptive systems. In: IEEE. *10th International Software Product Line Conference*. [S.l.], 2006.

10 BENCOMO, N. et al. Reflective component-based technologies to support dynamic variability. In: CITESEER. *VaMoS*. [S.l.], 2008. p. 141–150.

11 HARTMANN, H.; TREW, T. Using feature diagrams with context variability to model multiple product lines for software supply chains. In: IEEE. *12th International Software Product Line Conference (SPLC)*. [S.l.], 2008.

12 MENS, K. et al. A taxonomy of context-aware software variability approaches. In: *Companion Proceedings of the 15th International Conference on Modularity*. [S.l.: s.n.], 2016. p. 119–124.

13 CHENG, B. H. et al. Software engineering for self-adaptive systems. In: CHENG, B. H. et al. (Ed.). Berlin, Heidelberg: Springer-Verlag, 2009. cap. Software Engineering for Self-Adaptive Systems: A Research Roadmap, p. 1–26.

14 SOARES, L. R. et al. Analysis of non-functional properties in software product lines: a systematic review. In: IEEE. *Software Engineering and Advanced Applications (SEAA), 40th EUROMICRO Conference on*. [S.l.], 2014.

15 SOUSA, A. O. de et al. Quality evaluation of self-adaptive systems: Challenges and opportunities. In: *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*. [S.l.: s.n.], 2019. p. 213–218.

16 LAMSWEERDE, A. V. *Requirements engineering: From system goals to UML models to software*. [S.l.]: Chichester, UK: John Wiley & Sons, 2009.

17 YRJÖNEN, A.; MERILINNA, J. Extending the nfr framework with measurable non-functional requirements. In: *NFPinDSML@ MoDELS*. [S.l.: s.n.], 2009.

18 ŞORA, I. et al. Managing variability of self-customizable systems through composable components. *Software Process: Improvement and Practice*, Wiley Online Library, v. 10, n. 1, p. 77–95, 2005.

19 SIEGMUND, N. et al. Scalable prediction of non-functional properties in software product lines: Footprint and memory consumption. *Information and Software Technology*, Elsevier, 2013.

20 BENAVIDES, D.; SEGURA, S.; RUIZ-CORTéS, A. Automated analysis of feature models 20 years later: A literature review. *Information Systems*, v. 35, n. 6, p. 615 – 636, 2010.

21 HALLSTEINSEN, S. et al. Dynamic software product lines. *IEEE Computer*, IEEE Computer Society Press, 2008.

22 SOUZA, M. L. a. de J.; SANTOS, A. R.; ALMEIDA, E. S. de. Towards the selection of modeling techniques for dynamic software product lines. In: *Proceedings of the Fifth International Workshop on Product LinE Approaches in Software Engineering*. [S.l.: s.n.], 2015. (PLEASE).

23 SOUZA, M. L. de J. et al. Evaluating variability modeling techniques for dynamic software product lines: A controlled experiment. In: IEEE. *Software Components, Architectures and Reuse (SBCARS), 2016 X Brazilian Symposium on*. [S.l.], 2016.

24 REISNER, E. et al. Using symbolic evaluation to understand behavior in configurable software systems. In: IEEE. *2010 ACM/IEEE 32nd International Conference on Software Engineering*. [S.l.], 2010. v. 1, p. 445–454.

25 GUEDES, G. et al. Variability management in dynamic software product lines: A systematic mapping. In: IEEE. *Components, Architectures and Reuse Software (SBCARS), 2015 IX Brazilian Symposium on*. [S.l.], 2015.

26 CARVALHO, M. L. L. et al. On the implementation of dynamic software product lines: An exploratory study. *Journal of Systems and Software*, v. 136, p. 74–100, 2018.

27 SANTOS, I. S. et al. Dynamically adaptable software is all about modeling context variability and avoiding failures. *IEEE SOFTWARE*, 2017.

28 SALLER, K.; LOCHAU, M.; REIMUND, I. Context-aware DSPLs: Model-based runtime adaptation for resource-constrained systems. In: *17th International Software Product Line Conference Co-located Workshops*. [S.l.]: ACM, 2013. p. 106–113.

29 OPTIMIZATION, I. G. *Gurobi Optimizer Reference Manual*. 2016. Disponível em: <http://www.gurobi.com>.

30 CARVALHO, M. L. L. et al. On the implementation of dynamic software product lines: A preliminary study. In: *Proceedings of the X Brazilian Symposium on Components, Architectures and Reuse Software (SBCARS)*. [S.l.]: IEEE, 2016.

31 HUEBSCHER, M. C.; MCCANN, J. A. A survey of autonomic computing - degrees, models, and applications. *ACM Comput. Surv.*, v. 40, p. 7:1–7:28, 2008.

32 POHL, K.; BöCKLE, G.; LINDEN, F. J. v. d. Software product line engineering: Foundations, principles and techniques. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

33 LINDEN, F. J. v. d.; SCHMID, K.; ROMMES, E. *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007. ISBN 3540714367.

34 CLASSEN, A.; HEYMANS, P.; SCHOBBENS, P.-Y. What's in a feature: A requirements engineering perspective. In: *Proceedings of the 11th International in Fundamental Approaches to Software Engineering Conference, (FASE)*. [S.l.]: Springer, 2008. (Lecture Notes in Computer Science, v. 4961), p. 16–30.

35 DIJKSTRA, E. W. Structured programming. In: DAHL, O. J.; DIJKSTRA, E. W.; HOARE, C. A. R. (Ed.). [S.l.]: Academic Press Ltd., 1972. cap. Chapter I: Notes on Structured Programming, p. 1–82.

36 PARNAS, D. L. On the design and development of program families. 1976.

37 KANG, K. C. et al. Feature-oriented domain analysis (foda) feasibility study. November 1990.

38 R.WOLFINGER S.REITER, D. P. H.; DOPPLER, C. Suporting runtime system adaptation through product line engineering and plug-in techniques1. In: *Proceedings of the 7th International Conference on Composition-Based Software Systems (ICCBSS)*. [S.l.: s.n.], 2008. p. 21–30.

39 APEL, S. et al. *Feature-Oriented Software Product Lines; Concepts and Implementation*. [S.l.]: Springer, 2013.

40 MUTHIG, D.; PATZKE, T. Generic implementation of product line components. In: *Revised Papers from the International Conference NetObjectDays on Objects, Components, Architectures, Services, and Applications for a Networked World (NODe)*. [S.l.]: Springer-Verlag, 2003. p. 313–329.

41 GURP, J. V.; BOSCH, J.; SVAHNBERG, M. On the notion of variability in software product lines. In: *Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA)*. [S.l.]: IEEE Computer Society, 2001. p. 45–.

42 MCKINLEY, P. K. et al. Composing adaptive software. *Computer*, IEEE Computer Society Press, v. 37, n. 7, p. 56–64, jul. 2004.

43 BOSCH, J.; CAPILLA, R. Dynamic variability in software-intensive embedded system families. *IEEE Computer*, v. 45, n. 10, p. 28–35, 2012.

44 BARESI, L.; GUINEA, S.; PASQUALE, L. Service-oriented dynamic software product lines. *IEEE Computer*, v. 45, p. 42–48, 2012.

45 LEE, J.; KOTONYA, G.; ROBINSON, D. Engineering service-based dynamic software product lines. *Computer*, IEEE Computer Society, v. 45, p. 49–55, 2012.

46 CETINA, C. et al. Using feature models for developing self-configuring smart homes. In: *Fifth International Conference on Autonomic and Autonomous Systems, ICAS 2009, Valencia, Spain, 20-25 April 2009*. [S.l.: s.n.], 2009. p. 179–188.

47 BENCOMO, N. et al. Reflective component-based technologies to support dynamic variability. In: *Proceedings of the 2nd Workshop Variability Modeling of Software-Intensive Systems (VaMoS)*. [S.l.: s.n.], 2008.

48 ISTOAN, P. et al. Dynamic software product lines for service-based systems. In: *Proceedings of the 9th IEEE International Conference on Computer and Information Technology (CIT)*. [S.l.]: IEEE Computer Society, 2009.

49 BENCOMO, N. et al. Dynamically adaptive systems are product lines too: Using model-driven techniques to capture dynamic variability of adaptive systems. In: *Proceedings 2nd International Workshop Dynamic Software Product Lines (DSPL)*. [S.l.: s.n.], 2008. p. 23–32.

50 LEE, J.; WHITTLE, J.; STORZ, O. Bio-inspired mechanisms for coordinating multiple instances of a service feature in dynamic software product lines. In: . [S.l.: s.n.], 2011. p. 670–683.

51 MUCCINI, H.; HOEK, A. V. D. Towards testing product line architectures. *Electronic Notes in Theoretical Computer Science*, Elsevier Science, v. 82, n. 6, p. 1–11, 2003.

52 CETINA, C. et al. Designing and prototyping dynamic software product lines: Techniques and guidelines. In: *Proceedings of the 14th International Conference on Software Product Lines (SPLC)*. [S.l.]: Springer-Verlag, 2010.

53 BURÉGIO, V. A.; MEIRA, S. R. de L.; ALMEIDA, E. S. de. Characterizing dynamic software product lines - A preliminary mapping study. In: *Proceedings of the 14th International Conference in Software Product Lines (SPLC)*. [S.l.: s.n.], 2010. p. 53–60.

54 DINKELAKER RALF MITSCHKE, K. F. T.; MEZINI, M. A dynamic software product line approach using aspect models at runtime. In: *Proceedings of the 1st Workshop on Composition and Variability 10 Rennes, France colocated with AOSD*. [S.l.: s.n.], 2010.

55 LEE, J.; KANG, K. C. A feature-oriented approach to developing dynamically reconfigurable products in product line engineering. In: *Proceedings of the 10th International Conference (SPLC)*. [S.l.: s.n.], 2006. p. 131–140.

56 CETINA, C. et al. An architectural discussion on dspl. In: IRISH SOFTWARE ENGINEERING RESEARCH CENTRE (LERO). *2nd SPLC Workshop on Dynamic Software Product Line (DSPL)*. [S.l.]: Irish Software Engineering Research Centre (Lero), 2008. p. 59–68.

57 PEÑA, J. et al. Designing and managing evolving systems using a MAS product line approach. *Science Computer Programe*, v. 66, p. 71–86, 2007.

58 LEE, J. Dynamic feature deployment and composition for dynamic software product lines. In: *Proceedings of the 17th International Software Product Line Conference (SPLC)*. [S.l.]: ACM, 2013. p. 114–116.

59 MCGREGOR, J. *The Evolution of Product Line Assets*. [S.l.], 2003.

60 SVAHNBERG, M.; BOSCH, J. Evolution in software product lines. In: *Proceedings, 3rd International Workshop on Software Architectures for Products Families (IWSAPF-3)*. [S.l.]: Springer LNCS, 2000. p. 391–422.

61 BERG, K.; BISHOP, J.; MUTHIG, D. Tracing software product line variability: From problem to solution space. In: *Proceedings of the 2005 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries*. Republic of South Africa: South African Institute for Computer Scientists and Information Technologists, 2005. (SAICSIT '05), p. 182–191.

62 QUINTON, C. et al. Evolution in dynamic software product lines: Challenges and perspectives. In: *Proceedings 19th International Software Product Line Conference (SPL)*. [S.l.]: ACM, 2015. p. 126–130.

63 SEIDL, C.; HEIDENREICH, F.; ASSMANN, U. Co-evolution of models and feature mapping in software product lines. In: ACM. *Proceedings of the 16th International Software Product Line Conference-Volume 1*. [S.l.], 2012. p. 76–85.

64 HELLEBOOGH, A. et al. Adding variants on-the-fly: Modeling meta-variability in dynamic software product lines. In: *Proceedings of the 3rd International Workshop on Dynamic Software Product Lines (DSPL) at the 13th International Software Product Line Conference (SPLC)*. [S.l.: s.n.], 2009. p. 18–27.

65 HEIDER, W.; RABISER, R.; GRÜNBACHER, P. Facilitating the evolution of products in product line engineering by capturing and replaying configuration decisions. *International Journal on Software Tools for Technology Transfer*, Springer, v. 14, n. 5, p. 613–630, 2012.

66 TALIB, M. A. et al. Requirements for evolvable dynamic software product lines. In: *Software Product Lines - 14th International Conference, SPLC 2010, Jeju Island, South Korea, September 13-17, 2010. Workshop Proceedings (Volume 2 : Workshops, Industrial Track, Doctoral Symposium, Demonstrations and Tools)*. [S.l.: s.n.], 2010. p. 43–46.

67 MAIRIZA, D.; ZOWGHI, D.; NURMULIANI, N. An investigation into the notion of non-functional requirements. In: ACM. *Proceedings of the 2010 ACM Symposium on Applied Computing*. [S.l.], 2010.

68 DARDENNE, A.; LAMSWEERDE, A. van; FICKAS, S. Goal-directed requirements acquisition. In: *Selected Papers of the Sixth International Workshop on Software Specification and Design*. [S.l.]: Elsevier Science Publishers B. V., 1993. p. 3–50.

69 MENDONçA, D. F. et al. Goda: A goal-oriented requirements engineering framework for runtime dependability analysis. *Information and Software Technology*, v. 80, p. 245 – 264, 2016.

70 HORKOFF, J. et al. Using goal models downstream: a systematic roadmap and literature review. *International Journal of Information System Modeling and Design (IJISMD)*, IGI Global, v. 6, n. 2, p. 1–42, 2015.

71 ABRAHÃO, S. et al. Comparing the effectiveness of goal-oriented languages: results from a controlled experiment. In: *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. [S.l.: s.n.], 2018. p. 1–4.

72 ALI, R.; CHITCHYAN, R.; GIORGINI, P. Context for goal-level product line derivation. Bournemouth University, Fern Barrow, Poole, Dorset, BH12 5BB, UK, 2009.

73 YU, E.; MYLOPOULOS, J. Why goal-oriented requirements engineering. In: *Proceedings of the 4th International Workshop on Requirements Engineering: Foundations of Software Quality*. [S.l.: s.n.], 1998.

74 GOLDSBY, H. J. et al. Goal-based modeling of dynamically adaptive system requirements. In: IEEE. *15Th annual IEEE international conference and workshop on the engineering of computer based systems (ecbs 2008)*. [S.l.], 2008. p. 36–45.

75 KANG, K. C. et al. *Feature-oriented domain analysis (FODA) feasibility study*. [S.l.], 1990.

76 UCHÔA, A. G. et al. Reminder: an approach to modeling non-functional properties in dynamic software product lines. In: SPRINGER. *International Conference on Software Reuse*. [S.l.], 2017. p. 65–73.

77 SANCHEZ, L. E. et al. An approach based on feature models and quality criteria for adapting component-based systems. *Journal of Software Engineering Research and Development*, SpringerOpen, v. 3, n. 1, p. 1–30, 2015.

78 MORIN, B. et al. Models@ run.time to support dynamic adaptation. *Computer*, IEEE Computer Society Press, v. 42, n. 10, p. 44–51, out. 2009.

79 NEŠKOVIĆ, S.; MATIĆ, R. Context modeling based on feature models expressed as views on ontologies via mappings. *Computer Science and Information Systems*, v. 12, n. 3, p. 961–977, 2015.

80 PARRA, C. et al. Using constraint-based optimization and variability to support continuous self-adaptation. In: ACM. *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. [S.l.], 2012.

81 SAWYER, P. et al. Using constraint programming to manage configurations in self-adaptive systems. *Computer*, IEEE, 2012.

82 DOUNAS, L. et al. Runtime requirements monitoring framework for adaptive e-learning systems. In: *International Conference on Software & Systems Engineering and their Applications (ICSSEA'15)*. [S.l.: s.n.], 2015.

83 ROUVOY, R. et al. Music: Middleware support for self-adaptation in ubiquitous and service-oriented environments. In: *Software engineering for self-adaptive systems*. [S.l.]: Springer, 2009. p. 164–182.

84 GEIHS, K. et al. A comprehensive solution for application-level adaptation. *Software: Practice and Experience*, Wiley Online Library, v. 39, n. 4, p. 385–422, 2009.

85 LUENBERGER, D. G.; YE, Y. et al. *Linear and nonlinear programming*. [S.l.]: Springer, 1984.

86 EIBEN, A. E.; SMITH, J. E. et al. *Introduction to evolutionary computing*. [S.l.]: Springer, 2003.

87 AZAMATHULLA, H. M. et al. Comparison between genetic algorithm and linear programming approach for real time operation. *Journal of Hydro-environment Research*, Elsevier, v. 2, n. 3, p. 172–181, 2008.

88 FAJAR, M. et al. An integer programming based decision making framework for application engineering in software product line development. In: *Proc. Korea-Japan Joint Workshop on JCT*. [S.l.: s.n.], 2012.

89 CORMEN, T. H. et al. *Introduction to algorithms*. [S.l.]: MIT press, 2009.

90 WOLSEY, L. A.; NEMHAUSER, G. L. *Integer and combinatorial optimization*. [S.l.]: John Wiley & Sons, 1999.

91 GOLDBERG, D. E. Genetic algorithms in search optimization and machine learning. In: . [S.l.: s.n.], 1989.

92 MICHALEWICZ, Z. *Genetic algorithms+ data structures= evolution programs*. [S.l.]: Springer Science & Business Media, 2013.

93 GEN, M.; CHENG, R. Genetic algorithms and engineering optimization. In: . [S.l.: s.n.], 1997.

94 GOLDBERG, D. E.; HOLLAND, J. H. Genetic algorithms and machine learning. Kluwer Academic Publishers-Plenum Publishers; Kluwer Academic Publishers , 1988.

95 KACPRZYK, J.; PEDRYCZ, W. *Springer handbook of computational intelligence*. [S.l.]: Springer, 2015.

96 COSTA, L.; OLIVEIRA, P. Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems. *Computers & Chemical Engineering*, Elsevier, v. 25, n. 2-3, p. 257–266, 2001.

97 WARDLAW, R.; BHAKTIKUL, K. Comparison of genetic algorithm and linear programming approaches for lateral canal scheduling. *Journal of irrigation and drainage engineering*, American Society of Civil Engineers, v. 130, n. 4, p. 311–317, 2004.

98 ACHER, M. et al. Modeling context and dynamic adaptations with feature models. In: *4th International Workshop Models@ run. time at Models 2009 (MRT'09)*. [S.l.: s.n.], 2009.

99 SALLER, K.; LOCHAU, M.; REIMUND, I. Context-aware dspls: Model-based runtime adaptation for resource-constrained systems. In: *Proceedings of the 17th International Software Product Line Conference Co-located Workshops*. [S.l.: s.n.], 2013. (SPLC).

100 HALLSTEINSEN, S. et al. Dynamic software product lines. *Computer*, IEEE, v. 41, n. 4, 2008.

101 HADAR, I. et al. Comparing the comprehensibility of requirements models expressed in Use Case and Tropos: Results from a family of experiments. *Information and Software Technology*, v. 55, n. 10, p. 1823–1843, 2013.

102 FRANCO, J. M. et al. Improving self-adaptation planning through software architecture-based stochastic modeling. *Journal of Systems and Software*, Elsevier, v. 115, p. 42–60, 2016.

103 EDWARDS, R.; BENCOMO, N. Desire: further understanding nuances of degrees of satisfaction of non-functional requirements trade-off. In: *Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems*. [S.l.: s.n.], 2018. p. 12–18.

104 PAUCAR, L. H. G.; BENCOMO, N. The reassessment of preferences of non-functional requirements for better informed decision-making in self-adaptation. In: IEEE. *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*. [S.l.], 2016. p. 32–38.

105 ESFAHANI, N.; ELKHODARY, A.; MALEK, S. A learning-based framework for engineering feature-oriented self-adaptive software systems. *IEEE transactions on software engineering*, IEEE, 2013.

106 GREENWOOD, P. et al. Modelling service requirements variability: The diva way. *Service Engineering: European Research Results*, Springer, 2011.

107 GUEDES, G.; SILVA, C.; SOARES, M. Comparing configuration approaches for dynamic software product lines. In: *Proceedings of the 31st Brazilian Symposium on Software Engineering (SBES)*. [S.l.]: ACM, 2017.

108 NASCIMENTO, A. S.; RUBIRA, C. M.; CASTOR, F. Arcmape: A software product line infrastructure to support fault-tolerant composite services. In: IEEE. *High-Assurance Systems Engineering (HASE), 2014 IEEE 15th International Symposium on*. [S.l.], 2014.

109 SÁNCHEZ, L. E. et al. An approach for managing quality attributes at runtime using feature models. In: IEEE. *Software Components, Architectures and Reuse (SBCARS), Eighth Brazilian Symposium on*. [S.l.], 2014.

110 ALI, R.; DALPIAZ, F.; GIORGINI, P. Reasoning with contextual requirements: Detecting inconsistency and conflicts. *Information and Software Technology*, Elsevier, 2013.

111 GAMEZ, N.; FUENTES, L.; TROYA, J. M. Creating self-adapting mobile systems with dynamic software product lines. *IEEE Software*, IEEE, 2015.

112 WELSH, K. et al. Self-explanation in adaptive systems based on runtime goal-based models. In: *Transactions on Computational Collective Intelligence XVI*. [S.l.]: Springer, 2014. p. 122–145.

113 HUGHES, D. et al. An intelligent and adaptable flood monitoring and warning system. In: NESC. *UK E-Science all hands meeting*. [S.l.], 2006. p. 53–+.

114 UCHÔA, A. G. et al. Dymmer-nfp: Modeling non-functional properties and multiple context adaptation scenarios in software product lines. In: *16th International Conference on Software Reuse, ICSR, Proceedings*. [S.l.: s.n.], 2017. p. 175–183.

115 HEIDELBERG, U. *TSP LIB*. Disponível em: <https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>.

116 SAATY, T. L. How to make a decision: the analytic hierarchy process. *European journal of operational research*, North-Holland, v. 48, n. 1, p. 9–26, 1990.

117 KARLSSON, J.; WOHLIN, C.; REGNELL, B. An evaluation of methods for prioritizing software requirements. *Information and software technology*, Elsevier, v. 39, n. 14-15, p. 939–947, 1998.

118 RACHEVA, Z.; DANEVA, M.; BUGLIONE, L. Supporting the dynamic reprioritization of requirements in agile development of software products. In: IEEE. *Software Product Management, 2008. IWSPM'08. Second International Workshop on.* [S.l.], 2008.

119 WOHLIN, C. et al. *Experimentation in Software Engineering: An Introduction.* [S.l.]: Kluwer Academic Publishers, 2000.

120 GHERARDI, L.; HOCHGESCHWENDER, N. Rra: Models and tools for robotics run-time adaptation. In: IEEE. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* [S.l.], 2015. p. 1777–1784.

121 PIMENTEL, J. et al. Deriving software architectural models from requirements models for adaptive systems: the stream-a approach. *Requirements Engineering*, Springer, v. 17, n. 4, p. 259–281, 2012.

122 WOHLIN, C. et al. *Experimentation in software engineering.* [S.l.]: Springer Science & Business Media, 2012.

123 JURISTO, N.; MORENO, A. M. *Basics of software engineering experimentation.* [S.l.]: Springer Science & Business Media, 2013.

124 SIDNEY, S. Nonparametric statistics for the behavioral sciences. *The Journal of Nervous and Mental Disease*, LWW, v. 125, n. 3, p. 497, 1957.

125 PAREJO, J. A. et al. Multi-objective test case prioritization in highly configurable systems: A case study. *Journal of Systems and Software*, Elsevier, v. 122, p. 287–310, 2016.

126 PAL, A.; CHARKHGARD, H. Fpbh: A feasibility pump based heuristic for multi-objective mixed integer linear programming. *Computers & Operations Research*, Elsevier, v. 112, p. 104760, 2019.

127 HIERONS, R. M. et al. Sip: Optimal product selection from feature models using many-objective evolutionary optimization. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, ACM New York, NY, USA, v. 25, n. 2, p. 1–39, 2016.

128 WHITE, J. et al. Automated diagnosis of feature model configurations. *Journal of Systems and Software*, Elsevier, v. 83, n. 7, p. 1094–1107, 2010.

129 LUO, Y.-C.; GUIGNARD, M.; CHEN, C.-H. A hybrid approach for integer programming combining genetic algorithms, linear programming and ordinal optimization. *Journal of Intelligent Manufacturing*, Springer, v. 12, n. 5-6, p. 509–519, 2001.

130 MENDONCA MOISES BRANCO, D. C. M. *S.P.L.O.T. - Software Product Lines Tools.* 2009. Disponível em: <http://52.32.1.180:8080/SPLOT/index.html>.

131 WARNERS, J. P. A linear-time transformation of linear inequalities into conjunctive normal form. *Information Processing Letters*, Elsevier, v. 68, n. 2, p. 63–69, 1998.

132 SCHULER, R. An algorithm for the satisfiability problem of formulas in conjunctive normal form. *Journal of Algorithms*, Elsevier, v. 54, n. 1, p. 40–44, 2005.

133 JOHANSEN, M. F.; HAUGEN, Ø.; FLEUREY, F. Properties of realistic feature models make combinatorial testing of product lines feasible. In: SPRINGER. *International Conference on Model Driven Engineering Languages and Systems.* [S.l.], 2011. p. 638–652.

134 BERRE, D. L.; PARRAIN, A. The sat4j library, release 2.2. *Journal on Satisfiability, Boolean Modeling and Computation*, IOS Press, v. 7, n. 2-3, p. 59–64, 2010.

135 DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, IEEE, v. 6, n. 2, p. 182–197, 2002.

136 DURILLO, J. J.; NEBRO, A. J. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, Elsevier, v. 42, n. 10, p. 760–771, 2011.

137 SAYYAD, A. S. et al. Optimum feature selection in software product lines: Let your model and values guide your search. In: IEEE. *2013 1st International Workshop on Combining Modelling and Search-Based Software Engineering (CMSBSE).* [S.l.], 2013. p. 22–27.

138 PASCUAL, G. G.; PINTO, M.; FUENTES, L. Run-time adaptation of mobile applications using genetic algorithms. In: IEEE. *2013 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS).* [S.l.], 2013. p. 73–82.

139 SANTOS, I. de S. et al. Optimized feature selection for initial launch in dynamic software product lines. In: *ICEIS (2).* [S.l.: s.n.], 2018. p. 145–156.

140 GARCÍA, S. et al. Variability modeling of service robots: Experiences and challenges. In: *Proceedings of the 13th International Workshop on Variability Modelling of Software-Intensive Systems.* [S.l.: s.n.], 2019. p. 1–6.

141 FREDERICKS, E. M.; DEVRIES, B.; CHENG, B. H. Towards run-time adaptation of test cases for self-adaptive systems in the face of uncertainty. In: *Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems.* [S.l.: s.n.], 2014. p. 17–26.

142 BOWERS, K. M. et al. Providentia: Using search-based heuristics to optimize satisficement and competing concerns between functional and non-functional objectives in self-adaptive systems. *Journal of Systems and Software*, Elsevier, v. 162, p. 110497, 2020.

PART V

APPENDICES

Appendix

A

ARTIFACTS OF THE SURVEY

A.1 BACKGROUND FORM

This section presents the background form used to characterize the subjects according to their experience and expertise.

Modeling Dynamic Software Product Lines - Background Form

Modeling DSPL Controlled Experiment Background Form

*Obrigatório

General Information

1. **Full Name** *

2. **e-mail**

3. **Company and/or University**

4. **Academic Degree**
   Undergraduated, Master, Ph.D. , Post-Doc....

5. **Position in the Company**
   Tester, Developer, software engineer, other options...

6. **Years of experience in the market**
   Mark only one
   Marcar apenas uma oval.

   - < 1 year
   - >= 1 year and =< 5years
   - >= 5 years and < 10 years
   - > 10 years

Technical Knowledge

Select the option that best fits to your profile

7. **Do you ever work with Software Product Lines (SPL) approach?**
   Choose only one.
   Marcar apenas uma oval.

   - In Academy
   - In Industry
   - In Academy and Industry
   - I Know about SPL but I never work with it
   - I never heard about SPL

8. **How many years of experience do you have in SPL?**
   Choose only one
   Marcar apenas uma oval.

   - < 1 year
   - >= 1 year and < 3 years
   - >= 3 years and < 6 years
   - >= 6 years

9. **Do you ever work with Context Aware Applications (CAP)?**
   Choose only one.
   Marcar apenas uma oval.

   - In Academy
   - In Industry
   - In Academy and Industry
   - I Know about CAP but I never work with it
   - I never heard about CAP

10. **How many years of experience do you have with Context Aware Applications?**
    Choose only one
    Marcar apenas uma oval.

    - < 1 year
    - >= 1 year and < 3 years
    - >= 3 years and < 6 years
    - >= 6 years

11. **Do you ever work with Dynamic Software Product Line (DSPL) approach?**
    Choose only one.
    Marcar apenas uma oval.

    - In Academy
    - In Industry
    - In Academy and Industry
    - I know about the DSPL but I never work with it
    - I never heard about DSPL

12. **How many years of experience do you have in DSPL?**
Choose only one
Marcar apenas uma oval.

- < 1 year
- >= 1 year and < 3 years
- >= 3 years and < 6 years
- >= 6 years

13. **Do you ever work with Software Modeling (SM) (UML, Feature Model,etc)?**
Choose only one.
Marcar apenas uma oval.

- In Academy
- In Industry
- In Academy and Industry
- I know about SM but I never work with it
- I never heard about Software Modeling

14. **How many years of experience do you have in Software Modeling (SM)**
Choose only one
Marcar apenas uma oval.

- < 1 year
- >= 1 year and < 3 years
- >= 3 years and < 6 years
- >= 6 years

A.2 GUIDELINES

This section presents the guideline describing instructions on how to model DAS using the VTMs under evaluation, besides the tasks to be performed for the training of the subjects.

Modeling Context-aware Features in Dynamic Software Product Lines

Experimental Task

Group 1 – Task A (Smart Home with eCFM)

Start Time:______

The Smart Home Story

The Smart Home DSPL is a complex software system which aims to facilitate the daily routine of people. It improves the home automation and integrates electronics and devices with the home. By means of sensors this system can identify changes in the environment and can activate and deactivate features of the software system.

Our Smart Home DSPL is composed by some groups of features, which are described as follows:

1) Security - Optionally, the smart home software product has a security system composed by some subsystems:
    a. An optional request assistance function by calling the police or calling a neighbor. Both functions can be used at the same time;
    b. An optional alarm functionality, which may happen through sirens or blinking lights. Both functions can be used at the same time;
    c. An optional functionality to simulate user presence in the house;

2) Illumination - All smart home software products have an illumination system composed by:
    a. An optional automated illumination mode which turns the lights on according to the user needs.
    b. A user illumination mode that allows the user to have always the control.

3) Temperature - All smart home software products have a temperature control system composed by:
    a. The temperature may be controlled by the air conditioning, which can use Cold air mode or Hot air mode. At most one of these modes could be activated;
    b. Besides, the window may also be used to control the airflow making the environment colder or warmer.

4) Notifications - Optionally, the smart home software product has a notifications system, composed by:
    a. Voice, message, or displays to present notifications. All modes can be used at the same time.

5) Calls - Optionally, the smart home software product has a calls system, composed by:
    a. VoIP or Phone. At most one of these modes should be activated.

6) Constraints among system features:
    a. When the system requests assistance, the calling functionality must be available;
    b. When the system requests assistance, the message functionality must be available;
    c. When the alarm is triggered, the notifications must be available;
    d. When the air conditioning is working, the windows must be closed.

The context information guides the different DSPL adaptations, thus, the context states activate and deactivate the system features, as follows:

1. **Robbery** is composed by **threat** and **attempted** context states, since:
    a. Threat activates the presence illusion mode;
    b. Attempted activates both request assistance and alarm;
    c. The threat and attempted context states may occur at the same time.

2. **Time** receives information about the time of day, it is composed by **night** context state, since:
    a. Night activates automated illumination and deactivates calling to neighbor;
    b. Night is an optional context state, it can be occurring or not.

3. **Temperature** receives information about the environment weather, it is composed by **warm** and **cold** context states, since:
    a. Warm turn the Cold Air on;
    b. Cold turn the Hot Air on;
    c. Warm and cold may not occur at the same time.

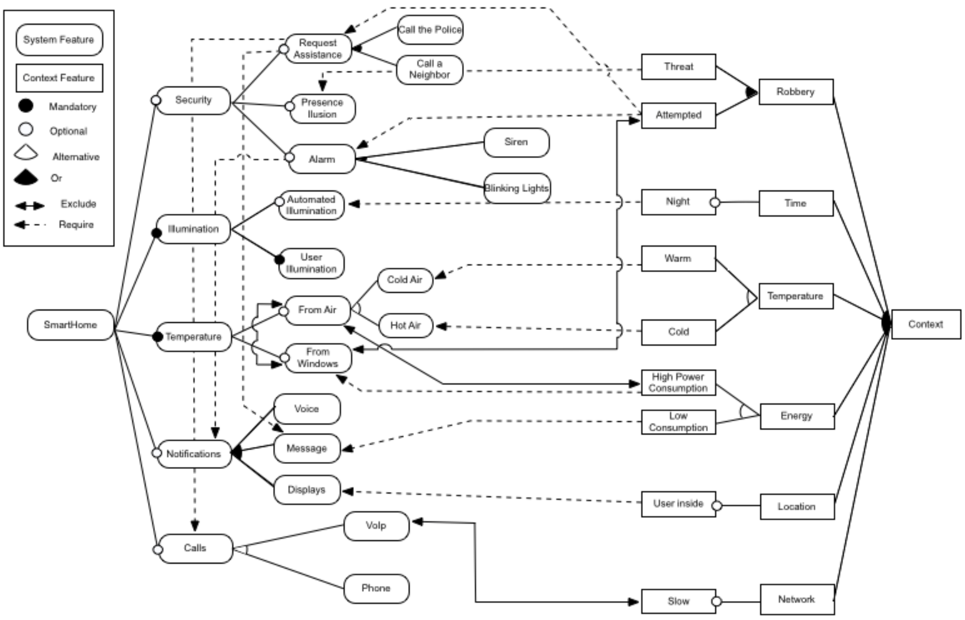
4. **Energy** receives information about the energy consumption, it is composed by **High power consumption**, **Low power consumption**, since:
    a. High power consumption deactivates From Air;
    b. Low power consumption activates message notifications;

c. High power consumption and Low power consumption may not occur at the same time.

5. **Location** receives information about the user location, it is composed by **User inside**, since:
    a. User inside activates voice message mode and turns displays on;
    b. User inside is an optional context state, it can be occurring or not.

6. **Network** receives information about the network speed, it is composed by **Slow**, since:
    a. Slow deactivates VoIp calls mode;
    b. Slow is an optional context state, it can be occurring or not.

Smart Home DSPL eCFM:



Comprehension and Modeling Questions:

1. Are there adaptation rules (Context (de)activates Features) in the model that are not mentioned in the story? If so, name them:

2. Are there adaptation rules (Context (de)activates Features) mentioned in the story, which have NOT been modeled? If so, name them:

3. Is there any relationship among contexts in the story indicating that they cannot be occurring at the same time? If so, describe it (them):

4. How many adaptation rules have been modeled? Describe them in the form *Context (de)active Feature in the following table:*

Context	(de)active	Feature

5. In which context situations the smart home will active the feature “from Air”, based on the model only?

6. What is the system configuration when is Night, Warm and the User is inside the house? Describe the active features (e.g., Feature A, Feature B):

7. According to the model, can the weather be Cold and Warm at the same time? If so, What is the configuration of the system in this case?

8. How do you modify the model to include the fact that when there is a Robbery in Action, which may happen at the same time of the Robbery Attempt or Threat, the system should Call the Police?

9. How do you modify the model to include the fact that when there are users outside the house, the system should activates the message notification?

END Time:______

Group 1 – Task B (Mobile Guide with CFM)

Start Time:______

The Mobile Visit Guide Story

The Mobile Visit Guide is a DSPL of mobile applications that show to the visitor information about the place s/he is visiting. By means of sensors this system can identify changes in the environment and activate and deactivate features of the mobile application.

Our Mobile Guide DSPL is composed by some groups of features, which are described as follows:

1) Show Files - All Mobile Visit Guide software products have a function to show files. This feature is composed by:
    a) A functionally in charge of providing audio content to the user;
    b) A functionally in charge of providing video content to the user;
    c) A functionally in charge of providing image content to the user;
    d) The Video, Image, and Audio functionalities may occur at the same time;
    e) All products of this line provide text content to the user. The type of text can be Normal or Cloud, and these types may not occur at the same time.

2) Show Events - Optionally, the products of the Mobile Guide can show events to the visitors. This feature is composed by:
    a) An optional function to show all events;
    b) An optional function to show local events.

3) RoadMap - Optionally, the products of the Mobile Guide can define a route, providing a roadmap to the user. This feature is composed by:
    a) A function to provide a roadmap in Normal mode;
    b) A function to provide a roadmap in 3D mode;
    c) Normal and 3D mode may occur at the same time.

The context information guides the different DSPL adaptations, thus, the context states activate and deactivate the system features, as follows:

1. **Slow network**, which deactivates 3D Roadmap functionality;

2. **Low Battery**, which deactivates both Image and Video;

3. **Normal Battery**, which activates Image;

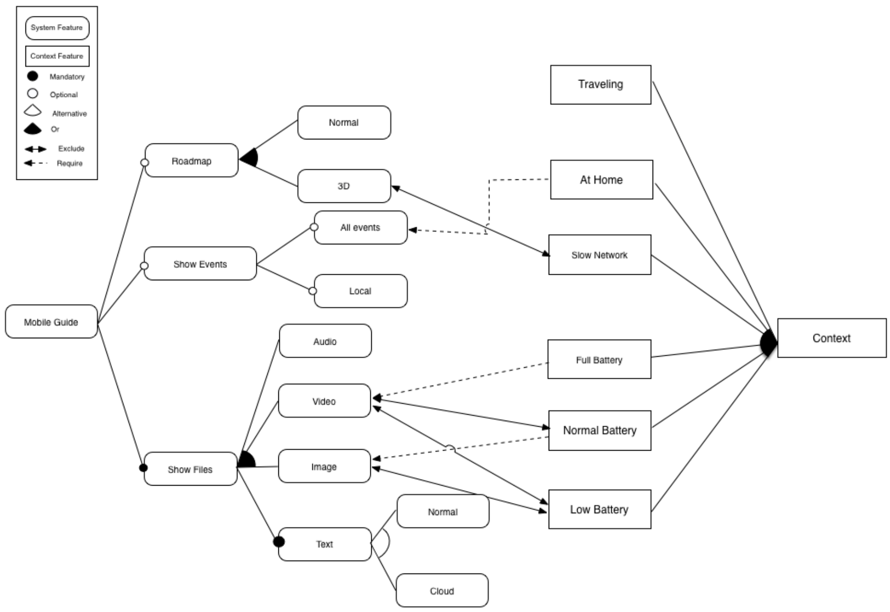
4. **Full Battery**, which activates both Image and Video;

5. The context states **Full Battery**, **Normal Battery** and **Low Battery** may not occur at the same time.

6. **Traveling**, which activates Local Events functionality;

7. **Travelling** and **At home** may not occur at the same time;

Mobile Visit Guide DSPL CFM:



Comprehension and Modeling Questions:

1. Are there adaptation rules (Context (de)activates Features) in the model that are not mentioned in the story? If so, name them:

2. Are there adaptation rules (Context (de)activates Features) mentioned in the story, which have NOT been modeled? If so, name them:

3. Is there any relationship among contexts in the story indicating that they cannot be occurring at the same time? If so, describe it (them):

4. How many adaptation rules have been modeled? Describe them in the form *Context (de)active Feature in the following table:*

Context	(de)active	Feature

5. In which context situations the user will have the feature Image activated based on the model only?

6. What is the system configuration when the Network is “slow” and the Battery is “Normal”? Describe the active features (e.g., Feature A, Feature B):

7. According to the model, can the battery charge level be Full and Low at the same time? If so, What is the configuration of the system in this case?

8. How do you modify the model to include the fact that when the user location is Seminar Room, it is activated the Cloud text feature?

9. How do you modify the model to include the fact that the when there is a earphone connected it is activated the Audio?

END Time:______

Modeling Context-aware Features in Dynamic Software Product Lines

Experimental Task: Modeling Context-aware Features

Group 2 – Task A (Smart Home with CFM)

Start Time:______

The Smart Home Story

The Smart Home DSPL is a complex software system which aims to facilitate the daily routine of people. It improves the home automation and integrates electronics and devices with the home. By means of sensors this system can identify changes in the environment and can activate and deactivate features of the software system.

Our Smart Home DSPL is composed by some groups of features, which are described as follows:

1) Security - Optionally, the smart home software product has a security system composed by some subsystems:
    a. An optional request assistance function by calling the police or calling a neighbor. Both functions can be used at the same time;
    b. An optional alarm functionality, which may happen through sirens or blinking lights. Both functions can be used at the same time;
    c. An optional functionality to simulate user presence in the house;

2) Illumination - All smart home software products have an illumination system composed by:
    a. An optional automated illumination mode which turns the lights on according to the user needs.
    b. A user illumination mode that allows the user to have always the control.

3) Temperature - All smart home software products have a temperature control system composed by:
    a. The temperature may be controlled by the air conditioning, which can use Cold air mode or Hot air mode. At most one of these modes could be activated;
    b. Besides, the window may also be used to control the airflow making the environment colder or warmer.

4) Notifications - Optionally, the smart home software product has a notifications system, composed by:
    a. Voice, message, or displays to present notifications. All modes can be used at the same time.

5) Calls - Optionally, the smart home software product has a calls system, composed by:
    a. VoIP or Phone. At most one of these modes should be activated.

6) Constraints among system features:
    a. When the system requests assistance, the calling functionality must be available;
    b. When the system requests assistance, the message functionality must be available;
    c. When the alarm is triggered, the notifications must be available;
    d. When the air conditioning is working, the windows must be closed.

The context information guides the different DSPL adaptations, thus, the context states activate and deactivate the system features, as follows:

1. **Threat of robbery** activates the presence illusion mode;

2. **Robbery Attempted** activates both request assistance and alarm;

3. **Night** activates automated illumination and deactivates calling to neighbor;

4. **High temperature** turns the Cold air on;

5. **Low temperature** turns the Hot air on;

6. **High temperature** and **Low temperature** may not occur at the same time.

7. **High consumption** deactivates From Air;

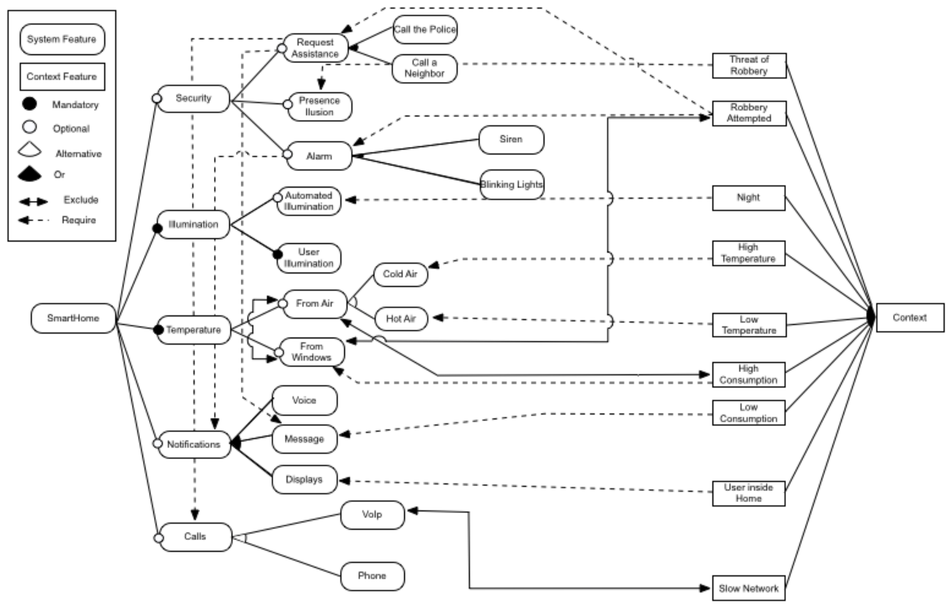
8. **Low consumption** activates message notifications;

9. **High consumption** and **Low consumption** may not occur at the same time;

10. **User inside Home** activates voice message mode and turns displays on;

11. **Slow Network** deactivates VoIp calls mode;

Smart Home DSPL CFM:



Comprehension and Modeling Questions:

1. Are there adaptation rules (Context (de)activates Features) in the model that are not mentioned in the story? If so, name them:

2. Are there adaptation rules (Context (de)activates Features) mentioned in the story, which have NOT been modeled? If so, name them:

3. Is there any relationship among contexts in the story indicating that they cannot be occurring at the same time? If so, describe it (them):

4. How many adaptation rules have been modeled? Describe them in the form *Context (de)active Feature* in the following table*:*

Context	(de)active	Feature

5. In which context situations the smart home will active the feature “from Air” based on the model only?

6. What is the system configuration when is Night, Warm (High Temperature) and the User is inside the house? Describe the active features (e.g., Feature A, Feature B):

7. According to the model, can the weather be Cold (low temperature) and Warm (high temperature) at the same time? If so, What is the configuration of the system in this case?

8. How do you modify the model to include the fact that when there is a Robbery in Action, which may happen at the same time of the Robbery Attempt or Threat, the system should Call the Police?

9. How do you modify the model to include the fact that when there are users outside the house, the system should activates the message notification?

END Time:______

Group 2 – Task B (Mobile Guide with eCFM)

Start Time:______

The Mobile Visit Guide Story

The Mobile Visit Guide is a DSPL of mobile applications that show to the visitor information about the place s/he is visiting. By means of sensors this system can identify changes in the environment and activate and deactivate features of the mobile application.

Our Mobile Guide DSPL is composed by some groups of features, which are described as follows:

1) Show Files - All Mobile Visit Guide software products have a function to show files. This feature is composed by:
    a) A functionally in charge of providing audio content to the user;
    b) A functionally in charge of providing video content to the user;
    c) A functionally in charge of providing image content to the user;
    d) The Video, Image, and Audio functionalities may occur at the same time;
    e) All products of this line provide text content to the user. The type of text can be Normal or Cloud, and these types may not occur at the same time.

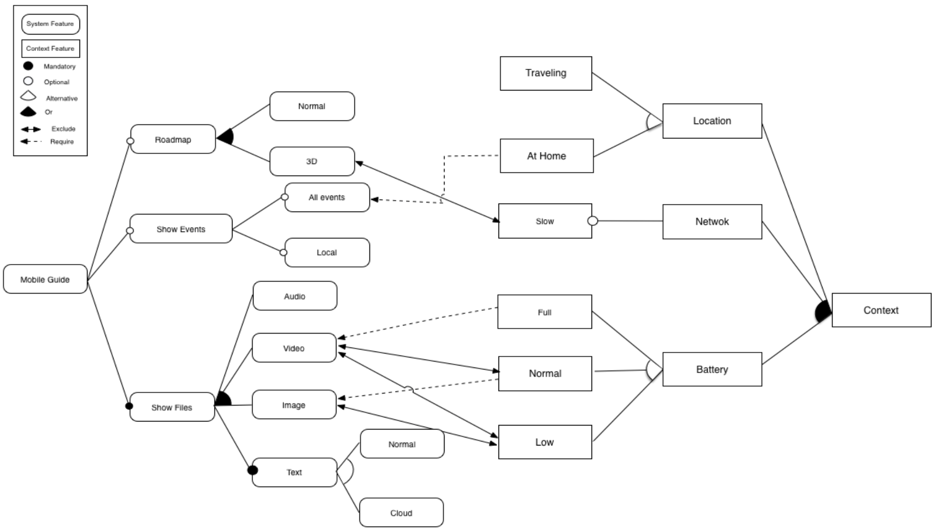
2) Show Events - Optionally, the products of the Mobile Guide can show events to the visitors. This feature is composed by:
    a) An optional function to show all events;
    b) An optional function to show local events.

3) RoadMap - Optionally, the products of the Mobile Guide can define a route, providing a roadmap to the user. This feature is composed by:
    a) A function to provide a roadmap in Normal mode;
    b) A function to provide a roadmap in 3D mode;
    c) Normal and 3D mode may occur at the same time.

The context information guides the different DSPL adaptations, thus, the context states activate and deactivate the system features, as follows:

1. **Network** receives information about the network speed, it is composed by **Slow**, since:
    a. Slow, which deactivates 3D Roadmap functionality;
    b. Slow is an optional context state, it can occurring or not.
2. **Battery** receives information about the battery charge level, it is composed by **Low Battery**, **Normal Battery** and **Full Battery**, since:
    a. Low Battery, which deactivates both Image and Video;
    b. Normal Battery, which activates Image;
    c. Full Battery, which activates both Image and Video;
    d. The context states Full, Normal and Low may not occur at the same time.
3. **Location** receives information about the current user location, it is composed by **Travelling** and **At Home**, since:
    a. Traveling, which activates Local Events functionality;
    b. Travelling and At home may not occur at the same time;

Mobile Visit Guide DSPL eCFM:



Comprehension and Modeling Questions:

1. Are there adaptation rules (Context (de)activates Features) in the model that are not mentioned in the story? If so, name them:

2. Are there adaptation rules (Context (de)activates Features) mentioned in the story, which have NOT been modeled? If so, name them:

3. Is there any relationship among contexts in the story indicating that they cannot be occurring at the same time? If so, describe it (them):

4. How many adaptation rules have been modeled? Describe them in the form *Context (de)active Feature* in the following table*:*

Context	(de)active	Feature

5. In which context situations the user will have the feature Image activated based on the model only?

6. What is the system configuration when the Network is “slow” and the Battery is “Normal”? Describe the active features (e.g., Feature A, Feature B):

7. According to the model, can the battery charge level be Full and Low at the same time? If so, What is the configuration of the system in this case?

8. How do you modify the model to include the fact that when the User Location is Seminar Room, it is activated the Cloud text feature?

9. How do you modify the model to include the fact that the earphone connected it is activated the Audio?

END Time:______

A.3 FEEDBACK FORM

This section presents the feedback form used to obtain answers about tasks that were performed by subjects. It reports the strengths and weaknesses of each VMT, besides of difficulties and problems found during the empirical study.

Modeling Dynamic Software Product Lines - Feedback Form

Feedback form of the controlled experiment on the use of DSPL variability modeling techniques.

*Obrigatório

1. **Full Name** *

2. **e-mail**

Please read carefully each statement before answering .

3. **I had enough time to perform the experiment tasks**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree						Totally agree

4. **The objectives of the experiment were perfectly clear to me**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree						Totally agree

5. **The description of the story was clear**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree						Totally agree

6. **The tasks were perfectly clear to me**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree						Totally agree

Context Feature Model (CFM)

7. **I experienced NO difficulty in understanding the CFM**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree	◯	◯	◯	◯	◯	Totally agree

8. **I experienced NO difficult to answer the first group of questions (1 - 3) with CFM model**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree	◯	◯	◯	◯	◯	Totally agree

9. **I experienced NO difficult to answer the second group of questions (4 - 7) with the CFM model**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree	◯	◯	◯	◯	◯	Totally agree

10. **I experienced NO difficult to answer the third group of questions (8 - 9) with the CFM model**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree	◯	◯	◯	◯	◯	Totally agree

11. **Do you think that technique CFM needs anything to improve its ease of use?**

Extended Context Feature Model (eCFM)

12. **I experienced NO difficulty in understanding the eCFM**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree	◯	◯	◯	◯	◯	Totally agree

13. **I experienced NO difficult to answer the first group of questions (1 - 3) with eCFM model**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree	◯	◯	◯	◯	◯	Totally agree

14. **I experienced NO difficult to answer the first group of questions (4 - 7) with eCFM model**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree	◯	◯	◯	◯	◯	Totally agree

15. **I experienced NO difficult to answer the first group of questions (8 - 9) with eCFM model**
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree	◯	◯	◯	◯	◯	Totally agree

16. **Do you think that technique eCFM needs anything to improve its ease of use?**

Final Questions

17. **I believe that technique extended CFM (eCFM) is more expressive then the technique CFM** *
Choose one that you agree the most
Marcar apenas uma oval.

	1	2	3	4	5	
Totally disagree	◯	◯	◯	◯	◯	Totally agree

18. **For me, the following technique is easier to learn and use** *
Choose the technique that you think easier for leaning
Marcar apenas uma oval.

◯ eCFM - extended Context Feature Model
◯ CFM - Context Feature Model

19. **Overall, what is the technique that you consider more suitable for DSPL modeling? Why?** *

20. **Do you have any comments about the training, activities or/and exercise?** *

Appendix B

ARTIFACTS OF THE EXPLORATORY STUDIES

B.1 EXPLORATORY STUDIES

This section presents all the data used to run the exploratory studies reported in Chapter 5 and Chapter 6.

B.1.1 Mobile Game DAS

Mobile Game DAS

Features							Contexts							
ID-Feature	**Name**	**Parent feature**	**Child feature**	**Type**	**Require**	**Exclude**	**ID-Context**	**Name**	**Context root**	**Context group**	**Context feature**	**Type**	**Require**	**Exclude**
F0	Mobile Game	--	F1,F4,F9	Mandatory	--	--	**C0**	Context	--	--	--	Mandatory	--	--
F1	Gaming Environment	F0	F2,F3	Optional	--	--	**C1**	Battery Level	C0	--	C2,C3	OR	--	--
F2	Single Player	F1	--	Alternative	--	--	**C2**	Battery < 30%	C1	C1	--	Alternative	F2,F7,F10	F3,F5,F8,F11
F3	Multiplayer	F1	--	Alternative	--	--	**C3**	Battery >= 30%	C1	C1	--	Alternative	F3,F5,F8,F11	F2,F7,F10
F4	Sound Effect	F0	F5,F6	Optional	--	--	**C4**	Network Available	C0	--	C5,C6	OR	--	--
F5	Vibration	F4	--	Optional	--	--	**C5**	WiFi connected	C4	C4	--	OR	F3,F5,F8,F11	F7
F6	Quality	F4	F7,F8	Mandatory	--	--	**C6**	Mobile Network connected	C4	C4	--	OR	F7,F10	F3,F5,F8,F11
F7	Low	F6	--	Alternative	--	--	**C7**	Mobile Data Ending	C0	--	C8,C9	OR	--	--
F8	High	F6	--	Alternative	--	--	**C8**	Using Mobile connection	C7	C7	--	Mandatory	F2,F7,F10	F3,F5,F8,F11
F9	Graphics Quality	F0	F10,F11	Mandatory	--	--	**C9**	Data Usage >= 70%	C7	C7	--	Mandatory	F2,F7,F10	F3,F5,F8,F11
F10	Low	F9	--	Alternative	--	--								
F11	High	F9	--	Alternative	--	--								

Goals	
ID	**Name**
G1	Providing Gaming Environment
G2	Set Sound Effect
G3	Adjust
G4	Adjust Sound

Softgoals	
ID	**Name**
SG1	High Quality Interaction
SG2	Efficiency in Resource

Features	
ID	**Name**
F0	Mobile Game
F1	Gaming Environment
F2	Single Player
F3	Multiplayer
F4	Sound Effect
F5	Vibration
F6	Quality
F7	Low
F8	High
F9	Graphics Quality
F10	Low
F11	High

Contexts	
ID	**Name**
C0	Context
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network
C5	WiFi connected
C6	Mobile Network
C7	Mobile Data
C8	Using Mobile
C9	Data Usage >=

Contexts	
ID	**Name**
C1	Battery Level
C2	Battery < 30%
C3	Battery >=
C4	Network
C5	WiFi
C6	Mobile
C7	Mobile Data
C8	Using Mobile
C9	Data Usage

1- Mobile Game Full

Softgoal	Sg1	Sg2		Softgoal	Sg1	Sg2	Sum	ivalue
Sg1	1	1	Normalize ->	Sg1	0.5	0.5	1	0.5
Sg2	1	1		Sg2	0.5	0.5	1	0.5
Sum	2	2		Sum	1	1	2	1

Number of softgoals	2
The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	2

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely	Extreme preferred
1	2	3	4	5	6	7	8	9

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9	10
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46	1.49
Random index (RI)	0									

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	Name
F1	Gaming Environment
F2	Single Player
F3	Multiplayer
F4	Sound Effect
F5	Vibration
F7	Low
F8	High
F10	Low
F11	High

Softgoals	
ID	Name
SG1	High Quality Interaction
SG2	Efficiency in Resource Usage

impDegree(Softgoal)		
Feature	Sg1	Sg2
F1	0	0
F2	-0.5	0
F3	0.5	-0.5
F4	0	0
F5	0.5	-0.5
F7	-0.5	0.5
F8	0.5	-0.5
F10	-0.5	0.5
F11	0.5	-0.5

Feature over NFRs (Contribution)	
Feature	Cont
F1	0
F2	-0.25
F3	0
F4	0
F5	0
F7	0
F8	0
F10	0
F11	0

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	2	0.3333333333	
C2	OR	1	0.3333333333
C3	OR	1	0.3333333333
C4	3	0.25	
C5	OR	1	0.25
C6	OR	1	0.25
C7	1	0.5	
C8	AND	0.5	0.25
C9	AND	0.5	0.25

ImpDegree(Context)						
Feature	C2	C3	C5	C6	C8	C9
F1	0	0	0	0	0	0
F2	1	-1	0	0	1	1
F3	-1	1	1	-1	0	-1
F4	0	0	0	0	0	0
F5	-1	1	1	-1	-1	-1
F7	1	-1	-1	1	1	1
F8	-1	1	1	-1	-1	-1
F10	1	0	0	1	1	1
F11	-1	1	1	-1	-1	-1

Feature (Co
Feature
F1
F2
F3
F4
F5
F7
F8
F10
F11

Contexts	
ID	Name
C2	Battery < 30%
C3	Battery >= 30%

Contexts	
ID	Name
C5	WiFi connected
C6	Mobile Network connected

C
ID
C8
C9

over Context (Contribution)
Cont
0
0.5
-0.25
0
0.5
0.5
-0.5
1.083333333
-0.5

Goal-Priorization					
Goal/HardGoal	Rank/Relation		RankValue		
G1	1		0.5		
Hg1 (F2)	OR		1		0.5
Hg2 (F3)	OR		1		0.5
G2	1		0.5		
Hg3 (F5)	OR		1		0.5
G4	OR	1	1	0.5	0.5
Hg4 (F7)	OR		1		0.5
Hg5 (F8)	OR		1		0.5
G3	1		0.5		
Hg6 (F10)	OR		1		0.5
Hg7 (F11)	OR		1		0.5

Feature over Goals (Contribution)	
Feature	Cont
F1	0
F2	0.5
F3	0.5
F4	0
F5	0.5
F7	0.5
F8	0.5
F10	0.5
F11	0.5

Objective function		
Feature	Utility value	Result
F0	0	xf0 = 1
F1	0	xf1 = 1
F2	0.75	xf2 = 1
F3	0.25	xf3 = 0
F4	0	xf4 = 1
F5	1	xf5 = 1
F6	0	xf6 = 0
F7	1	xf7 = 1
F8	0	xf8 = 0
F9	0	xf9 = 0
F10	1.583333333	xf10 = 1
F11	0	xf11 = 0

ontexts
Name
Using Mobile connection
Data Usage >= 70%

Goals	
ID	Name
G1	Providing Gaming Environment
G2	Set Sound Effect
G3	Adjust Graphic Quality
G4	Adjust Sound Quality

2- Mobile Game - Scenario 1 - CCF1 - Nenhum contexto

Softgoal	Sg1	Sg2		Softgoal	Sg1	Sg2	Sum	ivalue
Sg1	1	3	Normalize ->	Sg1	0.75	0.75	1.5	0.75
Sg2	0.3333333333	1		Sg2	0.25	0.25	0.5	0.25
Sum	1.333333333	4		Sum	1	1	2	1

Number of softgoals	2
The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	2

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely	Extreme preferred
1	2	3	4	5	6	7	8	9

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9	10
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46	1.49
Random index (RI)	0									

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	Name
F1	Gaming Environment
F2	Single Player
F3	Multiplayer
F4	Sound Effect
F5	Vibration
F7	Low
F8	High
F10	Low
F11	High

Softgoals	
ID	Name
SG1	High Quality Interaction
SG2	Efficiency in Resource Usage

impDegree(Softgoal)		
Feature	Sg1	Sg2
F1	0	0
F2	-0.5	0
F3	0.5	-0.5
F4	0	0
F5	0.5	-0.5
F7	-0.5	0.5
F8	0.5	-0.5
F10	-0.5	0.5
F11	0.5	-0.5

Feature over NFRs (Contribution)	
Feature	Cont
F1	0
F2	-0.375
F3	0.25
F4	0
F5	0.25
F7	-0.25
F8	0.25
F10	-0.25
F11	0.25

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	2	0.3333333333	
C2	OR	1	0.3333333333
C3	OR	1	0.3333333333
C4	3	0.25	
C5	OR	1	0.25
C6	OR	1	0.25
C7	1	0.5	
C8	AND	0.5	0.25
C9	AND	0.5	0.25

ImpDegree(Context)						
Feature	C2	C3	C5	C6	C8	C9
F1	0	0	0	0	0	0
F2	0	0	0	0	0	0
F3	0	0	0	0	0	0
F4	0	0	0	0	0	0
F5	0	0	0	0	0	0
F7	0	0	0	0	0	0
F8	0	0	0	0	0	0
F10	0	0	0	0	0	0
F11	0	0	0	0	0	0

Feature (Con
Feature
F1
F2
F3
F4
F5
F7
F8
F10
F11

Contexts	
ID	Name
C2	Battery < 30%
C3	Battery >= 30%

Contexts	
ID	Name
C5	WiFi connected
C6	Mobile Network connected

C
ID
C8
C9

over Context (Contribution)
Cont
0
0
0
0
0
0
0
0
0

Goal-Priorization					
Goal/HardGoal	Rank/Relation		RankValue		
G1	1		0.5		
Hg1 (F2)	OR		1		0.5
Hg2 (F3)	OR		1		0.5
G2	1		0.5		
Hg3 (F5)	OR		1		0.5
G4	OR	1	1	0.5	0.5
Hg4 (F7)	OR		1		0.5
Hg5 (F8)	OR		1		0.5
G3	1		0.5		
Hg6 (F10)	OR		1		0.5
Hg7 (F11)	OR		1		0.5

Feature over Goals (Contribution)	
Feature	Cont
F1	0
F2	0.5
F3	0.5
F4	0
F5	0.5
F7	0.5
F8	0.5
F10	0.5
F11	0.5

Objective function		CCF-01
Feature	Utility value	Result
F0	0	xf0 = 1
F1	0	xf1 = 1
F2	0.125	xf2 = 0
F3	0.75	xf3 = 1
F4	0	xf4 = 1
F5	0.75	xf5 = 1
F6	0	xf6 = 1
F7	0.25	xf7 = 1
F8	0.75	xf8 = 0
F9	0	xf9 = 1
F10	0.25	xf10 = 0
F11	0.75	xf11 = 1

ontexts
Name
Using Mobile connection
Data Usage >= 70%

Goals			
ID	Name	ID	Name
G1	Providing Gaming Environment	G2	Set Sound Effect
		G3	Adjust Graphic Quality
		G4	Adjust Sound Quality

3- Mobile Game - Scenario 2 - CCF3 (c5,c6)

Softgoal	Sg1	Sg2		Softgoal	Sg1	Sg2	Sum	ivalue
Sg1	1	3	Normalize ->	Sg1	0.75	0.75	1.5	0.75
Sg2	0.3333333333	1		Sg2	0.25	0.25	0.5	0.25
Sum	1.333333333	4		Sum	1	1	2	1

Number of softgoals	2
The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	2

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely	Extreme preferred
1	2	3	4	5	6	7	8	9

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9	10
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46	1.49
Random index (RI)	0									

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	Name
F1	Gaming Environment
F2	Single Player
F3	Multiplayer
F4	Sound Effect
F5	Vibration
F7	Low
F8	High
F10	Low
F11	High

Softgoals	
ID	Name
SG1	High Quality Interaction
SG2	Efficiency in Resource Usage

impDegree(Softgoal)		
Feature	Sg1	Sg2
F1	0	0
F2	-0.5	0
F3	0.5	-0.5
F4	0	0
F5	0.5	-0.5
F7	-0.5	0.5
F8	0.5	-0.5
F10	-0.5	0.5
F11	0.5	-0.5

Feature over NFRs (Contribution)	
Feature	Cont
F1	0
F2	-0.375
F3	0.25
F4	0
F5	0.25
F7	-0.25
F8	0.25
F10	-0.25
F11	0.25

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	2	0.3333333333	
C2	OR	1	0.3333333333
C3	OR	1	0.3333333333
C4	3	0.25	
C5	OR	1	0.25
C6	OR	1	0.25
C7	1	0.5	
C8	AND	0.5	0.25
C9	AND	0.5	0.25

ImpDegree(Context)						
Feature	C2	C3	C5	C6	C8	C9
F1	0	0	0	0	0	0
F2	0	0	0	0	0	0
F3	0	0	1	-1	0	0
F4	0	0	0	0	0	0
F5	0	0	1	-1	0	0
F7	0	0	-1	1	0	0
F8	0	0	1	-1	0	0
F10	0	0	0	1	0	0
F11	0	0	1	-1	0	0

Feature (Con
Feature
F1
F2
F3
F4
F5
F7
F8
F10
F11

Contexts	
ID	Name
C2	Battery < 30%
C3	Battery >= 30%

Contexts	
ID	Name
C5	WiFi connected
C6	Mobile Network connected

C
ID
C8
C9

over Context (Contribution)
Cont
0
0
0
0
0
0
0
0.25
0

Goal-Priorization

Goal/HardGoal	Rank/Relation		RankValue		
G1	1		0.5		
Hg1 (F2)	OR		1		0.5
Hg2 (F3)	OR		1		0.5
G2	1		0.5		
Hg3 (F5)	OR		1		0.5
G4	OR	1	1	0.5	0.5
Hg4 (F7)	OR		1		0.5
Hg5 (F8)	OR		1		0.5
G3	1		0.5		
Hg6 (F10)	OR		1		0.5
Hg7 (F11)	OR		1		0.5

Feature over Goals (Contribution)

Feature	Cont
F1	0
F2	0.5
F3	0.5
F4	0
F5	0.5
F7	0.5
F8	0.5
F10	0.5
F11	0.5

Objective function / CCF-03

Feature	Utility value	Result
F0	0	**xf0 = 1**
F1	0	**xf1 = 1**
F2	0.125	**xf2 = 0**
F3	0.75	**xf3 = 1**
F4	0	**xf4 = 1**
F5	0.75	**xf5 = 1**
F6	0	**xf6 = 1**
F7	0.25	**xf7 = 0**
F8	0.75	**xf8 = 1**
F9	0	**xf9 = 1**
F10	0.5	**xf10 = 0**
F11	0.75	**xf11 = 1**

ontexts

Name
Using Mobile connection
Data Usage >= 70%

Goals

ID	Name
G1	Providing Gaming Environment
G2	Set Sound Effect
G3	Adjust Graphic Quality
G4	Adjust Sound Quality

4- Mobile Game - Scenario 3 - CCF7 (c5,c6,c8)

Softgoal	Sg1	Sg2		Softgoal	Sg1	Sg2	Sum	ivalue
Sg1	1	1	Normalize ->	Sg1	0.5	0.5	1	0.5
Sg2	1	1		Sg2	0.5	0.5	1	0.5
Sum	2	2		Sum	1	1	2	1

Number of softgoals	2

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	2

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely	Extreme preferred
1	2	3	4	5	6	7	8	9

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9	10
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46	1.49
Random index (RI)	0									

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	Name
F1	Gaming Environment
F2	Single Player
F3	Multiplayer
F4	Sound Effect
F5	Vibration
F7	Low
F8	High
F10	Low
F11	High

Softgoals	
ID	Name
SG1	High Quality Interaction
SG2	Efficiency in Resource Usage

impDegree(Softgoal)		
Feature	Sg1	Sg2
F1	0	0
F2	-0.5	0
F3	0.5	-0.5
F4	0	0
F5	0.5	-0.5
F7	-0.5	0.5
F8	0.5	-0.5
F10	-0.5	0.5
F11	0.5	-0.5

Feature over NFRs (Contribution)	
Feature	Cont
F1	0
F2	-0.25
F3	0
F4	0
F5	0
F7	0
F8	0
F10	0
F11	0

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	2	0.3333333333	
C2	OR	1	0.3333333333
C3	OR	1	0.3333333333
C4	3	0.25	
C5	OR	1	0.25
C6	OR	1	0.25
C7	1	0.5	
C8	AND	0.5	0.25
C9	AND	0.5	0.25

ImpDegree(Context)						
Feature	C2	C3	C5	C6	C8	C9
F1	0	0	0	0	0	0
F2	0	0	0	0	1	0
F3	0	0	1	-1	0	0
F4	0	0	0	0	0	0
F5	0	0	1	-1	-1	0
F7	0	0	-1	1	1	0
F8	0	0	1	-1	-1	0
F10	0	0	0	1	1	0
F11	0	0	1	-1	-1	0

Feature (Con
Feature
F1
F2
F3
F4
F5
F7
F8
F10
F11

Contexts	
ID	Name
C2	Battery < 30%
C3	Battery >= 30%

Contexts	
ID	Name
C5	WiFi connected
C6	Mobile Network connected

C
ID
C8
C9

over Context tribution)
Cont
0
0.25
0
0
0.25
0.25
-0.25
0.5
-0.25

Goal-Priorization					
Goal/HardGoal	Rank/Relation		RankValue		
G1	1		0.5		
Hg1 (F2)	OR		1		0.5
Hg2 (F3)	OR		1		0.5
G2	1		0.5		
Hg3 (F5)	OR		1		0.5
G4	OR	1	1	0.5	0.5
Hg4 (F7)	OR		1		0.5
Hg5 (F8)	OR		1		0.5
G3	1		0.5		
Hg6 (F10)	OR		1		0.5
Hg7 (F11)	OR		1		0.5

Feature over Goals (Contribution)	
Feature	Cont
F1	0
F2	0.5
F3	0.5
F4	0
F5	0.5
F7	0.5
F8	0.5
F10	0.5
F11	0.5

Objective function		CCF-07
Feature	Utility value	**Result**
F0	0	**xf0 = 1**
F1	0	**xf1 = 1**
F2	0.5	**xf2 = 0**
F3	0.5	**xf3 = 1**
F4	0	**xf4 = 1**
F5	0.75	**xf5 = 1**
F6	0	**xf6 = 1**
F7	0.75	**xf7 = 1**
F8	0.25	**xf8 = 0**
F9	0	**xf9 = 1**
F10	1	**xf10 = 1**
F11	0.25	**xf11 = 0**

ontexts
Name
Using Mobile connection
Data Usage >= 70%

Goals			
		G2	Set Sound Effect
ID	Name	G3	Adjust Graphic Quality
G1	Providing Gaming Environment	G4	Adjust Sound Quality

5- Mobile Game - Scenario 4 - CCF8 (c5,c6,c8,c9)

Softgoal	Sg1	Sg2
Sg1	1	0.2
Sg2	5	1
Sum	6	1.2

Normalize ->

Softgoal	Sg1	Sg2	Sum	ivalue
Sg1	).1666666667	0.1666666667	).3333333333	.166666666
Sg2	).8333333333	0.8333333333	1.666666667	.833333333
Sum	1	1	2	1

Number of softgoals	2

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	2

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely	Extreme preferred
1	2	3	4	5	6	7	8	9

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9	10
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46	1.49
Random index (RI)	0									

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	Name
F1	Gaming Environment
F2	Single Player
F3	Multiplayer
F4	Sound Effect
F5	Vibration
F7	Low
F8	High
F10	Low
F11	High

Softgoals	
ID	Name
SG1	High Quality Interaction
SG2	Efficiency in Resource Usage

impDegree(Softgoal)		
Featur e	Sg1	Sg2
F1	0	0
F2	-0.5	0
F3	0.5	-0.5
F4	0	0
F5	0.5	-0.5
F7	-0.5	0.5
F8	0.5	-0.5
F10	-0.5	0.5
F11	0.5	-0.5

Feature over NFRs (Contribution)	
Feature	Cont
F1	0
F2	-0.08333333333
F3	-0.3333333333
F4	0
F5	-0.3333333333
F7	0.3333333333
F8	-0.3333333333
F10	0.3333333333
F11	-0.3333333333

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	2	0.3333333333	
C2	OR	1	0.3333333333
C3	OR	1	0.3333333333
C4	3	0.25	
C5	OR	1	0.25
C6	OR	1	0.25
C7	1	0.5	
C8	AND	0.5	0.25
C9	AND	0.5	0.25

ImpDegree(Context)						
Feature	C2	C3	C5	C6	C8	C9
F1	0	0	0	0	0	0
F2	0	0	0	0	1	1
F3	0	0	1	-1	-1	-1
F4	0	0	0	0	0	0
F5	0	0	1	-1	-1	-1
F7	0	0	-1	1	1	1
F8	0	0	1	-1	-1	-1
F10	0	0	0	1	1	1
F11	0	0	1	-1	-1	-1

Feature (Con
Featur e
F1
F2
F3
F4
F5
F7
F8
F10
F11

Contexts	
ID	Name
C2	Battery < 30%
C3	Battery >= 30%

Contexts	
ID	Name
C5	WiFi connected
C6	Mobile Network connected

C
ID
C8
C9

over Context (…tribution)
Cont
0
0.5
-0.5
0
0.5
0.5
-0.5
0.75
-0.5

Goal-Priorization					
Goal/HardGoal	Rank/Relation		RankValue		
G1	1		0.5		
Hg1 (F2)	OR		1		0.5
Hg2 (F3)	OR		1		0.5
G2	1		0.5		
Hg3 (F5)	OR		1		0.5
G4	OR	1	1	0.5	0.5
Hg4 (F7)	OR		1		0.5
Hg5 (F8)	OR		1		0.5
G3	1		0.5		
Hg6 (F10)	OR		1		0.5
Hg7 (F11)	OR		1		0.5

Feature over Goals (Contribution)	
Feature	Cont
F1	0
F2	0.5
F3	0.5
F4	0
F5	0.5
F7	0.5
F8	0.5
F10	0.5
F11	0.5

CCF-08

Objective function		
Feature	Utility value	Result
F0	0	xf0 = 1
F1	0	xf1 = 1
F2	0.9166666667	xf2 = 1
F3	-0.3333333333	xf3 = 0
F4	0	xf4 = 1
F5	0.6666666667	xf5 = 1
F6	0	xf6 = 1
F7	1.333333333	xf7 = 1
F8	-0.3333333333	xf8 = 0
F9	0	xf9 = 1
F10	1.583333333	xf10 = 1
F11	-0.3333333333	xf11 = 0

…ontexts
Name
Using Mobile connection
Data Usage >= 70%

Goals			
ID	Name	ID	Name
G1	Providing Gaming Environment	G2	Set Sound Effect
		G3	Adjust Graphic Quality
		G4	Adjust Sound Quality

6- Mobile Game - Scenario 5 - CCF9 (c2)

Softgoal	Sg1	Sg2		Softgoal	Sg1	Sg2	Sum	ivalue
Sg1	1	0.2	Normalize ->	Sg1	).1666666667	0.1666666667	).3333333333	.166666666
Sg2	5	1		Sg2	).8333333333	0.8333333333	1.666666667	.833333333
Sum	6	1.2		Sum	1	1	2	1

Number of softgoals	2

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	2

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely	Extreme preferred
1	2	3	4	5	6	7	8	9

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9	10
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46	1.49
Random index (RI)	0									

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	Name
F1	Gaming Environment
F2	Single Player
F3	Multiplayer
F4	Sound Effect
F5	Vibration
F7	Low
F8	High
F10	Low
F11	High

Softgoals	
ID	Name
SG1	High Quality Interaction
SG2	Efficiency in Resource Usage

impDegree(Softgoal)		
Featur e	Sg1	Sg2
F1	0	0
F2	-0.5	0
F3	0.5	-0.5
F4	0	0
F5	0.5	-0.5
F7	-0.5	0.5
F8	0.5	-0.5
F10	-0.5	0.5
F11	0.5	-0.5

Feature over NFRs (Contribution)	
Feature	Cont
F1	0
F2	-0.08333333333
F3	-0.3333333333
F4	0
F5	-0.3333333333
F7	0.3333333333
F8	-0.3333333333
F10	0.3333333333
F11	-0.3333333333

Context-Priorization			
Context	Rank/Relatio n	RankValue	
C1	2	0.3333333333	
C2	OR	1	0.3333333333
C3	OR	1	0.3333333333
C4	3	0.25	
C5	OR	1	0.25
C6	OR	1	0.25
C7	1	0.5	
C8	AND	0.5	0.25
C9	AND	0.5	0.25

ImpDegree(Context)						
Feature	C2	C3	C5	C6	C8	C9
F1	0	0	0	0	0	0
F2	1	0	0	0	0	0
F3	-1	0	0	0	0	0
F4	0	0	0	0	0	0
F5	-1	0	0	0	0	0
F7	1	0	0	0	0	0
F8	-1	0	0	0	0	0
F10	1	0	0	0	0	0
F11	-1	0	0	0	0	0

Feature (Con
Featur e
F1
F2
F3
F4
F5
F7
F8
F10
F11

Contexts	
ID	Name
C2	Battery < 30%
C3	Battery >= 30%

Contexts	
ID	Name
C5	WiFi connected
C6	Mobile Network connected

C
ID
C8
C9

over Context tribution)
Cont
0
0.3333333333
-0.3333333333
0
0.3333333333
0.3333333333
-0.3333333333
0.3333333333
-0.3333333333

Goal-Priorization					
Goal/HardGoal	Rank/Relation		RankValue		
G1	1		0.5		
Hg1 (F2)	OR		1		0.5
Hg2 (F3)	OR		1		0.5
G2	1		0.5		
Hg3 (F5)	OR		1		0.5
G4	OR	1	1	0.5	0.5
Hg4 (F7)	OR		1		0.5
Hg5 (F8)	OR		1		0.5
G3	1		0.5		
Hg6 (F10)	OR		1		0.5
Hg7 (F11)	OR		1		0.5

Feature over Goals (Contribution)	
Feature	Cont
F1	0
F2	0.5
F3	0.5
F4	0
F5	0.5
F7	0.5
F8	0.5
F10	0.5
F11	0.5

Objective function		CCF-09
Feature	Utility value	Result
F0	0	**xf0 = 1**
F1	0	**xf1 = 1**
F2	0.75	**xf2 = 1**
F3	-0.1666666667	**xf3 = 0**
F4	0	**xf4 = 1**
F5	0.5	**xf5 = 1**
F6	0	**xf6 = 1**
F7	1.166666667	**xf7 = 1**
F8	-0.1666666667	**xf8 = 0**
F9	0	**xf9 = 1**
F10	1.166666667	**xf10 = 1**
F11	-0.1666666667	**xf11 = 0**

ontexts
Name
Using Mobile connection
Data Usage >= 70%

Goals			
ID	Name	ID	Name
G1	Providing Gaming Environment	G2	Set Sound Effect
		G3	Adjust Graphic Quality
		G4	Adjust Sound Quality

7- Mobile Game - Scenario 6 - CCF11(c2,c5,c6)

Softgoal	Sg1	Sg2
Sg1	1	0.2
Sg2	5	1
Sum	6	1.2

Normalize ->

Softgoal	Sg1	Sg2	Sum	ivalue
Sg1	.1666666667	0.1666666667	.3333333333	.166666666
Sg2	.8333333333	0.8333333333	1.666666667	.833333333
Sum	1	1	2	1

Number of softgoals	2

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	2

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely	Extreme preferred
1	2	3	4	5	6	7	8	9

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9	10
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46	1.49
Random index (RI)	0									

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	Name
F1	Gaming Environment
F2	Single Player
F3	Multiplayer
F4	Sound Effect
F5	Vibration
F7	Low
F8	High
F10	Low
F11	High

Softgoals	
ID	Name
SG1	High Quality Interaction
SG2	Efficiency in Resource Usage

impDegree(Softgoal)		
Feature	Sg1	Sg2
F1	0	0
F2	-0.5	0
F3	0.5	-0.5
F4	0	0
F5	0.5	-0.5
F7	-0.5	0.5
F8	0.5	-0.5
F10	-0.5	0.5
F11	0.5	-0.5

Feature over NFRs (Contribution)	
Feature	Cont
F1	0
F2	-0.08333333333
F3	-0.3333333333
F4	0
F5	-0.3333333333
F7	0.3333333333
F8	-0.3333333333
F10	0.3333333333
F11	-0.3333333333

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	2	0.3333333333	
C2	OR	1	0.3333333333
C3	OR	1	0.3333333333
C4	3	0.25	
C5	OR	1	0.25
C6	OR	1	0.25
C7	1	0.5	
C8	AND	0.5	0.25
C9	AND	0.5	0.25

ImpDegree(Context)						
Feature	C2	C3	C5	C6	C8	C9
F1	0	0	0	0	0	0
F2	1	0	0	0	0	0
F3	-1	0	1	-1	0	0
F4	0	0	0	0	0	0
F5	-1	0	1	-1	0	0
F7	1	0	-1	1	0	0
F8	-1	0	1	-1	0	0
F10	1	0	0	1	0	0
F11	-1	0	1	-1	0	0

Feature (Con
Feature
F1
F2
F3
F4
F5
F7
F8
F10
F11

Contexts	
ID	Name
C2	Battery < 30%
C3	Battery >= 30%

Contexts	
ID	Name
C5	WiFi connected
C6	Mobile Network connected

C
ID
C8
C9

over Context (…tribution)
Cont
0
0.3333333333
-0.3333333333
0
0.3333333333
0.3333333333
-0.3333333333
0.5833333333
-0.3333333333

Goal-Priorization					
Goal/HardGoal	Rank/Relation		RankValue		
G1	1		0.5		
Hg1 (F2)	OR		1		0.5
Hg2 (F3)	OR		1		0.5
G2	1		0.5		
Hg3 (F5)	OR		1		0.5
G4	OR	1	1	0.5	0.5
Hg4 (F7)	OR		1		0.5
Hg5 (F8)	OR		1		0.5
G3	1		0.5		
Hg6 (F10)	OR		1		0.5
Hg7 (F11)	OR		1		0.5

Feature over Goals (Contribution)	
Feature	Cont
F1	0
F2	0.5
F3	0.5
F4	0
F5	0.5
F7	0.5
F8	0.5
F10	0.5
F11	0.5

Objective function		CCF-11
Feature	Utility value	Result
F0	0	xf0 = 1
F1	0	xf1 = 1
F2	0.75	xf2 = 1
F3	-0.1666666667	xf3 = 0
F4	0	xf4 = 1
F5	0.5	xf5 = 1
F6	0	xf6 = 1
F7	1.166666667	xf7 = 1
F8	-0.1666666667	xf8 = 0
F9	0	xf9 = 1
F10	1.416666667	xf10 = 1
F11	-0.1666666667	xf11 = 0

…ontexts
Name
Using Mobile connection
Data Usage >= 70%

Goals			
ID	Name	ID	Name
G1	Providing Gaming Environment	G2	Set Sound Effect
		G3	Adjust Graphic Quality
		G4	Adjust Sound Quality

8- Mobile Game - Scenario 7 - CCF15(c2,c5,c6)

Softgoal	Sg1	Sg2		Softgoal	Sg1	Sg2	Sum	ivalue
Sg1	1	0.2	Normalize ->	Sg1	).1666666667	0.1666666667	).3333333333	.166666666
Sg2	5	1		Sg2	).8333333333	0.8333333333	1.666666667	.833333333
Sum	6	1.2		Sum	1	1	2	1

Number of softgoals	2
The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	2

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely	Extreme preferred
1	2	3	4	5	6	7	8	9

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9	10
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46	1.49
Random index (RI)	0									

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	Name
F1	Gaming Environment
F2	Single Player
F3	Multiplayer
F4	Sound Effect
F5	Vibration
F7	Low
F8	High
F10	Low
F11	High

Softgoals	
ID	Name
SG1	High Quality Interaction
SG2	Efficiency in Resource Usage

impDegree(Softgoal)		
Featur e	Sg1	Sg2
F1	0	0
F2	-0.5	0
F3	0.5	-0.5
F4	0	0
F5	0.5	-0.5
F7	-0.5	0.5
F8	0.5	-0.5
F10	-0.5	0.5
F11	0.5	-0.5

Feature over NFRs (Contribution)	
Feature	Cont
F1	0
F2	-0.08333333333
F3	-0.3333333333
F4	0
F5	-0.3333333333
F7	0.3333333333
F8	-0.3333333333
F10	0.3333333333
F11	-0.3333333333

Context-Priorization			
Context	Rank/Relatio n	RankValue	
C1	2	0.3333333333	
C2	OR	1	0.3333333333
C3	OR	1	0.3333333333
C4	3	0.25	
C5	OR	1	0.25
C6	OR	1	0.25
C7	1	0.5	
C8	AND	0.5	0.25
C9	AND	0.5	0.25

ImpDegree(Context)						
Feature	C2	C3	C5	C6	C8	C9
F1	0	0	0	0	0	0
F2	1	0	0	0	0	0
F3	-1	0	1	-1	0	0
F4	0	0	0	0	0	0
F5	-1	0	1	-1	0	0
F7	1	0	-1	1	0	0
F8	-1	0	1	-1	0	0
F10	1	0	0	1	0	0
F11	-1	0	1	-1	0	0

Feature (Con
Featur e
F1
F2
F3
F4
F5
F7
F8
F10
F11

Contexts	
ID	Name
C2	Battery < 30%
C3	Battery >= 30%

Contexts	
ID	Name
C5	WiFi connected
C6	Mobile Network connected

C
ID
C8
C9

over Context (Contribution)
Cont
0
0.3333333333
-0.3333333333
0
0.3333333333
0.3333333333
-0.3333333333
0.5833333333
-0.3333333333

Goal-Priorization					
Goal/HardGoal	Rank/Relation		RankValue		
G1	1		0.5		
Hg1 (F2)	OR		1		0.5
Hg2 (F3)	OR		1		0.5
G2	1		0.5		
Hg3 (F5)	OR		1		0.5
G4	OR	1	1	0.5	0.5
Hg4 (F7)	OR		1		0.5
Hg5 (F8)	OR		1		0.5
G3	1		0.5		
Hg6 (F10)	OR		1		0.5
Hg7 (F11)	OR		1		0.5

Feature over Goals (Contribution)	
Feature	Cont
F1	0
F2	0.5
F3	0.5
F4	0
F5	0.5
F7	0.5
F8	0.5
F10	0.5
F11	0.5

Objective function		CCF-15
Feature	Utility value	Result
F0	0	xf0 = 1
F1	0	xf1 = 1
F2	0.75	xf2 = 1
F3	-0.1666666667	xf3 = 0
F4	0	xf4 = 1
F5	0.5	xf5 = 1
F6	0	xf6 = 1
F7	1.166666667	xf7 = 1
F8	-0.1666666667	xf8 = 0
F9	0	xf9 = 1
F10	1.416666667	xf10 = 1
F11	-0.1666666667	xf11 = 0

ontexts
Name
Using Mobile connection
Data Usage >= 70%

Goals			
ID	Name	ID	Name
		G2	Set Sound Effect
		G3	Adjust Graphic Quality
G1	Providing Gaming Environment	G4	Adjust Sound Quality

9- Mobile Game - Scenario 8 - CCF16(c2,c3,c5,c6,c8,c9)

Softgoal	Sg1	Sg2		Softgoal	Sg1	Sg2	Sum	ivalue
Sg1	1	0.2	Normalize ->	Sg1	).1666666667	0.1666666667	).3333333333	.166666666
Sg2	5	1		Sg2	).8333333333	0.8333333333	1.666666667	.833333333
Sum	6	1.2		Sum	1	1	2	1

Number of softgoals	2

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	2

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely	Extreme preferred
1	2	3	4	5	6	7	8	9

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9	10
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46	1.49
Random index (RI)	0									

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	Name
F1	Gaming Environment
F2	Single Player
F3	Multiplayer
F4	Sound Effect
F5	Vibration
F7	Low
F8	High
F10	Low
F11	High

Softgoals	
ID	Name
SG1	High Quality Interaction
SG2	Efficiency in Resource Usage

impDegree(Softgoal)		
Feature	Sg1	Sg2
F1	0	0
F2	-0.5	0
F3	0.5	-0.5
F4	0	0
F5	0.5	-0.5
F7	-0.5	0.5
F8	0.5	-0.5
F10	-0.5	0.5
F11	0.5	-0.5

Feature over NFRs (Contribution)	
Feature	Cont
F1	0
F2	-0.08333333333
F3	-0.3333333333
F4	0
F5	-0.3333333333
F7	0.3333333333
F8	-0.3333333333
F10	0.3333333333
F11	-0.3333333333

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	2	0.3333333333	
C2	OR	1	0.3333333333
C3	OR	1	0.3333333333
C4	3	0.25	
C5	OR	1	0.25
C6	OR	1	0.25
C7	1	0.5	
C8	AND	0.5	0.25
C9	AND	0.5	0.25

ImpDegree(Context)						
Feature	C2	C3	C5	C6	C8	C9
F1	0	0	0	0	0	0
F2	1	-1	0	0	1	1
F3	-1	1	1	-1	-1	-1
F4	0	0	0	0	0	0
F5	-1	1	1	-1	-1	-1
F7	1	-1	-1	1	1	1
F8	-1	1	1	-1	-1	-1
F10	1	0	0	1	1	1
F11	-1	1	1	-1	-1	-1

Feature (Con
Feature
F1
F2
F3
F4
F5
F7
F8
F10
F11

Contexts	
ID	Name
C2	Battery < 30%
C3	Battery >= 30%

Contexts	
ID	Name
C5	WiFi connected
C6	Mobile Network connected

C
ID
C8
C9

over Context (Contribution)
Cont
0
0.5
-0.5
0
0.5
0.5
-0.5
1.083333333
-0.5

Goal-Priorization					
Goal/HardGoal	**Rank/Relation**		**RankValue**		
G1	1		0.5		
Hg1 (F2)	OR		1		0.5
Hg2 (F3)	OR		1		0.5
G2	1		0.5		
Hg3 (F5)	OR		1		0.5
G4	OR	1	1	0.5	0.5
Hg4 (F7)	OR		1		0.5
Hg5 (F8)	OR		1		0.5
G3	1		0.5		
Hg6 (F10)	OR		1		0.5
Hg7 (F11)	OR		1		0.5

Feature over Goals (Contribution)	
Feature	**Cont**
F1	0
F2	0.5
F3	0.5
F4	0
F5	0.5
F7	0.5
F8	0.5
F10	0.5
F11	0.5

Objective function		CCF-16
Feature	**Utility value**	**Result**
F0	0	**xf0 = 1**
F1	0	**xf1 = 1**
F2	0.9166666667	**xf2 = 1**
F3	-0.3333333333	**xf3 = 0**
F4	0	**xf4 = 1**
F5	0.6666666667	**xf5 = 1**
F6	0	**xf6 = 1**
F7	1.333333333	**xf7 = 1**
F8	-0.3333333333	**xf8 = 0**
F9	0	**xf9 = 1**
F10	1.916666667	**xf10 = 1**
F11	-0.3333333333	**xf11 = 0**

ontexts
Name
Using Mobile connection
Data Usage >= 70%

Goals	
ID	**Name**
G1	Providing Gaming Environment
G2	Set Sound Effect
G3	Adjust Graphic Quality
G4	Adjust Sound Quality

CCF 1 - MobileGame					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf9	xf4	xf3	0	0	0
xf2	xf5	xf4	0	0.5	-0.375
xf3	xf6	xf5	0	0.5	0.25
xf5	xf7	xf6	0	0.5	0.25
xf6	xf8	xf7	0	0	0
xf10	xf9	xf8	0	0.5	-0.25
xf11	xf10	xf9	0	0.5	0.25
xf7	xf11	xf10	0	0.5	-0.25
xf8	xf12	xf11	0	0.5	0.25

CCF 3 - MobileGame					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf9	xf4	xf3	0	0	0
xf2	xf5	xf4	0	0.5	-0.375
xf3	xf6	xf5	0	0.5	0.25
xf5	xf7	xf6	0	0.5	0.25
xf6	xf8	xf7	0	0	0
xf10	xf9	xf8	0.25	0.5	-0.25
xf11	xf10	xf9	0	0.5	0.25
xf7	xf11	xf10	0	0.5	-0.25
xf8	xf12	xf11	0	0.5	0.25

CCF 7 - MobileGame					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf9	xf4	xf3	0	0	0
xf2	xf5	xf4	0.25	0.5	-0.25
xf3	xf6	xf5	0	0.5	0
xf5	xf7	xf6	0.25	0.5	0
xf6	xf8	xf7	0	0	0
xf10	xf9	xf8	0.5	0.5	0
xf11	xf10	xf9	-0.25	0.5	0
xf7	xf11	xf10	0.25	0.5	0
xf8	xf12	xf11	-0.25	0.5	0

CCF 8 - MobileGame					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf9	xf4	xf3	0	0	0
xf2	xf5	xf4	0.25	0.5	-0.25
xf3	xf6	xf5	0	0.5	0
xf5	xf7	xf6	0.25	0.5	0
xf6	xf8	xf7	0	0	0
xf10	xf9	xf8	0.5	0.5	0
xf11	xf10	xf9	-0.25	0.5	0
xf7	xf11	xf10	0.25	0.5	0
xf8	xf12	xf11	-0.25	0.5	0

CCF 9 - MobileGame					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf9	xf4	xf3	0	0	0
xf2	xf5	xf4	0.5	0.5	-0.08333333333
xf3	xf6	xf5	-0.5	0.5	-0.3333333333
xf5	xf7	xf6	0.5	0.5	-0.3333333333
xf6	xf8	xf7	0	0	0
xf10	xf9	xf8	0.75	0.5	0.3333333333
xf11	xf10	xf9	-0.5	0.5	-0.3333333333
xf7	xf11	xf10	0.5	0.5	0.3333333333
xf8	xf12	xf11	-0.5	0.5	-0.3333333333

CCF 11 - MobileGame					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	-0.0833333333
xf4	xf3	xf2	0	0	0
xf9	xf4	xf3	0	0	0
xf2	xf5	xf4	0.3333333333	0.5	-0.0833333333
xf3	xf6	xf5	-0.3333333333	0.5	-0.3333333333
xf5	xf7	xf6	0.3333333333	0.5	-0.3333333333
xf6	xf8	xf7	0	0	0
xf10	xf9	xf8	0.5833333333	0.5	0.3333333333
xf11	xf10	xf9	-0.3333333333	0.5	-0.3333333333
xf7	xf11	xf10	0.3333333333	0.5	0.3333333333
xf8	xf12	xf11	-0.3333333333	0.5	-0.3333333333

CCF 15 - MobileGame					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf9	xf4	xf3	0	0	0
xf2	xf5	xf4	0.3333333333	0.5	-0.0833333333
xf3	xf6	xf5	-0.3333333333	0.5	-0.3333333333
xf5	xf7	xf6	0.3333333333	0.5	-0.3333333333
xf6	xf8	xf7	0	0	0
xf10	xf9	xf8	0.5833333333	0.5	0.3333333333
xf11	xf10	xf9	-0.3333333333	0.5	-0.3333333333
xf7	xf11	xf10	0.3333333333	0.5	0.3333333333
xf8	xf12	xf11	-0.3333333333	0.5	-0.3333333333

CCF 16 - MobileGame					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf9	xf4	xf3	0	0	0
xf2	xf5	xf4	0.5	0.5	-0.0833333333
xf3	xf6	xf5	-0.5	0.5	-0.3333333333
xf5	xf7	xf6	0.5	0.5	-0.3333333333
xf6	xf8	xf7	0	0	0
xf10	xf9	xf8	1.083333333	0.5	0.3333333333
xf11	xf10	xf9	-0.5	0.5	-0.3333333333
xf7	xf11	xf10	0.5	0.5	0.3333333333
xf8	xf12	xf11	-0.5	0.5	-0.3333333333

ILP x GA							
Scenario	**ConG4DaS**	**ToffA-DAS**	**Fitness (Context + Soft goal + Goal)**	**ToffA-DAS+**	**Fitness (Context)**	**Fitness (Soft goal)**	**Fitness (Goal)**
ccf1	f2,f5,f8,f11	f3,f5,f7,f11	2.5	f2,f7,f11	0	-0.375	1.5
ccf3	f2,f5,f8,f11	f3,f5,f8,f11	2.75	f7,f10	0.25	-0.5	1
ccf7	f3,f5,f7,f10	f3,f5,f7,f10	2.25	f3,f7,f11	-0.5	0	1.5
ccf8	f2,f7,f10	f2,f5,f7,f10	4.5	f3,f5,f8,f10	0	-0.375	1.5
ccf9	f2,f7,f10	f2,f5,f7,f10	3.58333	f8,f11	-1	-0.6666666666	1
ccf11	f3,f7,f10	f2,f5,f7,f10	3.83333	f8,f11	-0.6666666666	-0.6666666666	1
ccf15	f3,f7,f10	f2,f5,f7,f10	3.83333	f2,f8,f11	-0.3333333333	-0.7499999993	1.5
ccf16	f2,f7,f10	f2,f5,f7,f10	4.83333	f8,f11	-1	-0.6666666666	1

Configuration	Pos CON	Pos ToffA	Difference	Sign	Absolute value	Signed Rank
1	3	3	0	0	0	0
2	3	4	-1	-1	1	-8
3	4	4	0	0	0	0
4	2	2	0	0	0	0
5	2	2	0	0	0	0
6	2	2	0	0	0	0
7	2	2	0	0	0	0
8	2	2	0	0	0	0

0	*Positive Sum	
-8	*Negative Sum	
0	*Test Statistic (Ws)	[Ws = Wc] | The sample with n = 4 is not enough to return the Wc with the level of significance.
--	*Critical Value (Wc)	We do not reject H0. There is no difference between ToffA-DAS and ConG4DaS in terms of POS.

1	*Sample size

H0: **There is no difference beteween releases**
H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**
There is sufficient evidence to suggest that **there is difference** between the **ToffA-DAS** and **ConG4Das** in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we reject H0 | We reject H0 when Test Stat > Critical Value **[Ws > Wc]**
There is sufficient evidence to suggest that **there is difference** between the **ToffA-DAS** and **ConG4Das-DAS+** in terms of configurations by considering satisfaction levels.

Configuration	Neg CON	Neg ToffA	Difference	Sign	Absolute value	Signed Rank
1	1	1	0	0	0	0
2	1	0	1	1	1	6.5
3	4	4	0	0	0	0
4	0	1	-1	-1	1	-6.5
5	0	1	-1	-1	1	-6.5
6	1	1	0	0	0	0
7	1	1	0	0	0	0
8	0	1	-1	-1	1	-6.5

6.5	*Positive Sum	
-19.5	*Negative Sum	
6.5	*Test Statistic (Ws)	[Ws = Wc] | The sample with n = 4 is not enough to return the Wc with the level of significance.
--	*Critical Value (Wc)	We do not reject H0. There is not difference between CCA-DAS and ConG4DaS in terms of NEG.
4	*Sample size	

H0: **There is no difference beteween releases**
H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**
There is sufficient evidence to suggest that **there is difference** between the Toff**A-DAS** and **ConG4Das-DAS+** in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we reject H0 | We reject H0 when Test Stat > Critical Value **[Ws > Wc]**
There is not sufficient evidence to suggest that **there is difference** between the **ToffA-DAS** and **ConG4Das** in terms of configurations by considering satisfaction levels.

Configuration	Diff CON	Diff ToffA	Difference	Sign	Absolute value	Signed Rank
1	2	2	3	0	0	0
2	2	0	0	1	2	8
3	0	0	-1	0	0	0
4	2	1	-3	1	1	6
5	2	1	-1	1	1	6
6	1	1	-1	0	0	0
7	1	1	-1	0	0	0
8	2	1	1	1	1	6

26	*Positive Sum	
0	*Negative Sum	
0	*Test Statistic (Ws)	[Ws = Wc] | The sample with n = 4 is not enough to return the Wc with the level of significance.
--	*Critical Value (Wc)	We do not reject H0. There is not difference between CCA-DAS and ConG4DaS in terms of DIFF.
4	*Sample size	

H0: **There is no difference beteween releases**
H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**
There is sufficient evidence to suggest that **there is difference** between the **ToffA-DAS** and **ConG4Das** in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we reject H0 | We reject H0 when Test Stat > Critical Value **[Ws > Wc]**
There is not sufficient evidence to suggest that **there is difference** between the **ToffA-DAS** and **ConG4Das** in terms of configurations by considering satisfaction levels.

Configuration	Pos ToffA	Pos ToffA+	Difference	Sign	Absolute value	Signed Rank
1	4	2	2	1	2	6.5
2	4	2	2	1	2	6.5
3	4	3	1	1	1	2.5
4	3	4	-1	-1	1	-2.5
5	4	2	2	1	2	6.5
6	3	1	2	1	2	6.5
7	3	2	1	1	1	2.5
8	3	2	1	1	1	2.5

H0: **There is no difference beteween releases**
H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**
There is sufficient evidence to suggest that **there is difference** between the **ToffA-DAS** and **ToffA-DAS+** in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we don't reject H0 | We don't reject H0 when Test Stat > Critical Value **[Ws > Wc]**
There is not sufficient evidence to suggest that **there is difference** between the **ToffA-DAS** and **ToffA-DAS+** in terms of configurations by considering satisfaction levels.

33.5	*Positive Sum	
-2.5	*Negative Sum	
2.5	*Test Statistic (Ws)	[Ws < Wc]
3	*Critical Value (Wc)	We reject H0. There is difference between ToffA-DAS and ToffA-DAS+ in terms of POS.

8	*Sample size

Configuration	Neg ToffA	Neg ToffA+	Difference	Sign	Absolute value	Signed Rank
1	4	3	1	1	1	3
2	4	2	2	1	2	6.5
3	4	3	1	1	1	3
4	4	4	0	0	0	0
5	4	2	2	1	2	6.5
6	4	2	2	1	2	6.5
7	4	3	1	1	1	3
8	4	2	2	1	2	6.5

35	*Positive Sum	
0	*Negative Sum	
0	*Test Statistic (Ws)	[Ws < Wc]
2	*Critical Value (Wc)	We reject H0. There is difference between ToffA-DAS and ToffA-DAS+ in terms of NEG.
7	*Sample size	

H0: **There is no difference beteween releases**
H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**
There is sufficient evidence to suggest that **there is difference** between the **ToffA-DAS** and **ToffA-DAS+** in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we don't reject H0 | We don't reject H0 when Test Stat > Critical Value **[Ws > Wc]**
There is not sufficient evidence to suggest that **there is difference** between the **ToffA-DAS** and **ToffA-DAS+** in terms of configurations by considering satisfaction levels.

Configuration	Diff ToffA	Diff ToffA+	Difference	Sign	Absolute value	Signed Rank
1	0	-1	3	1	1	6.5
2	0	0	0	0	0	0
3	0	0	-1	0	0	0
4	-1	0	-3	-1	1	-6.5
5	-1	0	-1	-1	1	-6.5
6	-1	-1	-1	0	0	0
7	-1	-1	-1	0	0	0
8	-1	0	1	-1	1	-6.5

6.5	*Positive Sum	
-19.5	*Negative Sum	
6.5	*Test Statistic (Ws)	[Ws = Wc] | The sample with n = 4 is not enough to return the Wc with the level of significance.
--	*Critical Value (Wc)	We do not reject H0. There is not difference between ToffA-DAS and ToffA-DAS+ in terms of Diff.
4	*Sample size	

H0: **There is no difference beteween releases**
H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**
There is sufficient evidence to suggest that **there is difference** between the **ToffA-DAS** and Toff**A-DAS+** in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we don't reject H0 | We don't reject H0 when Test Stat > Critical Value **[Ws > Wc]**
There is not sufficient evidence to suggest that **there is difference** between the **ToffA-DAS** and **ToffA-DAS+** in terms of configurations by considering satisfaction levels.

B.1.2 Smart Home DAS

Smart Home DAS														
Features							Contexts							
ID-Feature	Name	Parent feature	Child feature	Type	Require	Exclude	ID-Context	Name	Context root	Context group	Context feature	Type	Require	Exclude
F0	Manage home	--	F1,F4,F7,F10	Mandatory	--	--	**C0**	Context	--	--	--	Mandatory	--	--
F1	Refresh air inside home	F0	F2,F3	Mandatory	--	--	**C1**	Not raning	C0	--	C2	OR	--	--
F2	Open windows	F1	--	Alternative	--	--	**C2**	No raning detected by sensor (rain = false)	C1	C1	--	Mandatory	F2	F3
F3	Turn air ventilator	F1	--	Alternative	--	--	**C3**	Low food stock	C0	--	C4	OR	--	--
F4	Provide meal suggestion	F0	F5,F6	Mandatory	--	--	**C4**	Food stock below 15% (food stock < 0.15)	C3	C3	--	Mandatory	F6,F12	F5
F5	Suggest home cocked meal	F4	--	Alternative	--	--	**C5**	There are people at home	C0	--	C6	OR	--	--
F6	Suggest restaurant meal	F4	--	Alternative	--	--	**C6**	At least one person inside the house (people detected >=1)	C5	C5	--	Mandatory	F9	F8
F7	Control lights	F1	F8,F9	Mandatory	--	--	**C7**	Low budget	C0	--	C8	OR	--	--
F8	Uccupancy simulation	F7	--	Alternative	--	--	**C8**	More than 75% of monthly bufget used (Budget usage >0.75)	C7	C7	--	Mandatory	F5	F6,F8
F9	Turn on lights in occupied rooms	F7	--	Alternative	--	--	**C9**	Night time	C0	--	C10	OR	--	--
F10	Control food stock	F0	F11,F12	Mandatory	--	--	**C10**	It is night time (dark = true)	C9	C9	--	Mandatory	F3	F2
F11	Update stock	F10	--	Mandatory	--	--								
F12	Warm tenant about low stock	F10	--	Optional	--	--								

Goals	
ID	Name
G1	Refresh inside home
G2	Meal suggestions
G3	Lights control

Softgoals	
ID	Name
SG1	Safety
SG2	Save money
SG3	Energy efficiency

Features	
ID	Name
F0	Manage home
F1	Refresh air inside home
F2	Open windows
F3	Turn on air ventilator
F4	Provide meal suggestion
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F7	Control lights
F8	Ocupancy Simulation
F9	Turn on lights in occupied rooms
F10	Control food stock
F11	Update stock
F11	Warm tenant about low stock

Contexts	
ID	Name
C0	Context
C1	Not raning
C2	No raning detected by sensor (rain = false)
C3	Low food stock
C4	Food stock below 15% (food stock < 0.15)
C5	There are people at home
C6	Aat least one person inside the house (people detected >=1)
C7	Low budget
C8	More than 75% of montly buget used (budget usage >
C9	Nigth time
C10	It is nigth time (dark = true)

1- Smart Home Full

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	1	1	Normalize ->	Sg1	0.3333333333	0.3333333333	0.3333333333	1	0.3333333333
Sg2	1	1	1		Sg2	0.3333333333	0.3333333333	0.3333333333	1	0.3333333333
Sg3	1	1	1		Sg3	0.3333333333	0.3333333333	0.3333333333	1	0.3333333333
Sum	3	3	3		Sum	1	1	1	3	1

Features	
ID	Name
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)		
Feature	Sg1	Sg2
F2	-0.5	0
F3	1	0
F5	0	1
F6	-0.5	0
F8	0.5	-1
F9	-0.5	0.5
F12	0	0

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely	Extreme preferred
1	2	3	4	5	6	7	8	9

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9	10
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46	1.49
Random index (RI)	0.58									

Softgoals	
ID	Name
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Sg3
1
-0.5
0
0
0
0
0

Feature over NFRs (Contribution)	
Feature	Cont
F2	0.1666666667
F3	0.1666666667
F5	0.3333333333
F6	-0.1666666667
F8	-0.1666666667
F9	0
F12	0

Context-Priorization			
Context	Rank/Relatio	RankValue	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	1	0	0	0	-1
F3	-1	0	0	0	1
F5	0	-1	0	1	0
F6	0	1	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	1	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	0
F3	0
F5	0
F6	0
F8	-1
F9	1
F12	0.5

Goal-Prioriz	
Goal	Rank/Relation
G1	1
Hg1 (F2)	OR
Hg2 (F3)	OR
G2	1
Hg3 (F5)	OR
Hg4 (F6)	OR
G3	1
Hg5 (F8)	OR
Hg6 (F9)	OR

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ation	
RankValue	
0.5	
1	0.5
1	0.5
0.5	
1	0.5
1	0.5
0.5	
1	0.5
1	0.5

Feature over Goals (Contribution)

Feature	Cont
F2	0.5
F3	0.5
F5	0.5
F6	0.5
F8	0.5
F9	0.5
F12	0

Objective function

CCF-Full

Feature	Utility value	Result	GA ID
F0	0	xf0 = 1	f1
F1	0	xf1 = 1	f2
F2	0.6666666667	xf2 = 1	f6
F3	0.6666666667	xf3 = 0	f7
F4	0	xf4 = 1	f3
F5	0.8333333333	xf5 = 1	f8
F6	0.3333333333	xf6 = 0	f9
F7	0	xf7 = 1	f4
F8	-0.6666666667	xf8 = 0	f10
F9	1.5	xf9 = 1	f11
F10	0	xf10 = 1	f5
F11	0	xf11 = 1	f12
F12	0.5	xf12 = 1	f13

Goals

ID	Name
G1	Refresh inside home
G2	Meal suggestions
G3	Lights control

2- Smart Home (Nenhuma CCF)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	0.3333333333	1		Sg1	0.2	0.2	0.2	0.6	0.2
Sg2	3	1	3	Normalize ->	Sg2	0.6	0.6	0.6	1.8	0.6
Sg3	1	0.3333333333	1		Sg3	0.2	0.2	0.2	0.6	0.2
Sum	5	1.666666667	5		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.1
F3	0.1
F5	0.6
F6	-0.1
F8	-0.5
F9	0.2
F12	0

Context-Priorization			
Context	**Rank/Relatio n**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

ImpDegree(Context)					
Feature	**C2**	**C4**	**C6**	**C8**	**C10**
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	0	0
F8	0	0	0	0	0
F9	0	0	0	0	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	**Cont**
F2	-0.5
F3	0.5
F5	0
F6	0
F8	0
F9	0
F12	0

Goal-Priorization			
Goal	**Rank/Relation**	**RankValue**	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	**Name**
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.1
F3	1.1
F4	0
F5	1.1
F6	0.4
F7	0
F8	0
F9	0.7
F10	0
F11	0
F12	0

CCF-01

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

Goals
Name
Refresh inside home
Meal suggestions
Lights control

3- Smart Home (ccf2 = c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	0.3333333333	1		**Sg1**	0.2	0.2	0.2	0.6	**0.2**
Sg2	3	**1**	3	**Normalize ->**	**Sg2**	0.6	0.6	0.6	1.8	**0.6**
Sg3	1	0.3333333333	**1**		**Sg3**	0.2	0.2	0.2	0.6	**0.2**
Sum	5	1.666666667	5		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	**Equally to moderatelly**	**Moderate preferred**	**Moderately to strongly**	**Strong preferred**	**Strongly to very strongly**	**Very strong preferred**	**Very strong to extremely**
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.1
F3	0.1
F5	0.6
F6	-0.1
F8	-0.5
F9	0.2
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	0	0
F8	0	0	0	0	0
F9	0	0	0	0	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	0
F8	0
F9	0
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	**Utility value**
F0	0
F1	0
F2	0.1
F3	1.1
F4	0
F5	1.1
F6	0.4
F7	0
F8	0
F9	0.7
F10	0
F11	0
F12	0

CCF-02

Result	**GA ID**
xf0 = 1	**f1**
xf1 = 1	**f2**
xf2 = 0	**f6**
xf3 = 1	**f7**
xf4 = 1	**f3**
xf5 = 1	**f8**
xf6 = 0	**f9**
xf7 = 1	**f4**
xf8 = 0	**f10**
xf9 = 1	**f11**
xf10 = 1	**f5**
xf11 = 1	**f12**
xf12 = 0	**f13**

4- Smart Home (ccf3 = c8)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	3	1		**Sg1**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	**1**	0.3333333333	**Normalize ->**	**Sg2**	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	**1**		**Sg3**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

CCF-03

Objective function			
Feature	Utility value	Result	GA ID
F0	0	xf0 = 1	f1
F1	0	xf1 = 1	f2
F2	0.2142857143	xf2 = 0	f6
F3	1.214285714	xf3 = 1	f7
F4	0	xf4 = 1	f3
F5	0.6428571429	xf5 = 1	f8
F6	-0.2142857143	xf6 = 0	f9
F7	0	xf7 = 1	f4
F8	-0.4285714286	xf8 = 0	f10
F9	1.357142857	xf9 = 1	f11
F10	0	xf10 = 1	f5
F11	0	xf11 = 1	f12
F12	0	xf12 = 0	f13

5- Smart Home (ccf4 = c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	3	1		**Sg1**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	**1**	0.3333333333	**Normalize ->**	**Sg2**	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	**1**		**Sg3**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Extreme preferred
9

10
1.49

Features	
ID	Name
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

Softgoals	
ID	Name
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

impDegree(Softgoal)			
Feature	Sg1	Sg2	Sg3
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	Cont
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

ImpDegree(Context)					
Feature	**C2**	**C4**	**C6**	**C8**	**C10**
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	0	-1	0
F9	0	0	0	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	**Cont**
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-0.5
F9	0.5
F12	0

Goal-Priorization			
Goal	**Rank/Relation**	**RankValue**	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Contexts	
ID	**Name**
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Feature (Con	
Feature	
F2	
F3	
F5	
F6	
F8	
F9	
F12	

ID	
G1	
G2	
G3	

e over Goals
tribution)

CCF-04

Cont	Objective function		Result	GA ID
	Feature	Utility value		
0.5	F0	0	xf0 = 1	f1
0.5	F1	0	xf1 = 1	f2
0.5	F2	0.2142857143	xf2 = 0	f6
0.5	F3	1.214285714	xf3 = 1	f7
0.5	F4	0	xf4 = 1	f3
0.5	F5	0.6428571429	xf5 = 1	f8
0	F6	-0.2142857143	xf6 = 0	f9
	F7	0	xf7 = 1	f4
	F8	0.07142857143	xf8 = 0	f10
	F9	0.8571428571	xf9 = 1	f11
	F10	0	xf10 = 1	f5
	F11	0	xf11 = 1	f12
	F12	0	xf12 = 0	f13

Goals
Name
Refresh inside home
Meal suggestions
Lights control

6- Smart Home (ccf5 = c6)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	0.3333333333	1		Sg1	0.2	0.2	0.2	0.6	0.2
Sg2	3	1	3	Normalize ->	Sg2	0.6	0.6	0.6	1.8	0.6
Sg3	1	0.3333333333	1		Sg3	0.2	0.2	0.2	0.6	0.2
Sum	5	1.666666667	5		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.1
F3	0.1
F5	0.6
F6	-0.1
F8	-0.5
F9	0.2
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	0
F3	0	0	0	0	0
F5	0	0	0	0	0
F6	0	0	0	0	0
F8	0	0	-1	0	0
F9	0	0	1	0	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	0
F3	0
F5	0
F6	0
F8	-0.5
F9	0.5
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.6
F3	0.6
F4	0
F5	1.1
F6	0.4
F7	0
F8	-0.5
F9	1.2
F10	0
F11	0
F12	0

CCF-05

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

Goals
Name
Refresh inside home
Meal suggestions
Lights control

7- Smart Home (ccf6 = c6,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	0.3333333333	1		Sg1	0.2	0.2	0.2	0.6	0.2
Sg2	3	1	3	Normalize ->	Sg2	0.6	0.6	0.6	1.8	0.6
Sg3	1	0.3333333333	1		Sg3	0.2	0.2	0.2	0.6	0.2
Sum	5	1.666666667	5		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	Name
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

Softgoals	
ID	Name
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

impDegree(Softgoal)			
Feature	Sg1	Sg2	Sg3
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	Cont
F2	0.1
F3	0.1
F5	0.6
F6	-0.1
F8	-0.5
F9	0.2
F12	0

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	0	0
F8	0	0	-1	0	0
F9	0	0	1	0	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	0
F8	-0.5
F9	0.5
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function		
Feature	Utility value	
F0	0	
F1	0	
F2	0.1	
F3	1.1	
F4	0	
F5	1.1	
F6	0.4	
F7	0	
F8	-0.5	
F9	1.2	
F10	0	
F11	0	
F12	0	

CCF-06

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

8- Smart Home (ccf7 = c6,c8)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	0.1428571429
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-07

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	3	1		**Sg1**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	**1**	0.3333333333	**Normalize ->**	**Sg2**	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	**1**		**Sg3**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

CCF-08

Objective function			
Feature	Utility value	Result	GA ID
F0	0	xf0 = 1	f1
F1	0	xf1 = 1	f2
F2	0.2142857143	xf2 = 0	f6
F3	1.214285714	xf3 = 1	f7
F4	0	xf4 = 1	f3
F5	0.6428571429	xf5 = 1	f8
F6	-0.2142857143	xf6 = 0	f9
F7	0	xf7 = 1	f4
F8	-0.4285714286	xf8 = 0	f10
F9	1.357142857	xf9 = 1	f11
F10	0	xf10 = 1	f5
F11	0	xf11 = 1	f12
F12	0	xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	0.1428571429
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Extreme preferred
9

10
1.49

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

Goals
Name
Refresh inside home
Meal suggestions
Lights control

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	3	1		**Sg1**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	**1**	0.3333333333	**Normalize ->**	**Sg2**	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	**1**		**Sg3**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Extreme preferred
9

10
1.49

Features	
ID	Name
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	Sg1	Sg2	Sg3
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	Cont
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	Name
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	3	1		**Sg1**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	**1**	0.3333333333	**Normalize ->**	**Sg2**	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	**1**		**Sg3**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Features	
ID	Name
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	Sg1	Sg2	Sg3
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	Cont
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Extreme preferred
9

10
1.49

Softgoals	
ID	Name
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10	
1.49	

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	Name
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	Sg1	Sg2	Sg3
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	Cont
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	Name
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)	Objective function		CCF-08	
Cont	Feature	Utility value	Result	GA ID
0.5	F0	0	xf0 = 1	f1
0.5	F1	0	xf1 = 1	f2
0.5	F2	0.2142857143	xf2 = 0	f6
0.5	F3	1.214285714	xf3 = 1	f7
0.5	F4	0	xf4 = 1	f3
0.5	F5	0.6428571429	xf5 = 1	f8
0	F6	-0.2142857143	xf6 = 0	f9
	F7	0	xf7 = 1	f4
	F8	-0.4285714286	xf8 = 0	f10
	F9	1.357142857	xf9 = 1	f11
	F10	0	xf10 = 1	f5
	F11	0	xf11 = 1	f12
	F12	0	xf12 = 0	f13

Goals
Name
Refresh inside home
Meal suggestions
Lights control

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Extreme preferred
9

10
1.49

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	**C2**	**C4**	**C6**	**C8**	**C10**
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	**Cont**
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	**Rank/Relation**	**RankValue**	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	**Name**
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

…e over Goals (…tribution)	Objective function		CCF-08	
Cont	Feature	Utility value	Result	GA ID
0.5	F0	0	xf0 = 1	f1
0.5	F1	0	xf1 = 1	f2
0.5	F2	0.2142857143	xf2 = 0	f6
0.5	F3	1.214285714	xf3 = 1	f7
0.5	F4	0	xf4 = 1	f3
0.5	F5	0.6428571429	xf5 = 1	f8
0	F6	-0.2142857143	xf6 = 0	f9
	F7	0	xf7 = 1	f4
	F8	-0.4285714286	xf8 = 0	f10
	F9	1.357142857	xf9 = 1	f11
	F10	0	xf10 = 1	f5
	F11	0	xf11 = 1	f12
	F12	0	xf12 = 0	f13

Goals
Name
Refresh inside home
Meal suggestions
Lights control

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	Name
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

Softgoals	
ID	Name
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

impDegree(Softgoal)			
Feature	Sg1	Sg2	Sg3
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	Cont
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	3	1		**Sg1**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	**1**	0.3333333333	**Normalize ->**	**Sg2**	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	**1**		**Sg3**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Extreme preferred
9

10	
1.49	

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

CCF-08

Objective function			
Feature	Utility value	Result	GA ID
F0	0	xf0 = 1	f1
F1	0	xf1 = 1	f2
F2	0.2142857143	xf2 = 0	f6
F3	1.214285714	xf3 = 1	f7
F4	0	xf4 = 1	f3
F5	0.6428571429	xf5 = 1	f8
F6	-0.2142857143	xf6 = 0	f9
F7	0	xf7 = 1	f4
F8	-0.4285714286	xf8 = 0	f10
F9	1.357142857	xf9 = 1	f11
F10	0	xf10 = 1	f5
F11	0	xf11 = 1	f12
F12	0	xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	Name
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	Sg1	Sg2	Sg3
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	Cont
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	Name
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1	Normalize ->	Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sg2	0.3333333333	1	0.3333333333		Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	0.1428571429
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	Name
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

Softgoals	
ID	Name
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

impDegree(Softgoal)			
Feature	Sg1	Sg2	Sg3
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	Cont
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	Rank/Relation	RankValue	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)	Objective function		CCF-08	
Cont	Feature	Utility value	Result	GA ID
0.5	F0	0	xf0 = 1	f1
0.5	F1	0	xf1 = 1	f2
0.5	F2	0.2142857143	xf2 = 0	f6
0.5	F3	1.214285714	xf3 = 1	f7
0.5	F4	0	xf4 = 1	f3
0.5	F5	0.6428571429	xf5 = 1	f8
0	F6	-0.2142857143	xf6 = 0	f9
	F7	0	xf7 = 1	f4
	F8	-0.4285714286	xf8 = 0	f10
	F9	1.357142857	xf9 = 1	f11
Goals	F10	0	xf10 = 1	f5
Name	F11	0	xf11 = 1	f12
Refresh inside home	F12	0	xf12 = 0	f13
Meal suggestions				
Lights control				

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	0.1428571429
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	**C2**	**C4**	**C6**	**C8**	**C10**
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	**Cont**
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	**Rank/Relation**	**RankValue**	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	**Name**
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Objective function	
Feature	**Utility value**
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

Goals
Name
Refresh inside home
Meal suggestions
Lights control

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	3	1		**Sg1**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	**1**	0.3333333333	**Normalize ->**	**Sg2**	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	**1**		**Sg3**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Extreme preferred
9

10	
1.49	

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1	Normalize ->	Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	1	0.3333333333		Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Extreme preferred
9

10
1.49

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	**C2**	**C4**	**C6**	**C8**	**C10**
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Contexts	
ID	**Name**
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Feature over Context (Contribution)	
Feature	**Cont**
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	**Rank/Relation**	**RankValue**	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	0.1428571429
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Objective function	
Feature	**Utility value**
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	**f1**
xf1 = 1	**f2**
xf2 = 0	**f6**
xf3 = 1	**f7**
xf4 = 1	**f3**
xf5 = 1	**f8**
xf6 = 0	**f9**
xf7 = 1	**f4**
xf8 = 0	**f10**
xf9 = 1	**f11**
xf10 = 1	**f5**
xf11 = 1	**f12**
xf12 = 0	**f13**

Goals
Name
Refresh inside home
Meal suggestions
Lights control

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	3	1		**Sg1**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	**1**	0.3333333333	**Normalize ->**	**Sg2**	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	**1**		**Sg3**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function

CCF-08

Feature	Utility value	Result	GA ID
F0	0	xf0 = 1	f1
F1	0	xf1 = 1	f2
F2	0.2142857143	xf2 = 0	f6
F3	1.214285714	xf3 = 1	f7
F4	0	xf4 = 1	f3
F5	0.6428571429	xf5 = 1	f8
F6	-0.2142857143	xf6 = 0	f9
F7	0	xf7 = 1	f4
F8	-0.4285714286	xf8 = 0	f10
F9	1.357142857	xf9 = 1	f11
F10	0	xf10 = 1	f5
F11	0	xf11 = 1	f12
F12	0	xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

Goals
Name
Refresh inside home
Meal suggestions
Lights control

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	0.1428571429
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	0.4285714286
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	1	3	1		Sg1	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	1	0.3333333333	Normalize ->	Sg2	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	1		Sg3	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		Sum	1	1	1	3	1

Number of softgoals	3

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	2	3	4	5	6	7	8

Random inconsistency indices for n (number of softgoals)	n	1	2	3	4	5	6	7	9
	RI	0	0	0.58	0.9	1.12	1.24	1.32	1.46
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	0	0

Extreme preferred
9

10	
1.49	

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

ImpDegree(Context)					
Feature	**C2**	**C4**	**C6**	**C8**	**C10**
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	**Cont**
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Goal-Priorization			
Goal	**Rank/Relation**	**RankValue**	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

Contexts	
ID	**Name**
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	3	1		**Sg1**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	**1**	0.3333333333	**Normalize ->**	**Sg2**	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	**1**		**Sg3**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Extreme preferred
9

10	
1.49	

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	**C2**	**C4**	**C6**	**C8**	**C10**
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	**Cont**
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Contexts	
ID	**Name**
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Goal-Priorization			
Goal	**Rank/Relation**	**RankValue**	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function

Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	3	1		**Sg1**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	**1**	0.3333333333	**Normalize ->**	**Sg2**	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	**1**		**Sg3**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Extreme preferred
9

10
1.49

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)
Cont
0.5
0.5
0.5
0.5
0.5
0.5
0

Goals
Name
Refresh inside home
Meal suggestions
Lights control

Objective function	
Feature	Utility value
F0	0
F1	0
F2	0.2142857143
F3	1.214285714
F4	0
F5	0.6428571429
F6	-0.2142857143
F7	0
F8	-0.4285714286
F9	1.357142857
F10	0
F11	0
F12	0

CCF-08

Result	GA ID
xf0 = 1	f1
xf1 = 1	f2
xf2 = 0	f6
xf3 = 1	f7
xf4 = 1	f3
xf5 = 1	f8
xf6 = 0	f9
xf7 = 1	f4
xf8 = 0	f10
xf9 = 1	f11
xf10 = 1	f5
xf11 = 1	f12
xf12 = 0	f13

9- Smart Home (ccf8 = c6,c8,c10)

Softgoal	Sg1	Sg2	Sg3		Softgoal	Sg1	Sg2	Sg3	Sum	ivalue
Sg1	**1**	3	1		**Sg1**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sg2	0.3333333333	**1**	0.3333333333	**Normalize ->**	**Sg2**	0.1428571429	0.1428571429	0.1428571429	0.4285714286	**0.1428571429**
Sg3	1	3	**1**		**Sg3**	0.4285714286	0.4285714286	0.4285714286	1.285714286	**0.4285714286**
Sum	2.333333333	7	2.333333333		**Sum**	1	1	1	3	1

Number of softgoals	**3**

The CI of a randomly-generated pair wise comparison matrix (%)	
Consistency index (CI)	3

Equal preferred	Equally to moderatelly	Moderate preferred	Moderately to strongly	Strong preferred	Strongly to very strongly	Very strong preferred	Very strong to extremely
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Random inconsistency indices for n (number of softgoals)	**n**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**
	RI	**0**	**0**	**0.58**	**0.9**	**1.12**	**1.24**	**1.32**	**1.46**
Random index (RI)	0.58								

Consistency ratio (CR) < 0.1 (10%)		
CR = CI/RI	**0**	**0**

Features	
ID	**Name**
F2	Open Windows
F3	Turn on air ventilator
F5	Suggest home cooked meal
F6	Suggest restaurant meal
F8	Occupancy Simulation
F9	Turn on lights in accupied rooms
F12	Warm tenant about low stock

impDegree(Softgoal)			
Feature	**Sg1**	**Sg2**	**Sg3**
F2	-0.5	0	1
F3	1	0	-0.5
F5	0	1	0
F6	-0.5	0	0
F8	0.5	-1	0
F9	-0.5	0.5	0
F12	0	0	0

Feature over NFRs (Contribution)	
Feature	**Cont**
F2	0.2142857143
F3	0.2142857143
F5	0.1428571429
F6	-0.2142857143
F8	0.07142857143
F9	-0.1428571429
F12	0

Context-Priorization			
Context	**Rank/Relation**	**RankValue**	
C1	1	0.5	
C2	OR	1	0.5
C3	1	0.5	
C4	OR	1	0.5
C5	1	0.5	
C6	OR	1	0.5
C7	1	0.5	
C8	OR	1	0.5
C9	1	0.5	
C10	OR	1	0.5

Extreme preferred
9

10
1.49

Softgoals	
ID	**Name**
SG1	Safety
SG2	Save Money
SG3	Energy Efficiency

ImpDegree(Context)					
Feature	C2	C4	C6	C8	C10
F2	0	0	0	0	-1
F3	0	0	0	0	1
F5	0	0	0	0	0
F6	0	0	0	-1	0
F8	0	0	-1	-1	0
F9	0	0	1	1	0
F12	0	0	0	0	0

Feature over Context (Contribution)	
Feature	Cont
F2	-0.5
F3	0.5
F5	0
F6	-0.5
F8	-1
F9	1
F12	0

Contexts	
ID	Name
C1	Battery Level
C2	Battery < 30%
C3	Battery >= 30%
C4	Network Available
C5	WiFi connected

Goal-Priorization			
Goal	Rank/Relation	RankValue	
G1	1	0.5	
Hg1 (F2)	OR	1	0.5
Hg2 (F3)	OR	1	0.5
G2	1	0.5	
Hg3 (F5)	OR	1	0.5
Hg4 (F6)	OR	1	0.5
G3	1	0.5	
Hg5 (F8)	OR	1	0.5
Hg6 (F9)	OR	1	0.5

Feature (Con
Feature
F2
F3
F5
F6
F8
F9
F12

ID
G1
G2
G3

e over Goals tribution)	Objective function		CCF-08	
Cont	Feature	Utility value	Result	GA ID
0.5	F0	0	xf0 = 1	f1
0.5	F1	0	xf1 = 1	f2
0.5	F2	0.2142857143	xf2 = 0	f6
0.5	F3	1.214285714	xf3 = 1	f7
0.5	F4	0	xf4 = 1	f3
0.5	F5	0.6428571429	xf5 = 1	f8
0	F6	-0.2142857143	xf6 = 0	f9
	F7	0	xf7 = 1	f4
	F8	-0.4285714286	xf8 = 0	f10
	F9	1.357142857	xf9 = 1	f11
	F10	0	xf10 = 1	f5
	F11	0	xf11 = 1	f12
	F12	0	xf12 = 0	f13

Goals
Name
Refresh inside home
Meal suggestions
Lights control

CCF 1 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	-0.5	0.5	0.1
xf3	xf7	xf6	0.5	0.5	0.1
xf5	xf8	xf7	0	0.5	0.6
xf6	xf9	xf8	0	0.5	-0.1
xf8	xf10	xf9	0	0.5	-0.5
xf9	xf11	xf10	0	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 2 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	-0.5	0.5	0.1
xf3	xf7	xf6	0.5	0.5	0.1
xf5	xf8	xf7	0	0.5	0.6
xf6	xf9	xf8	0	0.5	-0.1
xf8	xf10	xf9	0	0.5	-0.5
xf9	xf11	xf10	0	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 3 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	-0.5	0.5	0.2142857143
xf3	xf7	xf6	0.5	0.5	0.2142857143
xf5	xf8	xf7	0	0.5	0.1428571429
xf6	xf9	xf8	-0.5	0.5	-0.2142857143
xf8	xf10	xf9	-1	0.5	0.07142857143
xf9	xf11	xf10	1	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 4 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	-0.5	0.5	0.2142857143
xf3	xf7	xf6	0.5	0.5	0.2142857143
xf5	xf8	xf7	0	0.5	0.1428571429
xf6	xf9	xf8	-0.5	0.5	-0.2142857143
xf8	xf10	xf9	-0.5	0.5	0.07142857143
xf9	xf11	xf10	0.5	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 5 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.1
xf3	xf7	xf6	0	0.5	0.1
xf5	xf8	xf7	0	0.5	0.6
xf6	xf9	xf8	0	0.5	-0.1
xf8	xf10	xf9	-0.5	0.5	-0.5
xf9	xf11	xf10	0.5	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 6 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	-0.5	0.5	0.1
xf3	xf7	xf6	0.5	0.5	0.1
xf5	xf8	xf7	0	0.5	0.6
xf6	xf9	xf8	0	0.5	-0.1
xf8	xf10	xf9	-0.5	0.5	-0.5
xf9	xf11	xf10	0.5	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 1 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf14	xf13	0	0	0
xf1	xf15	xf14	0	0	0
xf4	xf16	xf15	0	0	0
xf7	xf17	xf16	0	0	0
xf10	xf18	xf17	0	0	0
xf2	xf19	xf18			
xf3	xf20	xf19		Name	
xf5	xf21	xf20		Refresh inside home	
xf6	xf22	xf21		Meal suggestions	
xf8	xf23	xf22		Lights control	
xf9	xf24	xf23			
xf11	xf25	xf24	1	1	1
xf12	xf26	xf25			

CCF 2 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf14	xf13	0	0	0
xf1	xf15	xf14	0	0	0
xf4	xf16	xf15	0	0	0
xf7	xf17	xf16	0	0	0
xf10	xf18	xf17	0	0	0
xf2	xf19	xf18			
xf3	xf20	xf19		Name	
xf5	xf21	xf20		Refresh inside home	
xf6	xf22	xf21		Meal suggestions	
xf8	xf23	xf22		Lights control	
xf9	xf24	xf23			
xf11	xf25	xf24	1	1	1
xf12	xf26	xf25			

CCF 9 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.1
xf3	xf7	xf6	0	0.5	0.1
xf5	xf8	xf7	-0.5	0.5	0.6
xf6	xf9	xf8	0.5	0.5	-0.1
xf8	xf10	xf9	0	0.5	-0.5
xf9	xf11	xf10	0	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 10 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	-0.5	0.5	0.1
xf3	xf7	xf6	0.5	0.5	0.1
xf5	xf8	xf7	-0.5	0.5	0.6
xf6	xf9	xf8	0.5	0.5	-0.1
xf8	xf10	xf9	0	0.5	-0.5
xf9	xf11	xf10	0	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 11 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.2142857143
xf3	xf7	xf6	0	0.5	0.2142857143
xf5	xf8	xf7	-0.5	0.5	0.1428571429
xf6	xf9	xf8	0	0.5	-0.2142857143
xf8	xf10	xf9	-0.5	0.5	0.07142857143
xf9	xf11	xf10	0.5	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 12 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	-0.5	0.5	0.2142857143
xf3	xf7	xf6	0.5	0.5	0.2142857143
xf5	xf8	xf7	-0.5	0.5	0.1428571429
xf6	xf9	xf8	0	0.5	-0.2142857143
xf8	xf10	xf9	-0.5	0.5	0.07142857143
xf9	xf11	xf10	0.5	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 13 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.1
xf3	xf7	xf6	0	0.5	0.1
xf5	xf8	xf7	-0.5	0.5	0.6
xf6	xf9	xf8	0.5	0.5	-0.1
xf8	xf10	xf9	-0.5	0.5	-0.5
xf9	xf11	xf10	0.5	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 14 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	-0.5	0.5	0.1
xf3	xf7	xf6	0.5	0.5	0.1
xf5	xf8	xf7	-0.5	0.5	0.6
xf6	xf9	xf8	0.5	0.5	-0.1
xf8	xf10	xf9	-0.5	0.5	-0.5
xf9	xf11	xf10	0.5	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 15 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.2142857143
xf3	xf7	xf6	0	0.5	0.2142857143
xf5	xf8	xf7	-0.5	0.5	0.1428571429
xf6	xf9	xf8	0	0.5	-0.2142857143
xf8	xf10	xf9	-1	0.5	0.07142857143
xf9	xf11	xf10	1	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 16 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	-0.5	0.5	0.2142857143
xf3	xf7	xf6	0.5	0.5	0.2142857143
xf5	xf8	xf7	-0.5	0.5	0.1428571429
xf6	xf9	xf8	0	0.5	-0.2142857143
xf8	xf10	xf9	-1	0.5	0.07142857143
xf9	xf11	xf10	1	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 17 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0.5	0.5	0.1
xf3	xf7	xf6	-0.5	0.5	0.1
xf5	xf8	xf7	0	0.5	0.6
xf6	xf9	xf8	0	0.5	-0.1
xf8	xf10	xf9	0	0.5	-0.5
xf9	xf11	xf10	0	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 18 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.1
xf3	xf7	xf6	0	0.5	0.1
xf5	xf8	xf7	0	0.5	0.6
xf6	xf9	xf8	0	0.5	-0.1
xf8	xf10	xf9	0	0.5	-0.5
xf9	xf11	xf10	0	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 19 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0.5	0.5	0.2142857143
xf3	xf7	xf6	-0.5	0.5	0.2142857143
xf5	xf8	xf7	0	0.5	0.1428571429
xf6	xf9	xf8	-0.5	0.5	-0.2142857143
xf8	xf10	xf9	-0.5	0.5	0.07142857143
xf9	xf11	xf10	0.5	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 20 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.2142857143
xf3	xf7	xf6	0	0.5	0.2142857143
xf5	xf8	xf7	0	0.5	0.1428571429
xf6	xf9	xf8	-0.5	0.5	-0.2142857143
xf8	xf10	xf9	-0.5	0.5	0.07142857143
xf9	xf11	xf10	0.5	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 21 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0.5	0.5	0.1
xf3	xf7	xf6	-0.5	0.5	0.1
xf5	xf8	xf7	0	0.5	0.6
xf6	xf9	xf8	0	0.5	-0.1
xf8	xf10	xf9	-0.5	0.5	-0.5
xf9	xf11	xf10	0.5	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 22 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.1
xf3	xf7	xf6	0	0.5	0.1
xf5	xf8	xf7	0	0.5	0.6
xf6	xf9	xf8	0	0.5	-0.1
xf8	xf10	xf9	-0.5	0.5	-0.5
xf9	xf11	xf10	0.5	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 23 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0.5	0.5	0.2142857143
xf3	xf7	xf6	-0.5	0.5	0.2142857143
xf5	xf8	xf7	0	0.5	0.1428571429
xf6	xf9	xf8	-0.5	0.5	-0.2142857143
xf8	xf10	xf9	-1	0.5	0.07142857143
xf9	xf11	xf10	1	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 24 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.2142857143
xf3	xf7	xf6	0	0.5	0.2142857143
xf5	xf8	xf7	0	0.5	0.1428571429
xf6	xf9	xf8	-0.5	0.5	-0.2142857143
xf8	xf10	xf9	-1	0.5	0.07142857143
xf9	xf11	xf10	1	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0	0	0

CCF 25 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0.5	0.5	0.1
xf3	xf7	xf6	-0.5	0.5	0.1
xf5	xf8	xf7	-0.5	0.5	0.6
xf6	xf9	xf8	0.5	0.5	-0.1
xf8	xf10	xf9	0	0.5	-0.5
xf9	xf11	xf10	0	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 26 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.1
xf3	xf7	xf6	0	0.5	0.1
xf5	xf8	xf7	-0.5	0.5	0.6
xf6	xf9	xf8	0.5	0.5	-0.1
xf8	xf10	xf9	0	0.5	-0.5
xf9	xf11	xf10	0	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 27 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0.5	0.5	0.2142857143
xf3	xf7	xf6	-0.5	0.5	0.2142857143
xf5	xf8	xf7	-0.5	0.5	0.1428571429
xf6	xf9	xf8	0	0.5	-0.2142857143
xf8	xf10	xf9	-0.5	0.5	0.07142857143
xf9	xf11	xf10	0.5	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 28 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.2142857143
xf3	xf7	xf6	0	0.5	0.2142857143
xf5	xf8	xf7	-0.5	0.5	0.1428571429
xf6	xf9	xf8	0	0.5	-0.2142857143
xf8	xf10	xf9	-0.5	0.5	0.07142857143
xf9	xf11	xf10	0.5	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 29 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0.5	0.5	0.1
xf3	xf7	xf6	-0.5	0.5	0.1
xf5	xf8	xf7	-0.5	0.5	0.6
xf6	xf9	xf8	0.5	0.5	-0.1
xf8	xf10	xf9	-0.5	0.5	-0.5
xf9	xf11	xf10	0.5	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 30 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.1
xf3	xf7	xf6	0	0.5	0.1
xf5	xf8	xf7	-0.5	0.5	0.6
xf6	xf9	xf8	0.5	0.5	-0.1
xf8	xf10	xf9	-0.5	0.5	-0.5
xf9	xf11	xf10	0.5	0.5	0.2
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 31 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0.5	0.5	0.2142857143
xf3	xf7	xf6	-0.5	0.5	0.2142857143
xf5	xf8	xf7	-0.5	0.5	0.1428571429
xf6	xf9	xf8	0	0.5	-0.2142857143
xf8	xf10	xf9	-1	0.5	0.07142857143
xf9	xf11	xf10	1	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

CCF 32 - SmartHome					
ToffA-DAS	ToffA-DAS+				
Feature ID	Feature ID	Feature ID	Utiliy values - Contexts	Utility values - Goals	Utility values - Soft goals
xf0	xf1	xf0	0	0	0
xf1	xf2	xf1	0	0	0
xf4	xf3	xf2	0	0	0
xf7	xf4	xf3	0	0	0
xf10	xf5	xf4	0	0	0
xf2	xf6	xf5	0	0.5	0.2142857143
xf3	xf7	xf6	0	0.5	0.2142857143
xf5	xf8	xf7	-0.5	0.5	0.1428571429
xf6	xf9	xf8	0	0.5	-0.2142857143
xf8	xf10	xf9	-1	0.5	0.07142857143
xf9	xf11	xf10	1	0.5	-0.1428571429
xf11	xf12	xf11	0	0	0
xf12	xf13	xf12	0.5	0	0

ILP x GA					Fitness values		
Scenario	ConG4DaS	ToffA-DAS	Fitness (Context + Soft goal + Goal)	ToffA-DAS+	Fitness (Context)	Fitness (Soft goal)	Fitness (Goal)
ccf1	f3,f5,f8	f3,f5,f9	2.5	f2,f6,f8	-0.5	-0.5	1.5
ccf2	f3,f5,f8	f3,f5,f9	2.5	f2,f6,f8	-0.5	-0.5	1.5
ccf3	f3,f5,f9	f3,f5,f9	3.5	f2,f6,f9,f12	0	-0.1428571429	1.5
ccf4	f3,f5,f9	f3,f6,f9	3	f2,f6,f8	-1.5	0.07142857143	1.5
ccf5	f3,f5,f9	f3,f5,f9	2.5	f2,f6,f8	-0.5	-0.5	1.5
ccf6	f3,f5,f9	f3,f5,f9	3	f2,f6,f8	-1	-0.5	1.5
ccf7	f3,f5,f9	f3,f5,f9	3.5	f2,f6,f8	-2	0.07142857143	1.5
ccf8	f3,f5,f9	f3,f5,f9	3.5	f2,f6,f8	-2	0.07142857143	1.5
ccf9	f3,f6,f8,f12	f3,f6,f9,f12	2.7	f2,f6,f8	1	-0.5	1.5
ccf10	f3,f6,f8,f12	f3,f6,f9,f12	3.2	f2,f5,f8,f12	-0.5	0.2	1.5
ccf11	f3,f6,f9,f12	f2,f5,f9,f12	2.5	f3,f5,f8	-1	0.4285714286	1.5
ccf12	f3,f6,f9,f12	f3,f5,f9,f12	3	f3,f6,f9	1	-0.1428571429	1.5
ccf13	f3,f6,f9,f12	f2,f6,f9,f12	3.2	f2,f5,f8	-1	0.2	1.5
ccf14	f3,f6,f9,f12	f3,f6,f9,f12	3.7	f2,f5,f8	-1.5	0.2	1.5
ccf15	f3,f6,f9,f12	f2,f6,f9	2.5	f3,f5,f8	-1.5	0.4285714286	1.5
ccf16	f3,f6,f9,f12	f3,f5,f9	3	f2,f5,f8	-2	0.4285714286	1.5
ccf17	f2,f5,f8	f2,f6,f9	2.5	f3,f6,f8,f12	-0.5	-0.5	1.5
ccf18	f3,f5,f8	f3,f5,f9	2	f3,f6,f8	0	-0.5	1.5
ccf19	f2,f5,f9	f2,f5,f9	3	f3,f6,f8,f12	-1.5	0.07142857143	1.5
ccf20	f2,f5,f9	f2,f5,f9	2.5	f2,f6,f8	-1	0.07142857143	1.5
ccf21	f2,f5,f9	f2,f5,f9	3	f3,f5,f8	-1	0.2	1.5
ccf22	f3,f5,f9	f3,f5,f9	2.5	f3,f6,f8,f12	-0.5	-0.5	1.5
ccf23	f2,f5,f9	f2,f5,f9	3.5	f3,f6,f8,f12	-2	0.07142857143	1.5
ccf24	f2,f5,f9	f2,f5,f9	3	f2,f6,f8	-1.5	0.07142857143	1.5
ccf25	f2,f6,f8,f12	f2,f6,f9,f12	3.2	f3,f5,f8	-1	0.2	1.5
ccf26	f3,f5,f8,f12	f3,f6,f9,f12	2.7	f3,f6,f8	0.5	-0.5	1.5
ccf27	f2,f6,f9,f12	f2,f6,f9,f12	3	f3,f6,f9	-0.5	-0.1428571429	1.5
ccf28	f2,f6,f9,f12	f2,f6,f9,f12	2.5	f2,f6,f8	-0.5	0.07142857143	1.5
ccf29	f2,f6,f9,f12	f2,f6,f9,f12	3.7	f2,f6,f8	1	-0.5	1.5
ccf30	f3,f6,f9,f12	f3,f6,f9,f12	3.2	f2,f6,f8	0.5	-0.5	1.5
ccf31	f2,f6,f9	f2,f5,f9,f12	3.5	f3,f5,f8	-2	0.4285714286	1.5
ccf32	f2,f6,f9,f12	f2,f6,f9,f12	3	f3,f5,f9	-1.5	0.4285714286	1.5

Configuration	Pos CON	Pos CCA	Difference	Sign	Absolute value	Signed Rank
1	2	1	1	1	1	25.5
2	2	1	1	1	1	25.5
3	2	2	0	0	0	0
4	2	2	0	0	0	0
5	1	1	0	0	0	0
6	1	1	0	0	0	0
7	2	2	0	0	0	0
8	2	2	0	0	0	0
9	2	1	1	1	1	25.5
10	2	1	1	1	1	25.5
11	1	2	-1	-1	1	-25.5
12	1	2	-1	-1	1	-25.5
13	1	1	0	0	0	0
14	1	1	0	0	0	0
15	1	1	0	0	0	0
16	1	2	-1	-1	1	-25.5
17	1	0	1	1	1	25.5
18	2	1	1	1	1	25.5
19	3	3	0	0	0	0
20	3	2	1	1	1	25.5
21	0	1	-1	-1	1	-25.5
22	1	1	0	0	0	0
23	3	3	0	0	0	0
24	3	3	0	0	0	0
25	1	0	1	1	1	25.5
26	2	1	1	1	1	25.5
27	2	2	0	0	0	0
28	2	2	0	0	0	0
29	0	0	0	0	0	0
30	1	2	-1	-1	1	-25.5
31	2	2	0	0	0	0
32	2	2	0	0	0	0
Sum	52	48				
Median	2	1.5				

229.5	*Positive Sum	
-127.5	*Negative Sum	
127.5	*Test Statistic (Ws)	[Ws > Wc]
21	*Critical Value (Wc)	We reject H0. There is difference between ToffA-DAS and ToffA-DAS+ in terms of POS.
14	*Sample size	

H0: **There is no difference beteween releases**
H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**
There is sufficient evidence to suggest that there is difference between the ToffA-DAS and ConG4Das in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we reject H0 | We reject H0 when Test Stat > Critical Value **[Ws > Wc]**
There is sufficient evidence to suggest that there is difference between the ToffA-DAS and ConG4Das in terms of configurations by considering satisfaction levels.

Two-Tailed Test
A two-tailed test has two critical values, one on each side of the distribution, which is often assumed to be symmetrical (e.g. Gaussian and Student-t distributions.).
When using a two-tailed test, a significance level (or alpha) used in the calculation of the critical values must be divided by 2. The critical value will then use a portion of this alpha on each side of the distribution.
To make this concrete, consider an alpha of 5%. This would be split to give two alpha values of 2.5% on either side of the distribution with an acceptance area in the middle of the distribution of 95%.
We can refer to each critical value as the lower and upper critical values for the left and right of the distribution respectively. Test statistic values more than or equal to the lower critical value and less than or equal to the upper critical value indicate the failure to reject the null hypothesis. Whereas test statistic values less than the lower critical value and more than the upper critical value indicates rejection of the null hypothesis for the test.
We can summarize this interpretation as follows:
Lower CR <= Test Statistic <= Upper CR: Failure to reject the null hypothesis of the statistical test.
Test Statistic < Lower CR OR Test Statistic > Upper CR: Reject the null hypothesis of the statistical test.
If the distribution of the test statistic is symmetric around a mean of zero, then we can shortcut the check by comparing the absolute (positive) value of the test statistic to the upper critical value.
|Test Statistic| <= Upper Critical Value: Failure to reject the null hypothesis of the statistical test.
Where |Test Statistic| is the absolute value of the calculated test statistic.

Configuration	Neg CON	Neg ToffA	Difference	Sign	Absolute value	Signed Rank
1	0	1	-1	-1	1	-25
2	0	1	-1	-1	1	-25
3	1	1	0	0	0	0
4	1	1	0	0	0	0
5	1	1	0	0	0	0
6	1	1	0	0	0	0
7	1	1	0	0	0	0
8	1	1	0	0	0	0
9	0	1	-1	-1	1	-25
10	0	1	-1	-1	1	-25
11	2	1	1	1	1	25
12	2	1	1	1	1	25
13	1	1	0	0	0	0
14	1	1	0	0	0	0
15	2	2	0	0	0	0
16	2	1	1	1	1	25
17	1	2	-1	-1	1	-25
18	0	1	-1	-1	1	-25
19	0	0	0	0	0	0
20	0	1	-1	-1	1	-25
21	2	1	1	1	1	25
22	1	1	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	2	-2	-1	2	-32
26	0	1	-1	-1	1	-25
27	1	1	0	0	0	0
28	1	1	0	0	0	0
29	2	2	0	0	0	0
30	1	0	1	1	1	25
31	1	1	0	0	0	0
32	1	1	0	0	0	0
Sum	27	32				
Median	1	1				

125	*Positive Sum	
-232	*Negative Sum	
125	*Test Statistic (Ws)	[Ws > Wc]
21	*Critical Value (Wc)	We reject H0. There is difference between ToffA-DAS and ConG4Das in terms of POS.
14	*Sample size	

H0: **There is no difference between releases**
H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**
There is sufficient evidence to suggest that there is difference between the ToffA-DAS and ConG4Das in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we reject H0 | We reject H0 when Test Stat > Critical Value **[Ws > Wc]**
There is sufficient evidence to suggest that there is difference between the ToffA-DAS and ConG4Das in terms of configurations by considering satisfaction levels.

Two-Tailed Test
A two-tailed test has two critical values, one on each side of the distribution, which is often assumed to be symmetrical (e.g. Gaussian and Student-t distributions.).
When using a two-tailed test, a significance level (or alpha) used in the calculation of the critical values must be divided by 2. The critical value will then use a portion of this alpha on each side of the distribution.
To make this concrete, consider an alpha of 5%. This would be split to give two alpha values of 2.5% on either side of the distribution with an acceptance area in the middle of the distribution of 95%.
We can refer to each critical value as the lower and upper critical values for the left and right of the distribution respectively. Test statistic values more than or equal to the lower critical value and less than or equal to the upper critical value indicate the failure to reject the null hypothesis. Whereas test statistic values less than the lower critical value and more than the upper critical value indicates rejection of the null hypothesis for the test.
We can summarize this interpretation as follows:
Lower CR <= Test Statistic <= Upper CR: Failure to reject the null hypothesis of the statistical test.
Test Statistic < Lower CR OR Test Statistic > Upper CR: Reject the null hypothesis of the statistical test.
If the distribution of the test statistic is symmetric around a mean of zero, then we can shortcut the check by comparing the absolute (positive) value of the test statistic to the upper critical value.
|Test Statistic| <= Upper Critical Value: Failure to reject the null hypothesis of the statistical test.
Where |Test Statistic| is the absolute value of the calculated test statistic.

Configuration	Diff CON	Diff ToffA	Difference	Sign	Absolute value	Signed Rank
1	2	0	2	1	2	25
2	2	0	2	1	2	25
3	1	1	0	0	0	0
4	1	1	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	1	1	0	0	0	0
8	1	1	0	0	0	0
9	2	0	2	1	2	25
10	2	0	2	1	2	25
11	-1	1	-2	-1	2	-25
12	-1	1	-2	-1	2	-25
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	-1	-1	0	0	0	0
16	-1	1	-2	-1	2	-25
17	0	-2	2	1	2	25
18	2	0	2	1	2	25
19	3	3	0	0	0	0
20	3	1	2	1	2	25
21	-2	0	-2	-1	2	-25
22	0	0	0	0	0	0
23	3	3	0	0	0	0
24	3	3	0	0	0	0
25	1	-2	3	1	3	32
26	2	0	2	1	2	25
27	1	1	0	0	0	0
28	1	1	0	0	0	0
29	-2	-2	0	0	0	0
30	0	2	-2	-1	2	-25
31	1	1	0	0	0	0
32	1	1	0	0	0	0

SUM	25	16
Median	1	0.5

232	*Positive Sum	
-125	*Negative Sum	
125	*Test Statistic (Ws)	[Ws > Wc]
21	*Critical Value (Wc)	We reject H0. There is difference between ToffA-DAS and ConG4Das in terms of POS.
14	*Sample size	

H0: **There is no difference beteween releases**
H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**
There is sufficient evidence to suggest that there is difference between the ToffA-DAS and ConG4Das in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we reject H0 | We reject H0 when Test Stat > Critical Value **[Ws > Wc]**
There is sufficient evidence to suggest that there is difference between the ToffA-DAS and ConG4Das in terms of configurations by considering satisfaction levels.

Two-Tailed Test
A two-tailed test has two critical values, one on each side of the distribution, which is often assumed to be symmetrical (e.g. Gaussian and Student-t distributions.).
When using a two-tailed test, a significance level (or alpha) used in the calculation of the critical values must be divided by 2. The critical value will then use a portion of this alpha on each side of the distribution.
To make this concrete, consider an alpha of 5%. This would be split to give two alpha values of 2.5% on either side of the distribution with an acceptance area in the middle of the distribution of 95%.
We can refer to each critical value as the lower and upper critical values for the left and right of the distribution respectively. Test statistic values more than or equal to the lower critical value and less than or equal to the upper critical value indicate the failure to reject the null hypothesis. Whereas test statistic values less than the lower critical value and more than the upper critical value indicates rejection of the null hypothesis for the test.
We can summarize this interpretation as follows:
Lower CR <= Test Statistic <= Upper CR: Failure to reject the null hypothesis of the statistical test.
Test Statistic < Lower CR OR Test Statistic > Upper CR: Reject the null hypothesis of the statistical test.
If the distribution of the test statistic is symmetric around a mean of zero, then we can shortcut the check by comparing the absolute (positive) value of the test statistic to the upper critical value.
|Test Statistic| <= Upper Critical Value: Failure to reject the null hypothesis of the statistical test.
Where |Test Statistic| is the absolute value of the calculated test statistic.

Configuration	Pos ToffA	Pos ToffA+	Difference	Sign	Absolute value	Signed Rank
1	5	3	2	1	2	21
2	5	3	2	1	2	21
3	5	3	2	1	2	21
4	5	3	2	1	2	21
5	5	3	2	1	2	21
6	5	3	2	1	2	21
7	5	3	2	1	2	21
8	5	3	2	1	2	21
9	3	3	0	0	0	0
10	3	5	-2	-1	2	-21
11	3	5	-2	-1	2	-21
12	5	3	2	1	2	21
13	3	5	-2	-1	2	-21
14	3	5	-2	-1	2	-21
15	3	5	-2	-1	2	-21
16	5	5	0	0	0	0
17	3	5	-2	-1	2	-21
18	5	3	2	1	2	21
19	5	3	2	1	2	21
20	5	3	2	1	2	21
21	5	5	0	0	0	0
22	5	3	2	1	2	21
23	5	3	2	1	2	21
24	5	3	2	1	2	21
25	3	5	-2	-1	2	-21
26	3	3	0	0	0	0
27	3	3	0	0	0	0
28	3	3	0	0	0	0
29	3	3	0	0	0	0
30	3	3	0	0	0	0
31	3	5	-2	-1	2	-21
32	3	4	-1	-1	1	-9
Sum	130	117				
Median	5	3				

315	*Positive Sum	
-177	*Negative Sum	
177	*Test Statistic (Ws)	[Ws > Wc]
81	*Critical Value (Wc)	We reject H0. There is difference between ToffA-DAS and ToffA-DAS+ in terms of POS.
24	*Sample size	

H0: **There is no difference beteween releases**

H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**

There is sufficient evidence to suggest that there is difference between the ToffA-DAS and ToffA-DAS+ in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we reject H0 | We reject H0 when Test Stat > Critical Value **[Ws > Wc]**

There is sufficient evidence to suggest that there is difference between the ToffA-DAS and ToffA-DAS+ in terms of configurations by considering satisfaction levels.

Two-Tailed Test
A two-tailed test has two critical values, one on each side of the distribution, which is often assumed to be symmetrical (e.g. Gaussian and Student-t distributions.).
When using a two-tailed test, a significance level (or alpha) used in the calculation of the critical values must be divided by 2. The critical value will then use a portion of this alpha on each side of the distribution.
To make this concrete, consider an alpha of 5%. This would be split to give two alpha values of 2.5% on either side of the distribution with an acceptance area in the middle of the distribution of 95%.
We can refer to each critical value as the lower and upper critical values for the left and right of the distribution respectively. Test statistic values more than or equal to the lower critical value and less than or equal to the upper critical value indicate the failure to reject the null hypothesis. Whereas test statistic values less than the lower critical value and more than the upper critical value indicates rejection of the null hypothesis for the test.
We can summarize this interpretation as follows:
Lower CR <= Test Statistic <= Upper CR: Failure to reject the null hypothesis of the statistical test.
Test Statistic < Lower CR OR Test Statistic > Upper CR: Reject the null hypothesis of the statistical test.
If the distribution of the test statistic is symmetric around a mean of zero, then we can shortcut the check by comparing the absolute (positive) value of the test statistic to the upper critical value.
|Test Statistic| <= Upper Critical Value: Failure to reject the null hypothesis of the statistical test.
Where |Test Statistic| is the absolute value of the calculated test statistic.

Configuration	Neg ToffA	Neg ToffA+	Difference	Sign	Absolute value	Signed Rank
1	3	3	0	0	0	0
2	3	3	0	0	0	0
3	3	4	-1	-1	1	-20
4	3	3	0	0	0	0
5	3	3	0	0	0	0
6	3	3	0	0	0	0
7	3	3	0	0	0	0
8	3	3	0	0	0	0
9	4	3	1	1	1	20
10	4	2	2	1	2	29
11	4	2	2	1	2	29
12	3	4	-1	-1	1	-20
13	4	2	2	1	2	29
14	4	2	2	1	2	29
15	4	2	2	1	2	29
16	3	2	1	1	1	20
17	4	3	1	1	1	20
18	3	3	0	0	0	0
19	3	3	0	0	0	0
20	3	3	0	0	0	0
21	3	2	1	1	1	20
22	3	3	0	0	0	0
23	3	3	0	0	0	0
24	3	3	0	0	0	0
25	4	2	2	1	2	29
26	4	3	1	1	1	20
27	4	4	0	0	0	0
28	4	3	1	1	1	20
29	4	3	1	1	1	20
30	4	3	1	1	1	20
31	4	2	2	1	2	29
32	4	3	1	1	1	20

Sum	111	90
Median	3	3

383	*Positive Sum	
-40	*Negative Sum	
40	*Test Statistic (Ws)	[Ws = Wc]
40	*Critical Value (Wc)	We do not reject H0. There is no difference between ToffA-DAS and ToffA-DAS+ in terms of POS.
18	*Sample size	

H0: **There is no difference beteween releases**
H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**
There is sufficient evidence to suggest that there is difference between the ToffA-DAS and ToffA-DAS+ in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we reject H0 | We reject H0 when Test Stat > Critical Value **[Ws > Wc]**
There is sufficient evidence to suggest that there is difference between the ToffA-DAS and ToffA-DAS+ in terms of configurations by considering satisfaction levels.

Two-Tailed Test
A two-tailed test has two critical values, one on each side of the distribution, which is often assumed to be symmetrical (e.g. Gaussian and Student-t distributions.).
When using a two-tailed test, a significance level (or alpha) used in the calculation of the critical values must be divided by 2. The critical value will then use a portion of this alpha on each side of the distribution.
To make this concrete, consider an alpha of 5%. This would be split to give two alpha values of 2.5% on either side of the distribution with an acceptance area in the middle of the distribution of 95%.
We can refer to each critical value as the lower and upper critical values for the left and right of the distribution respectively. Test statistic values more than or equal to the lower critical value and less than or equal to the upper critical value indicate the failure to reject the null hypothesis. Whereas test statistic values less than the lower critical value and more than the upper critical value indicates rejection of the null hypothesis for the test.
We can summarize this interpretation as follows:
Lower CR <= Test Statistic <= Upper CR: Failure to reject the null hypothesis of the statistical test.
Test Statistic < Lower CR OR Test Statistic > Upper CR: Reject the null hypothesis of the statistical test.
If the distribution of the test statistic is symmetric around a mean of zero, then we can shortcut the check by comparing the absolute (positive) value of the test statistic to the upper critical value.
|Test Statistic| <= Upper Critical Value: Failure to reject the null hypothesis of the statistical test.
Where |Test Statistic| is the absolute value of the calculated test statistic.

Configuration	Diff ToffA	Diff ToffA+	Difference	Sign	Absolute value	Signed Rank
1	2	0	2	1	2	17
2	2	-1	3	1	3	25
3	2	-1	3	1	3	25
4	2	0	2	1	2	17
5	2	0	2	1	2	17
6	2	0	2	1	2	17
7	2	0	2	1	2	17
8	2	0	2	1	2	17
9	-1	0	-1	-1	1	-6
10	-1	0	-1	-1	1	-6
11	-1	3	-4	-1	4	-29.5
12	2	-1	3	1	3	25
13	-1	3	-4	-1	4	-29.5
14	-1	3	-4	-1	4	-29.5
15	-1	3	-4	-1	4	-29.5
16	2	3	-1	-1	1	-6
17	-1	0	-1	-1	1	-6
18	2	0	2	1	2	17
19	2	0	2	1	2	17
20	2	0	2	1	2	17
21	2	3	-1	-1	1	-6
22	2	0	2	1	2	17
23	2	0	2	1	2	17
24	2	0	2	1	2	17
25	-1	3	-4	-1	4	-29.5
26	-1	0	-1	-1	1	-6
27	-1	-1	0	0	0	0
28	-1	0	-1	-1	1	-6
29	-1	0	-1	-1	1	-6
30	-1	0	-1	-1	1	-6
31	-1	3	-4	-1	4	-29.5
32	-1	1	-2	-1	2	-17
SUM	19	21				
Median	2	0				

279	*Positive Sum	
-248	*Negative Sum	
248	*Test Statistic (Ws)	[Ws > Wc]
147	*Critical Value (Wc)	We reject H0. There is difference between ToffA-DAS and ToffA-DAS+ in terms of POS.
31	*Sample size	

H0: **There is no difference beteween releases**

H1: **There is a difference** (the median change was non-zero)

If the **Test stat** is less than the **Critical Value**, we reject H0 | We reject H0 when Test Stat < Critical Value **[Ws < Wc]**

There is sufficient evidence to suggest that there is difference between the ToffA-DAS and ToffA-DAS+ in terms of configurations by considering satisfaction levels.

If the **Test stat** is highest than the **Critical Value**, we reject H0 | We reject H0 when Test Stat > Critical Value **[Ws > Wc]**

There is not sufficient evidence to suggest that there is difference between the ToffA-DAS and ToffA-DAS+ in terms of configurations by considering satisfaction levels.

Two-Tailed Test
A two-tailed test has two critical values, one on each side of the distribution, which is often assumed to be symmetrical (e.g. Gaussian and Student-t distributions.).
When using a two-tailed test, a significance level (or alpha) used in the calculation of the critical values must be divided by 2. The critical value will then use a portion of this alpha on each side of the distribution.
To make this concrete, consider an alpha of 5%. This would be split to give two alpha values of 2.5% on either side of the distribution with an acceptance area in the middle of the distribution of 95%.
We can refer to each critical value as the lower and upper critical values for the left and right of the distribution respectively. Test statistic values more than or equal to the lower critical value and less than or equal to the upper critical value indicate the failure to reject the null hypothesis. Whereas test statistic values less than the lower critical value and more than the upper critical value indicates rejection of the null hypothesis for the test.
We can summarize this interpretation as follows:
Lower CR <= Test Statistic <= Upper CR: Failure to reject the null hypothesis of the statistical test.
Test Statistic < Lower CR OR Test Statistic > Upper CR: Reject the null hypothesis of the statistical test.
If the distribution of the test statistic is symmetric around a mean of zero, then we can shortcut the check by comparing the absolute (positive) value of the test statistic to the upper critical value.
|Test Statistic| <= Upper Critical Value: Failure to reject the null hypothesis of the statistical test.
Where |Test Statistic| is the absolute value of the calculated test statistic.