Cátia Isabel Martins Moreira

Licenciada em Engenharia Informática e de Computadores

A DSL for querying Building Data Streams of Energy, Weather and Occupation Data

Dissertação para obtenção do Grau de Mestrado em
Engenharia Informática

Orientador: Vasco Amaral, Prof. Auxiliar, Universidade Nova de Lisboa

Co-orientador: Paulo Carreira, Prof. Auxiliar, Instituto Superior Técnico

Júri:

Presidente: Prof. Doutora Maria Armanda S. Rodrigues Grueau

Arguente: Prof. Doutor Rui Neves Silva

Vogal: Prof. Doutor Vasco Miguel Moreira do Amaral

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2014

A DSL for querying Building Data Streams of Energy, Weather and Occupation Data

Arriving at one goal is the starting point to another.

John Dewey

ACKNOWLEDGEMENTS

First of all I would like to thanks to my parents for all their endless help and support, since I probably would not be at this stage of my career without their advice and encouragement.

To my boyfriend/colleague João Taborda with whom I always could discuss my implementation problems, who calmed me down when problems arose and to whom I must thank for helping me to complete this thesis. Thank you for accompanying me in this journey.

Finally, to my adviser Prof. Vasco Amaral for all of his guidance and recommendations given during the course of this thesis. I also would like to thank to the students and Prof. Paulo Carreira from the Instituto Superior Técnico by their assistance during this dissertation.

RESUMO

O consumo supérfluo de energia é visto pela sociedade moderna como um problema Sócio-Económico e Ambiental dos dias atuais, agravada pelo aumento anual dos preços de energia. Também é de salientar que as pessoas, não proficientes em tecnologias, têm dificuldades em determinar onde é possível poupar, devido à ausência de mecanismos que sensibilizem para este problema.

Uma das preocupações do utilizador em relação à poupança de energia é a possibilidade de fazer a mesma atividade com todo o conforto e a mesma eficiência de trabalho. As técnicas mais utilizadas para controlar o consumo de energia baseiam-se em utilizar equipamentos menos dispendiosos, alterar o programa de funcionamento dos equipamentos para modos mais económicos ou desligar os aparelhos que estão a efetuar atividades desnecessárias. No entanto, não há um retorno de informação verdadeiramente contextual e informativa sobre as ações realizadas e, portanto, o utilizador não sabe o quanto é que essas técnicas influenciaram a sua fatura no final no mês.

A fim de dar algum controlo ao utilizador sobre o seu consumo doméstico de energia, sistemas de monitorização de energia foram desenvolvidos. Estes sistemas permitem o acesso à informação sobre o consumo e ajudam a perceber onde é que se está a desperdiçar energia, porém existem estudos que revelam a discrepância entre a informação fornecida por estes dispositivos e aquilo que os utilizadores realmente necessitam.

De forma a resolver o problema, esta tese tem o objetivo de oferecer uma linguagem especialmente desenhada (DSL) para interrogar os dados do consumo domestico. Esta linguagem foi desenvolvida com o intuito de expressar as preocupações do utilizador através de formulação de questões, sendo portanto, utilizável por não-programadores. O processo de desenvolvimento desta linguagem teve em conta a opinião de peritos e de utilizadores domésticos nas fases de conceção e validação. Estas opiniões foram obtidas através de reuniões e de um inquérito com o propósito de obter as informações mais significativas para os utilizadores.

Palavras-chave: Energia, Consumo, Questionar, Monitorizar, Dados em tempo real, DSL, Linguagens de interrogação específicas para um domínio.

ABSTRACT

The superfluous consumption of energy is faced by the modern society as a Socio-Economical and Environmental problem of the present days. This situation is worsening given that it is becoming clear that the tendency is to increase energy price every year. It is also noticeable that people, not necessarily proficient in technology, are not able to know where savings can be achieved, due to the absence of accessible awareness mechanisms.

One of the home user concerns is to balance the need of reducing energy consumption, while producing the same activity with all the comfort and work efficiency. The common techniques to reduce the consumption are to use a less wasteful equipment, altering the equipment program to a more economical one or disconnecting appliances that are not necessary at the moment. However, there is no direct feedback from this performed actions, which leads to the situation where the user is not aware of the influence that these techniques have in the electrical bill.

With the intension to give some control over the home consumption, Energy Management Systems (EMS) were developed. These systems allow the access to the consumption information and help understanding the energy waste. However, some studies have proven that these systems have a clear mismatch between the information that is presented and the one the user finds useful for his daily life, leading to demotivation of use.

In order to create a solution more oriented towards the user's demands, a specially tailored language (DSL) was implemented. This solution allows the user to acquire the information he considers useful, through the construction of questions about his energy consumption. The development of this language, following the Model Driven Development (MDD) approach, took into consideration the ideas of facility managers and home users in the phases of design and validation. These opinions were gathered through meetings with experts and a survey, which was conducted to the purpose of collecting statistics about what home users want to know.

Keywords: Energy, Consumption, Querying, Monitoring, Real-time data Stream, DSL, Domain Specific Query Language

CONTENTS

Acknowledgements i
Resumo iii
Abstract v
Contents vii
List of Figures xi
List of Tables xiii
List of Listings xv
1 Introduction 1
1.1 Motivation 2
1.2 Objective and Problem Statement 3
1.3 Solution Overview 3
1.4 Expected Contributions 4
1.5 Document Organization 5
2 General Concepts 7
2.1 Energy Management Systems 7
2.2 Software Language Engineering 7
2.2.1 Model Driven Development 8
2.2.2 Domain Specific Languages 8
2.3 Data Stream Management System 9
3 State of the Art 11
3.1 Query Languages 11
3.1.1 Textual Query Languages 11
3.1.2 Non-Textual Query Languages 12
3.1.3 Comparison between the Query Languages 13
3.2 Existing Energy Reporting Solutions 14
3.2.1 House Monitoring Systems 15
3.2.2 House Appliances' Monitoring System 16
3.2.3 Comparison of Energy Monitoring Systems 17
3.3 Final Conclusions 19

4 Technology 21
4.1 Domain Specific Languages 21
4.1.1 Extensible Platform of Integrated Languages for model management (Epsilon) 21
4.1.2 A Tool for Multi-Paradigm Modeling (AToMPM) 23
4.1.3 Choice of the platform 24
4.2 Data Storage and Manipulation 24
4.2.1 Relational Database Management System 25
4.2.2 Data Stream Management System 25
5 EQuery Language 27
5.1 Domain Analysis 28
5.1.1 Analyzing the information provided by the EMS 28
5.1.2 Discussion of the gathered information with domain experts 29
5.1.3 Understanding the home users' real concerns 31
5.1.4 Categorization of the query elements 39
5.1.5 Definition of language models 40
5.2 Design 40
5.2.1 Definition of the abstract syntax 40
5.2.2 Definition of the concrete syntax 41
5.2.3 Elements Incompatibilities 42
5.3 Implementation 43
5.3.1 Ecore and EMF Models 44
5.3.2 Well-formed Rules 44
5.3.3 Language Target Semantics 45
5.3.4 Language Architecture 50
6 EQuery Validation 53
6.1 Language Expressiveness 53
6.2 Usability Studies 55
6.2.1 Evaluation Process 55
6.2.2 Results Analysis 56
6.2.3 Discussion 60
7 Conclusion 61
7.1 Contributions 61
7.2 Future Work 62
8 References 63
9 Appendix 67
9.1 Survey 67

9.2 Features Model 77
9.3 Domain model 78
9.4 Language Metamodel 79
9.5 EMF 80
9.6 Epsilon Validation Language 85
9.7 Epsilon Generation Language 89
9.8 Entity Relationship Model 96
9.9 Usability Test 99

LIST OF FIGURES

Figure 1 – Solution Overview 4
Figure 2 – DSL implementation phases 8
Figure 3 – Natural Query Languages 12
Figure 4 – Non-Textual Query Languages 13
Figure 5 – House Monitoring Systems 15
Figure 6 – House Appliances' Monitoring System 16
Figure 7 – Epsilon Architecture 22
Figure 8 – EQuery Language Overview 28
Figure 9 – Validation of "Which categories of equipment has higher consumption?" 32
Figure 10 – Prefered way for visualizing the results 34
Figure 11 – Relevant Questions for people who pays and do not pays the bill 34
Figure 12 – Relevant Questions for people with different types of household 35
Figure 13 – Relevant questions for the different age groups 35
Figure 14 – Respondents concerns by area of information 37
Figure 15 – Respondents concerns by area of information 38
Figure 16 – Categorization of the query elements 39
Figure 17 – Fragment of the language metamodel 41
Figure 18 – Support Elements 41
Figure 19 – Query Elements 42
Figure 20 – Definition of the Higher Condition on the Ecore 44
Figure 21 – Conversion example from the concrete syntax to SQL code 46
Figure 22 – "What was the last year consumptions?" in the EQuery 46
Figure 23 – "Which areas have higher consumption?" in the EQuery 47
Figure 24 – "Which appliances have consumed more in the peak period?" in the EQuery 47
Figure 25 – "Which equipment are in stand-by mode?" in the EQuery 48
Figure 26 – "Which equipment may be damaged?" in the EQuery 49
Figure 27 – "What is the best tariff for my consumption?" in the EQuery 49
Figure 28 – "How much will I pay in the end of the month?" in the EQuery 50
Figure 29 – "Warning about equipment that are working when they are not needed?" in the EQuery 50
Figure 30 – Language Architecture 51
Figure 31 – Definition of a Notification in Esper 52
Figure 32 – EQuery Eclipse Editor 52
Figure 33 – Evaluation Procedure 55
Figure 34 – Concrete Syntax classification by the participants 58
Figure 35 – EQuery readability chart 59
Figure 36 – Online Survey for gathering the concerns about the energy consumption (Page 1) 67
Figure 37 – Online Survey for gathering the concerns about the energy consumption (Page 2) 68
Figure 38 – Online Survey for gathering the concerns about the energy consumption (Page 3) 69

Figure 39 – Online Survey for gathering the concerns about the energy consumption (Page 4) 71
Figure 40 – Online Survey for gathering the concerns about the energy consumption (Page 5) 72
Figure 41 – Online Survey for gathering the concerns about the energy consumption (Page 6) 74
Figure 42 – Online Survey for gathering the concerns about the energy consumption (Page 7) 75
Figure 43 – Online Survey for gathering the concerns about the energy consumption (Page 8) 76
Figure 44 – Online Survey for gathering the concerns about the energy consumption (Page 9) 76
Figure 45 – Features Model 77
Figure 46 – Domain Model 78
Figure 47 – Ecore Model 79
Figure 48 – Structure Group entities in the MySQL database 96
Figure 49 – Support Group entities in the MySQL database 97
Figure 50 – Historic Group entities in the MySQL database 97
Figure 51 – Entity Relationship Model 98
Figure 52 – Usability Test (Part I. Page 1) 99
Figure 53 – Usability Test (Part I. Page 2) 100
Figure 54 – Usability Test (Part I. Page 3) 101
Figure 55 – Usability Test (Part II. Page 1) 102
Figure 56 – Usability Test (Part II. Page 2) 103
Figure 57 – Usability Test (Part II. Page 3) 104
Figure 58 – Usability Test (Part II. Page 4) 105
Figure 59 – Usability Test (Part II. Page 5) 106

LIST OF TABLES

Table 1 – Table of comparison between DSMS and DBMS 9
Table 2 – Comparison of Query Languages (The symbol ○ indicates that the functionality is not available, the ◑ specifies that does not support it completely and the ● defines that is fully supported.) 14
Table 3 – Comparison of Energy Monitoring Systems 18
Table 4 – Areas of Information 29
Table 5 – Questions retrieved from the meetings with energy experts 30
Table 6 – Areas of Information after the Survey 36
Table 7 – Elements Incompatibilities 42
Table 8 – Language Expressiveness of the pre-established queries (The symbol ○ indicates that is unavailable and the ● that is available in the DSL.) 53
Table 9 – Information about the Usability Test subjects 57
Table 10 – Statistics from the participants' effectiveness and efficiency, using EQuery Language in the tasks elaboration 58
Table 11 – Statistics from the participants' effectiveness and efficiency, when using an Excel Spreadsheet to perform the given tasks 59
Table 12 – Participants withdraw statistics 59
Table 13 – EQuery statistics by group 60

LIST OF LISTINGS

Listing 1 – Definition of the Higher Condition on the EMF 44
Listing 2 – EVL rule that states that the Information About action can only be used with the Tariff subject 45
Listing 3 – Formulation of a SQL query by the main.egl 45
Listing 4 – Function that defines the group by option in a SQL query 45
Listing 5 – "What was the last year consumptions?" in MySQL 46
Listing 6 – “Which areas have higher consumption?” in MySQL 47
Listing 7 – “Which appliances have consumed more in the peak period?” in MySQL 48
Listing 8 – “Which equipment are in stand-by mode?” in MySQL 48
Listing 9 – “Which equipment may be damaged?” in MySQL 49
Listing 10 – “What is the best tariff for my consumption?” in MySQL 49
Listing 11 – “How much will I pay in the end of the month?” in MySQL 50
Listing 12 – “Warning about equipment that are working when they are not needed?” in Esper 50
Listing 13 – Elements representation in the EMF file 80
Listing 14 – Definition of the validation rules in the EVL file. 85
Listing 15 – The EGL file with the SQL semantics 89
Listing 16 – The EGL file that associates each element to the necessary SQL code 89

1

INTRODUCTION

In the last century the energy consumption has increased considerably, in straight correlation with the human activity and population growth. Consequently, this increase is contributing for the exhaustion of our fossil energy reserves, as well as for the degradation of the environment and global warming. Several discussions about this issue have taken place in the international political agenda, and legislation agreements (e.g. citation: Europe Directive 2012/27/EU on Energy Efficiency). In order to solve this problem, sustainable energy sources must be uncovered to replace the imminent worldwide petroleum exhaustion. Additionally, the human kind must invest on using energy in more efficient ways, without severely influence the civilization activities. Subsequently, and due to the increasing of prices every year, the community started to debate about what can be done in order to reduce the consumption, as well as the trade-offs between comfort and energy saving.

In order to start saving it is important to known where the reduction can be made. This problem occurs because the use of energy is harder to keep track, for instance, it is easier to compare two supermarket bills and understand where the money was spent and where it is possible to do some saving. This difference is based on what is possible to see, in the supermarket scenario it is possible to understand the reason that led to the reduction, whereas in the energy bill it is only possible to see the cutback, but not why it was done.

The other main concern, when energy saving is discussed, is the capability to do the same activity with all the comfort and the equivalent amount of work efficiency [1]. In fact, this is achievable, since the activities can be accomplished in a more economical way, by adding some preventive actions. Using a less wasteful equipment, changing the schedule of an activity, altering the equipment program to a more economical one or disconnecting appliances that are not necessary at the moment, are some precautions to take into account. Another way of saving energy is by focusing on the user and his energy consumption activities to detect everyday routines, since 80% of energy used is due to the habits and daily decisions [2] [3]. However, in order to do this preventive actions, more information is needed, such as the consumption of each equipment used in the activities, the type of activities and our every-day habits.

The average rate of energy consumption is influenced by three main points: the ambience characteristics, such as the areas and activities that can be executed; the preferences of the users and

the price of energy [2]. Thus, it is necessary to know each equipment, their consumption and the daily routines, to understand how to waste less energy. For example, knowing that a particular light has a great influence on the final consumption or that the air conditioning is wasting energy. This type of information combined with simple techniques, such as using the blinds for letting the heat get in or for keep it out, or by merely turn off the unnecessary lights and equipment, will help the home user to reduce his energy bill. It has been proved that these procedures, implementable with minimum effort, are able to cut back the consumption by 40% [4]. As expected, the direct consequence of this reduction is that it not only helps to save money but it also helps the environment.

In overall, the energy consumption should be reduced, though this is difficult to do without the awareness of how much each equipment consumes. Therefore, knowing the routines, the equipment and the activities performed, is the ideal way to take an action in order to reduce the usage and the costs.

1.1 MOTIVATION

According to Anderson and White [5], "Everyone understands money. Watts, kilowatts and especially kilowatt-hours will never be universally understood or accepted as units of energy consumption". In fact, it is simpler to understand how much money was saved or what are the current spending, instead of, the amount of kilowatt-hours consumed in the last hour. Therefore, presenting the real-time consumption can encourage a changing of habits [1][5]. A clear example is that by discovering that using a certain type of light bulbs or that a specific equipment is consuming more than necessary, may led to adopt new routines [5] or on adding preventive actions.

In order for adjusting the daily use of energy, it is essential to distinguish and emphasize the different types of relevant information. To be more specific, it must be specified what is the most useful and influential information, to induce a change of habits and achieve minimum consumption in a balanced way. As a result, the information given can be divided into the following categories [5][6]:

- *Real-Time Information*, where the value of actual consumption is indicated, allowing some exchanging to be done. For example, turning off lights in rooms that are not being used at the moment, or disconnect equipment that is in stand-by mode;
- *Historical Information*, that helps knowing the previous actions and consumptions. This type of knowledge is valuable to detect routines and activities accountable for the major use of energy;
- *Predictable Information*, when being aware of the future amount to pay or that pre-established goals will be achieved, is an encouragement. For example, seeing that in the end of the current month there will be a smaller amount to pay.
- *Comparative Information*, where the progress is illustrated, such as the comparison between this month and the last. This category can also be used to promote competition, since it is also a way to reduce consumption [1].
- *Warning Information*, to monitor the consumption and alert for risky situations. This kind of awareness is valuable when anomalies are detected, like having the electrical stove on, when nobody is home.

There are currently some solutions that have been developed through the years, which already manage some of this information in a visual way. These solutions are called Energy Management Systems (EMS) and their goal is to provide a way for the users to monitor their consumption, however

these solutions have some flaws. The first one is the demotivation, seeing that the user starts losing his enthusiasm with time, neglecting the EMS and also their energy saving measures [7]. Another problem is the fact that these systems are built on the notion that "One Size Fits All" [8], which is incorrect because each user has different concerns. Therefore, although allowing the user to access his energy consumption information, there is however a restriction problem when there is the need to combine it in order to produce information that is adaptable to the user needs [3]. Thus, the ideal solution would be that people could express their own adequate questions in a flexible way.

In order to do that, it is necessary to offer a specially tailored language for questioning against the consumption data, but that at the same time usable by non-programmers [1]. Such languages, called Domain Specific Languages (DSL), are languages, usable by programmers and non-programmers, designed for precise domains, which in this context is the formulation of queries about the energy consumption. This language will offer the user the possibility to create his desirable query, with the advantages of using symbolic and graphical notations [9] to express their concerns, which is more expressive and easy to use, than General Purpose Languages (GPL) [9][10].

1.2 Objective and Problem Statement

This thesis aims to create a DSL that allows the information about the consumption of energy to be queried. The idea of this DSL is for the users to consult the details that are more useful for them, as which room is consuming more energy, if there is inefficient equipment or which activity can be performed in a more economical form. In order to create these queries the combination of information is necessary, like the equipment information, the room configuration, the consumed energy for each equipment, the room occupation, and others. However the combination of all of these data may enable the possibility of several types of queries, so the first research interrogation is:

How can we categorize the questions?

With this first interrogation resolved, another problem arises, since the proposed solution has to be intuitive for the users without any programming knowledge to be able to combine different data. Therefore, the second research question is:

Can we create a DSL, expressive enough, which allows the users to create their own questions in terms of the domain of energy consumption to run against stored data?

1.3 Solution Overview

The development of this solution started by identification of the necessary terms in the energy domain. Therefore, it was necessary to understand what information is already offered by the existing solutions. Additionally, to recognize flaws in these solutions, several meetings were organized with domain experts, reaching the conclusion that each user is different and that has different concerns about his energy consumption. For this reason, an online survey was published and from it was possible to retrieve the most relevant questions for the user, as well as the necessary elements for definition of the metamodel.

With the information about the elements necessary to construct a query, the next step was to identify the possible conflicts between elements, in other words, the elements than cannot be present in the same question or elements' requirements. The final step was the language implementation, which included the conversion of the queries to the SQL semantics, and its validation with real users.

In the end, we have a solution that allows the definition of queries, where the query is validated to remove inconsistencies, and then converted to the SQL semantics, being runnable against real-time and historic data.

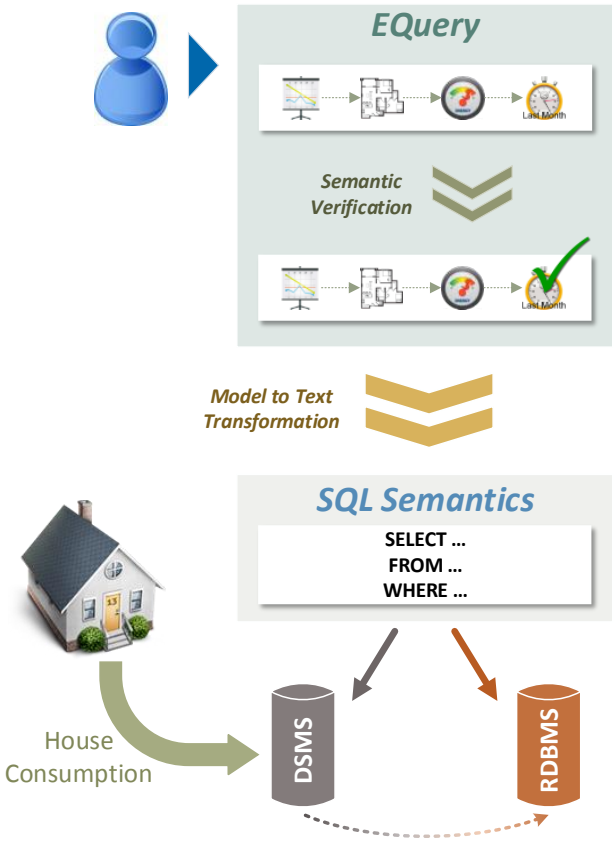


Figure 1 – Solution Overview

1.4 Expected Contributions

With the work and the research done in this domain, it is expected that the following contributions will be achieved:

- Establish a form to categorize the questions, so that the language be able to run the queries expressed by the user;
- Create a language usable by programmers and non-programmers, for the formulation of queries, in this domain.

1.5 DOCUMENT ORGANIZATION

The remainder of this thesis is structured as follows:

- ***General Concepts (Chapter 2) –*** Consists in a brief explanation about the concepts used in this thesis;
- ***State of the Art (Chapter 3) –*** Represents an overview about the work that has been made in this domain;
- ***Technology (Chapter 4) –*** References the technologies that are used in the solution.
- ***EQuery Language (Chapter 5) –*** Presents the several steps necessary to the development of the proposed solution;
- ***EQuery Validation (Chapter 6)*** – Explains the evalution process that was conducted.
- ***Conclusion (Chapter 7) –*** Contains the conclusions aspects, contributions and future work for this dissertation.
- ***References (Chapter 8)*** – The references used through this document.
- ***Appendix (Chapter 9) –*** The figures and models, which are too large to be in the referenced location.

2

GENERAL CONCEPTS

In this chapter it is presented a brief description of some concepts that are discussed throughout this document. These concepts are divided into three groups: Energy Management Systems, Software Language Engineering and Data Stream Management Systems.

2.1 ENERGY MANAGEMENT SYSTEMS

The use of electrical energy is important in our daily lives, since it allows us to perform a variety of actions. However, the use of energy also leads to the pursuit for ways to reduce it, not only for the costs associated, but also with the intention of saving the planet. Nowadays, when trying to moderate the consumption of energy, we have two options available, (1) preventive actions, such as turning off all the unnecessary equipment, reducing its intensity and resorting to the environment conditions; and (2) Energy Management Systems (EMS).

An Energy Management System (EMS) assists in the monitoring, controlling and enhancing the use of energy, promoting efficiency. This type of systems are also advantageous, since they can provide information from the measuring and monitoring, allowing more thoughtful decisions to be taken [11][12].

A more specific type of EMS is Building Management System (BMS), where it is possible to monitor and control equipment that requires a significant energy consumption. BMS is applied in office buildings, universities or factories, bringing benefits not only to the building owner, since they save money by keeping the same level of conditions.

2.2 SOFTWARE LANGUAGE ENGINEERING

In the past years, has been an increase of interest in language engineering [10], more specifically in software languages for the design and creation of programming languages. Due to this demand, it was necessary to create methods for the development of software languages. Therefore, the Software Language Engineering (SLE) arises, as an application of methodology that is systematic, disciplined and quantifiable, in the development, usage and maintenance of software languages [13].

2.2.1 Model Driven Development

Model Driven Development (MDD) is a paradigm where the models are the main artifact in the development process [14]. Therefore, with MDD it is possible to separate the systems design from its implementation, since the representation can be accomplished regardless the details of implementation, such as specific platforms or programming techniques [9][15]. This approach has several models on different layers of abstraction that are used to describe the system, which raises the level of abstraction and hides complexity [14]. These layers are written in specific languages that can be transformed, by an interpreter or by a code generator, into executable code [9]. Some examples of these transformations are model-to-model (M2M) and model-to-text (M2T).

It has been proven that the usage of the MDD approach increases the developer productivity, which reduces the time to market, since it is a solution that can rapidly generate code [16][17]. Additionally, this approach has more quality and is less error-prone [18].

2.2.2 Domain Specific Languages

Domain Specific Languages (DSLs) are languages designed specifically for a domain, based on its concepts and features. Contrary to General Purpose Languages (GPLs), DSLs are focused in the domain of the problem and not in the computational solution, which makes them more expressive and easy to use [9][10]. Opting for the use of DSLs brings some benefits when comparing with GPLs. DSLs are more expressive, since the domain notions are used in the solution, resulting in better understanding by programmers and non-programmers [19][20]. Although sacrificing some flexibility, DSLs are more productive and reliable [9][15], reaching the market quicker, diminishing the maintenance cost and having more capability for optimization [21]. The DSLs are also recommended by their quality, due to the fact that they have less errors and more efficient code [9][13][22].

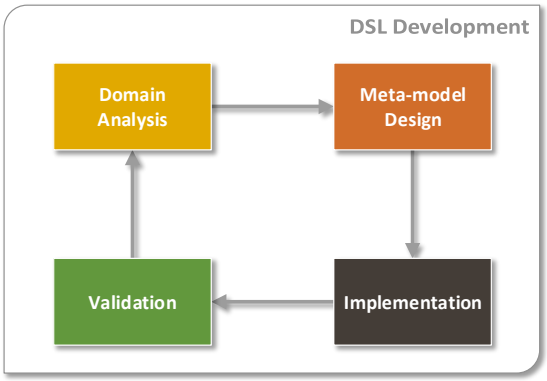


Figure 2 – DSL implementation phases

According to Markus Vöelter [9], the development of a DSL is an iterative process, which comprehends four stages (Figure 2): Domain Analysis, Design, Implementation and Validation. However, seeing that we are developing a DSL for questioning the energy consumption, these phases were performed in view of this field. The Domain Analysis consisted in the gathering of the energy questions that would be essential to the user to execute and, in the understanding of the terms that are necessary to construct those questions. With these concepts gathered the next step is the development of the language metamodel, followed by its implementation. The final stage resides in validating the language with real users, to understand the level of usability in the assembly of energy questions. Although sometimes being neglected, this final stage is essential, since by validating the

language with real users, it is possible to establish the parts that require more attention to detail and also, the improvements that can be achieved in the next iteration of the development process [13][23].

2.3 DATA STREAM MANAGEMENT SYSTEM

Applications that deal with continuous data are increasing in importance, such as stock market quotes, networking monitoring, telecommunications data management or sensor networks [24]. These applications use continuous data streams rather than finite stored data sets. This fact leads to the necessity of having long-running continuous queries as opposed to one-time queries [24], and since the traditional databases are ill-equipped to handle complex and numerous continuous queries over data streams, another type of Database Management System was necessary.

Data Stream Management System (DSMS) is similar to Database Management System (DBMS), but designed for working with transient data instead of static data. DSMS are valuable because they allow the data to be processed in real-time [25], which is essential when fast decisions are needed. For example, in home automation or in the stock market this type of decision is important, since the information provided by these systems expires quickly [26][25]. Another difference between DSMS and DBMS is the storage of data, since the DSMS are not meant for permanent storage, only for temporarily saving data that can be useful in historical queries [25]. The Table 1 summarizes these differences.

Table 1 – Table of comparison between DSMS and DBMS

	Data Stream Management System (DSMS)	**Database Management System (DBMS)**
Data	Transient Data	Static Data
Queries	Continuous queries	One time queries
Storage	Limited Storage	“Unlimited” Storage
Update rate	High	Low
Process	Applies the received data to the defined queries	Applies the queries to the existing data

Given that the user’s formulated queries about the current use of energy may need the access to real-time information, this thesis will resort to the DSMS for their execution. This real-time information is retrieved from sensors, which are (1) the occupancy sensor, for determining the room occupancy; (2) the temperature sensor, for establishing the room temperature and outside temperature; (3) the daylight sensor, for ascertaining the outside level of light; and (4) the electric current sensor, for detecting the electrical current, either direct current (DC) or alternating current (AC). However, it is also possible to add more sensors, for retrieving different information and expand the queries’ domain.

3

STATE OF THE ART

This chapter describes the research done about the existing solutions in this domain, the Query Languages and the existing Energy Management Systems. The first section presents a brief description on the existing types of query languages, showing some existing tools that employ these languages. The second section mentions some of the current solutions for the resident to monitor his home consumption. A few of these systems already allows the user to understand his daily and monthly consumption or even to monitor each equipment independently.

3.1 QUERY LANGUAGES

The objective of a query language is retrieving and combine information present in a database returning the expected result. Over the years, several query languages have been developed for this interaction and they can be classified in two categories, the Textual Query Languages and the Non-Textual Query Languages.

3.1.1 Textual Query Languages

This language type is divided between Natural Query Languages and Artificial Query Languages. The first one was developed with the purpose to be more simply used, removing the need for extensive training [27]. Artificial Query Language or Formal Query Languages, is the type of language that is more used when accessing a database.

3.1.1.1 Natural Query Languages

Natural Query Languages are languages where the user can perform the required query in his natural language (English, Portuguese, or other). This type of language is the ideal solution for non-experts, since the user uses his natural language and does not have to learn a new one. However, it has some disadvantages like the failures due to ambiguities and the coverage, because it is complicated to support all kind of sentence composition [28][29] The following images presents

two examples of tools that use this kind of language. At the left side is presented the Easy Query[1], which offers an ad-hoc query builder for creating the requested query as a sentence. With this tool, the user has to go to Column and Condition tab and chosen the wanted data. The SQL and Result tab shows the query in text and query results. On the right we have the SimpleQL[2], a tool designed to simplify the data analysis. SimpleQL offers a web interface where the data can be queried through questions, being adaptable to every type of database.

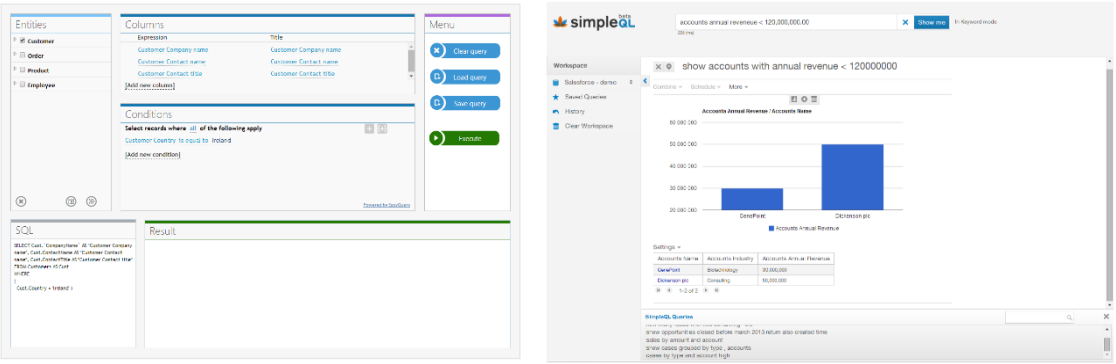


Figure 3 – Natural Query Languages

3.1.1.2 Artificial Formal Query Languages

The Artificial Query Languages are the most common languages for performing an access to a database. The most familiar is the SQL (Structured Query Language) used for accessing relational databases, however much more languages are included in this category, like the OQL (Object Query Language) that is applied in object oriented databases. These type of languages have advantages, such as the formalism and the decrease of ambiguity. On other hand, the user has to learn the syntax and understand how the information is linked [28].

Included in this category is the XQuery[3] and Java Object Oriented Querying[4] (JOOQ). The XQuery is used to query and transform data in the form of XML, while JOOQ is a software library in java that provides a Domain Specific Language (DSL) to construct queries.

3.1.2 Non-Textual Query Languages

Non-textual Query Languages are a type of languages that use Visual Query Language (VQL), for expressing the requested information. These type of languages are easier to learn and can be more efficient with non-expert users, but they can also improve the programmer's productivity. A disadvantage of these languages is that they have less formalism than the textual ones.

In the Figure 4 it is represented the GestureQuery and the Visual Trace Modeling Language (VTML). Gesture Query (Figure 4a) is a multi-touch interface for querying a database, allowing a rapid interaction and query refined, even by non-expert [30]. Its interface is composed by three parts, the list of tables, the database selector and the workspace, where the questions are formed. This tool allows the user to perform the following actions: (1) filter; (2) sorting; (3) preview; (4) group by and aggregate; (5) rearrange; (6) join and (7) union. Whereas the second one is a language that, according

[1] http://devtools.korzh.com/easyquery
[2] http://simpleql.com/
[3] http://www.w3.org/XML/Query/
[4] http://www.jooq.org/

to the authors, offers a relatively simple and intuitive way for querying the database. VTLM (Figure 4b) uses the standard UML class diagrams to model queries. The reason for this approach, is that it can be used by any organization familiar with UML and implemented with standard tools [31].

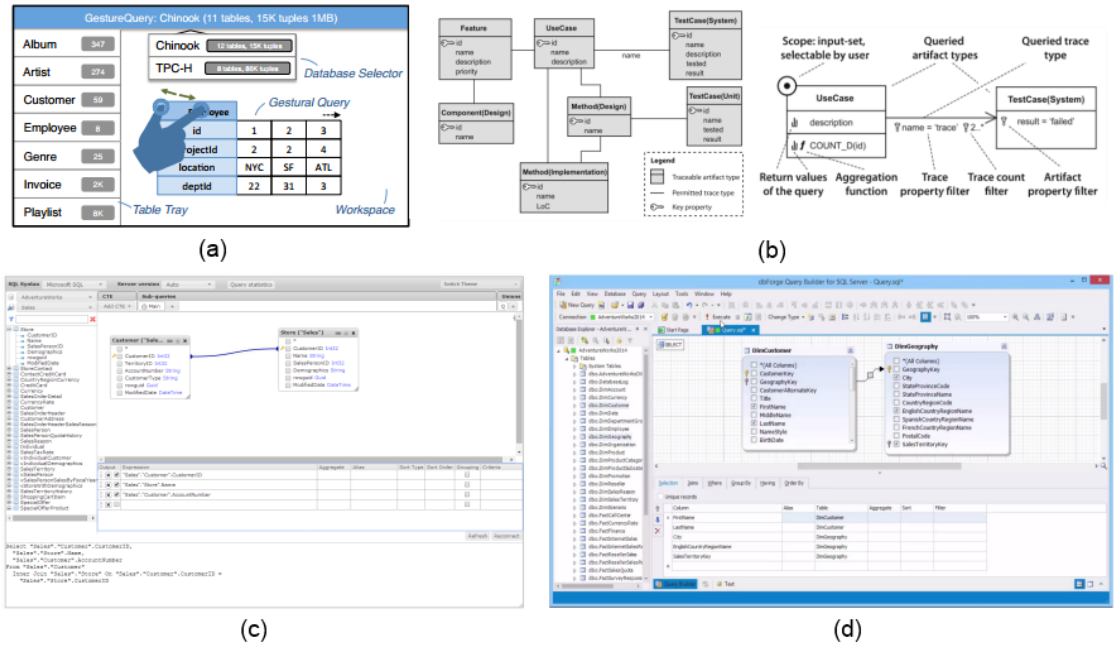


Figure 4 – Non-Textual Query Languages

Included in this category is the QBE (Query-by-Example) and the Hybrid Approach [28]. The QBE consists in providing to the user a table that he fills with the required commands and then the system creates the query, based on this table [32]. The Hybrid Approach is the more common and provides the option to create the query through a visual dashboard or text. The Figure 4 presents two tools that use this approach, Active Query Builder[5] (Figure 4c) and dbForge Query Builder[6] (Figure 4d). These tools allow expert users to build SQL Queries in a visual form, offering also the ability to make SQL text editing.

3.1.3 Comparison between the Query Languages

From the Table 2, it is possible to perceive that the Natural Query Languages are the only ones adaptable to non-programmers, since this type of language was designed with the intent to be more simple and easier to use without being necessary to given extensive training. However, due to the language limitations, some education has to be given in order to understand the database notions and hidden constraints [27]

The Formal Query Languages are more difficult to use, because of query complexity and the necessity to understand the database structure [27][33], while the Visual Query Languages are more intuitive for a user, since it is easier to express the relations between objects. Nevertheless, they have limited expressiveness, since they are implemented to be simple and easy to use [33].

[5] http://www.activequerybuilder.com
[6] http://www.devart.com

Table 2 – Comparison of Query Languages (The symbol ○ indicates that the functionality is not available, the ◑ specifies that does not support it completely and the ● defines that is fully supported.)

		TEXTUAL QUERY LANGUAGES				**NON-TEXTUAL QUERY LANGUAGES**			
		Natural Query Languages		***Artificial Query Languages***		***Visual***		***Hybrid***	
		Easy Query	SimpleQL	XQuery	JOOQ	Gesture Query	VMTL	Active Query Builder	DbForge Query Builder
USER	***Programmer***	●	●	●	●	●	●	●	●
	Non-Programmer	◑	◑	○	○	◑	○	○	○
Graphical Queries		●	●	○	○	●	●	●	●
Ad-hoc Queries		●	●	●	●	●	●	●	●
Continuous Queries		○	○	○	○	○	○	○	○
Requires knowledge of queries notions		●	●	●	●	●	●	●	●

In the above table several tools from different types of query languages were presented, each one of them having into account different goals. Although all of them permitting the creation of queries that can combine different information to generate new one, these languages are not able to deal with continuous data. Consequently, they are not the ideal ones to question the real-time data received from the electric current sensors.

3.2 EXISTING ENERGY REPORTING SOLUTIONS

A significant share of our daily activities in a modern society depends on energy to be successfully executed. In Europe, as stated in the Eurostat study of 2010 [34], the residential consumption represents up to 27% of the entire energy produced. However, this energy is not efficiently used because a large part of it can be wasted without the residents being aware of it. This fact represents a major problem for people, since they are paying for something that they are not maximizing its use, because they do not have a way to understand where the energy is being wasted in activities that are unnecessary. For this reason, there is a demand for systems that gather information about the household consumption, in order to decrease the waste and cost of the consumed energy.

Energy Management Systems (EMS) encourage a more efficient use of energy, because they can monitor, control and enhance its use. These systems allow more thoughtful decisions to be taken, since the resident is more aware of his energy consumption. There are two types of EMS, House Monitoring Systems and House Appliances' Monitoring Systems. The systems in the first group offer the access to the global consumption, while the second one has more detailed information, allowing the identification of the consumption of each appliance [11].

3.2.1 House Monitoring Systems

House Monitoring Systems are designed to help the home users to keep track of their global energy consumption. These systems can provide different types of information, such as the consumption by month or the consumption preview. However, this group of systems, cannot help the user to completely understand where the consumption is higher, since they do not have the information distributed by appliances.

The following image presents six different systems for monitoring the domestic energy consumption. All of these systems are similar, since they have the same goal, therefore, the functionalities are identical between them.

Figure 5 – House Monitoring Systems

The Energy Detective (TED)[7] is an EMS designed for helping reducing the bill (Figure 5a). TED offers different applications for Residential or Commercial/Industrial use. This system provides the information about the current consumption in a wireless device, a webpage for monitoring and the possibility to define alerts to the mobile phone. The monitoring view allows the user to access the following information: (1) current consumption; (2) average of money spent by day; (3) money spent since midnight; (4) money spent that month and (5) bill projections.

Engage Efergy[8] (Figure 5b) is a system for monitoring the use of energy, allowing the user to access the data from his computer, smartphone or tablet. This platform allows the following actions: (1) checking the real-time usage; (2) setting a budget target; (3) consulting the current spending and (4) examining the energy consumption history.

GEO Minim[9] is a system that consists in a display that provides the information about the energy consumption (Figure 5c). This system allows the home users to access the following information: (1) current consumption; (2) consumption by day, week or month; (3) daily target setting and (4) tariff indicator.

[7] http://www.theenergydetective.com/
[8] http://engage.efergy.com/
[9] http://www.greenenergyoptions.co.uk/

Owl[10] is a UK company that provides two solution for energy monitoring, a simple (Figure 5d) and a smart one. Both of this solutions offers the user information similar to the GEO Minim: (1) current consumption; (2) consumption by day, week or month; (3) consumption average by day, week or month; (4) tariff indicator and (5) budget settings.

Eco-Eye[11] is a system that allows users to check their energy consumption through a display device (Figure 5e). This system offers the following information: (1) current consumption; (2) money spent by day, week, month or year and (3) target definition.

Current Cost[12] provides different types of displays (Figure 5f) for helping identifying the waste of energy consumption. These displays offer the following knowledge: (1) current consumption and money spent and (2) money spent by day, week, or month.

3.2.2 House Appliances' Monitoring System

These types of systems are similar to the previous ones, since they have the same purpose: inform the user about his energy consumption. Nevertheless, these systems are more suitable if the home user wants to understand which appliance have a higher usage of energy, instead of receiving information pertaining to his main consumption. The Figure 6 presents six systems of this kind.

Figure 6 – House Appliances' Monitoring System

OpenEnergyMonitor[13] is an open-source project for energy monitoring (Figure 6a). This system is Arduino compatible and consists in four parts: (1) the emonTx, responsible for monitoring part; (2) the emonGLCD, a wireless display; (3) the emonBase, responsible for receiving the data from the emonTX and (4) the emoncms, a web-app for data visualization. The web-app allows the user to define the data feeds of the information that is being retrieved and create his dashboard with visualization of this feeds.

Cloggy[14] (Figure 6b) is a solution for home users to manage their energy consumption. This product allows for the monitoring of the energy usage through a smart plug that also allows for the control

[10] http://www.theowl.com/
[11] http://www.eco-eye.com/
[12] http://www.currentcost.com/products.html
[13] http://openenergymonitor.org/emon/
[14] http://www.cloogy.com/en/

of the connected appliances. The information offered by this system is viewable by a webpage, a smartphone or through a small monitor, being the following: (1) access to the current consumption; (2) average consumption; (3) consumption preview and (4) consumption by appliance.

Tendril[15] is a company that offers systems to control the consumption of energy (Figure 6c). This system allows the home user to keep track of his usage of energy through web applications, mobile application or though devices. With this system the user can access the following information: (1) real-time consumption; (2) yesterday's consumption; (3) consumption by appliance; (4) consumption from the last day, month or year and (5) consumption preview.

The remote energy dynamics (re:dy)[16] is a system offered by EDP (Energia de Portugal), the main supplier of energy in Portugal (Figure 6d). This system allows the user to access his consumption through a smartphone or a webpage where the provided information is: (1) real time information about the current consumption; (2) chart base historic; (3) consumption by appliance and (4) recommend energy prices.

GEO Ensemble[17] (Figure 6e) like Geo Minim is a system that consists in a display that provides the information about the energy consumption (Figure 5c). The Ensemble allows the user to access the knowledge already offered by the Minim, but also allows to access the information through a web-page or a mobile app and knowing the consumption by appliance. Thus, this system offers the following information: (1) current consumption; (2) consumption by day, week, month or year; (3) target setting; (4) tariff indicator and (5) consumption by appliance.

AlertMe Energy[18] is a system for accessing the real-time consumption information from the household (Figure 6f). As the previous, this system allows the following information: (1) current consumption; (2) consumption from day, night, evening and morning; (3) consumption prediction; (4) consumption by appliance and (5) monthly consumptions.

3.2.3 Comparison of Energy Monitoring Systems

Given the presented EMS, it is possible to assert that a home user has currently at his disposal a wide plethora of systems that can help him controlling his energy consumption. However, in order to better understand the information provided by these systems, the Table 3 was created to summarize the functionalities of each one.

[15] http://www.tendrilinc.com/
[16] https://energia.edp.pt/particulares/servicos/redy.aspx
[17] http://www.greenenergyoptions.co.uk/
[18] https://www.alertme.com/

Table 3 – Comparison of Energy Monitoring Systems

		HOUSE MONITORING SYSTEMS						**HOUSE APPLIANCES' MONITORING SYSTEM**					
		TED	*Engage Efergy*	*GEO Minim*	*Owl*	*Eco-Eye*	*Current Cost*	*Emoncms*	*Cloogy*	*Tendril*	*re:dy*	*GEO Ensemble*	*AlertMe Energy*
Consumption	*Current*	●	●	●	●	●	●	●	●	●	●	●	●
	Daily	●	●	●	●	●	●	●	○	●	○	●	●
	Weekly	○	○	●	●	●	●	●	○	○	○	●	○
	Monthly	●	●	●	●	●	●	●	●	●	●	●	●
	Annual	○	○	○	○	●	○	●	○	●	○	○	○
Consumption Evolution	*Daily*	●	●	○	○	○	○	●	●	●	●	○	●
	Weekly	●	○	○	○	○	○	●	●	○	○	○	○
	Monthly	◑	●	○	○	○	○	●	●	●	●	○	●
	Annual	●	●	○	○	○	○	●	●	●	●	○	○
Consumption Average	*Daily*	●	○	○	●	○	○	○	●	○	○	○	○
	Weekly	○	○	○	●	○	○	○	○	○	○	○	○
	Monthly	○	○	○	●	○	○	○	○	○	○	○	○
Consumption by Space		○	○	○	○	○	○	○	○	●	○	○	○
Consumption by Appliance		○	○	○	○	○	○	●	●	●	●	●	●
Tariff	*The best tariff*	○	○	○	○	○	○	○	●	○	●	○	○
	Bill preview	◑	○	○	○	○	○	○	●	◑	●	○	◑
	Budget limit	○	●	●	●	●	○	○	●	●	○	●	○
Notification		◑	○	○	○	○	○	◑	●	○	●	○	○

The previous table shows the several types of information provided by the referred Energy Management Systems (EMS), which is divided in six categories:

- *Consumption*: Implies that the marked EMS presents the different values of consumption (current, daily, weekly, monthly or annually). These values are presented in two different units, kwh and currency (€, £ or $);
- *Consumption Evolution*: This category is applied when the EMS gives the possibility to observe the consumption values through a period of time;
- *Average Consumption*: Marked when the EMS presents the average value of a period of time;
- *Spaces Consumption*: States that the EMS offers information about the consumption of each space.
- *Appliances Consumption*: When the EMS offers information about the consumption of each appliance.

- *Tariff*: Used when the EMS provides information about the payable values.
- *Notification:* Whether the EMS gives alerts about the consumption.

Between all of these systems, the user can access different types of information, becoming more aware of his energy consumption. There are however two mains problems: the first one is how to choose the more appropriate system, and the other is to know if that system can answer the user's needs.

From the previously described systems it is possible to observe that the most of them provide enough information about the Consumption. When we observe the Consumption Evolution, the House Monitoring Systems offer a smaller amount of solutions, compared with the House Appliances' Monitoring Systems. If the user wants a solution that provides information about the Average Consumption, the number of choices is really limited. Another additional issue with the presented systems are the Notifications and the Space Consumption, since few of them provides these functionalities, which is a characteritic that should be expected in these types of systems [7]. Especially the Alert features since it helps to keep the user motivated.

It is possible to conclude that these systems already provide information that is helpful when pretending to control the energy consumption. However, as mentioned, these systems are based on the concept that "One Size Fits All" [8] and none of them allows the combination of information to retrieve new information. The only one that offers similar functionality is the Emoncms which allows to configure the data feeds, however this system requires some specific knowledge about the data that is being read and how to handle it.

3.3 Final Conclusions

In the two previous sections, there was an analysis of the available tools in for monitoring the energy consumption (Section 3.2) and for performing queries (Section 3.1). The Energy Monitoring Systems (EMS) are already a good option for who wants to understand their consumption and control the energy waste. However, these systems are created to be adaptable to all of the users, which makes it impossible for the user to retrieve personalize the want information. Additionally the studied EMS are ill-prepared for providing information about the equipment, area or category consumption, as well as a complete notification system with several types of warnings and also with motivation tips that could led to more savings. Another issue detected was that the majority of the EMS fail when the user wants to predict how much he will pay in the end of the month, since they only require the price per kWh, discarding that existence of tariffs with different prices per period.

On the study of the existing Query Language, it was ascertain that already exist some tools that are concerned with the usability by a non-programmer. Nevertheless, these languages have a higher degree of complexity due to the fact that they are used for general purposes, not being domain specific. Therefore, there is always the need to train the user with notion about the existing data and their relationship.

With this in mind, the proposed language should keep the best ideas from each topic and correct the identified weaknesses. In other words, the language should provide more information about the house equipment and also offer notifications. Moreover, this language should also be able to form personalized questions and also be usable by non-programmers without the necessity to acquire knowledge in query notions.

4

TECHNOLOGY

In this chapter it will be discussed the tools used in order to successfully develop the solution for the described problem. As a reminder, the problem that this thesis wants to solve is how to provide a language that allows users to create their own questions about the consumption of energy.

Therefore, two different types of tools will be present. Tools for the Domain Specific Language (DSL) implementation and Data Storage and Manipulation that will storage information and execute the queries.

4.1 DOMAIN SPECIFIC LANGUAGES

As mentioned in the section 2.2.2, the Domain Specific Languages are tailored for a specific problem, offering more maintainability, reliability and productivity, since the code size is reduced. This facts led to an increase of interest and to a rise of tools for its implementation.

Currently there are several tools that support the development of a DSL, being the most used the MetaEdit+[19], the GME[20], the AtomPM[21] and the Epsilon[22]. However, due to the fact that the MetaEdit+ is commercial and that GME is platform dependent (runnable only on Microsoft Windows), the only tools considered were the AtomPM and the Epsilon.

4.1.1 Extensible Platform of Integrated Languages for model management (Epsilon)

Epsilon is a platform composed by a family of task specific languages and tools, integrated in the Eclipse Generative Modeling Technologies (GMT) project. This tool offers an infrastructure that can

[19] http://www.metacase.com/mep/
[20] http://www.isis.vanderbilt.edu/Projects/gme/
[21] http://syriani.cs.ua.edu/atompm/atompm.htm
[22] https://www.eclipse.org/epsilon

operate with different modeling framework, such as the Eclipse Modeling Framework (EMF), Meta Data repository (MDF) and plain XML [35][36][37].

The Epsilon Object Language (EOL) is the main language of Epsilon, since the others are built from it. EOL is a language based on the Object Constraint Language (OCL), allowing model modification, conditional and loop statements, user interaction, and others [35][38]. Although EOL being used as a general purpose model management language [35], its main purpose is to be reused by the other task specific languages provided: Epsilon Generation Language (EGL) for model-to-text transformation; Epsilon Transformation Language (ETL) for model-to-model transformations; Epsilon Merging Language (EML) for merging model, Epsilon Validation Language (EVL) for model validation; Epsilon Wizard Language (EWL) for model refactoring; Epsilon Comparison Language (ECL) for model comparison and Epsilon Flock for model migration.

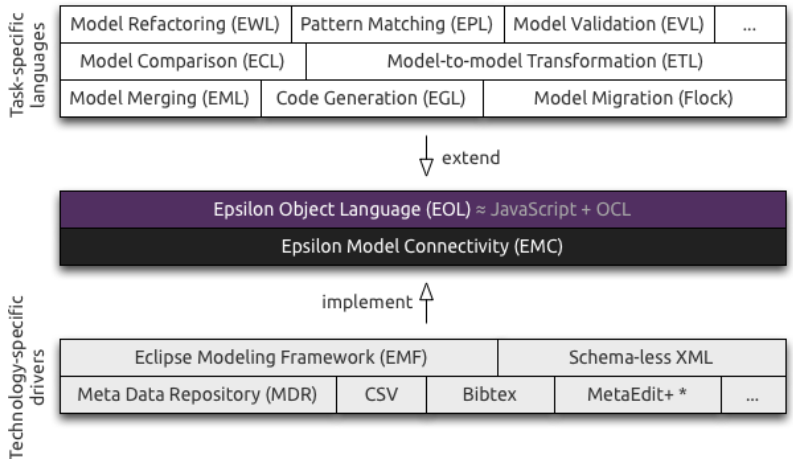


Figure 7 – Epsilon Architecture[23]

As illustrated in the Figure 7, all the task specific languages are built on top of the EOL and, in order to support all the different types of models, it was necessary to implement the Epsilon Model Connectivity (EMC) [39], that deals with all the several types of modelling technologies.

4.1.1.1 Task Specific Languages

This section will focus on the languages that will be needed in the solution. As mentioned in the previous section Epsilon supports different types of languages, where each one is responsible for particular assignments.

Epsilon Object Language (EOL) is an imperative programming language for creating, modifying and querying EMF models. It was also build with the intention to reuse OCL navigation mechanism, offering support for other languages in model access, model modifications or error streams. EOL is the core language in which all the other task specific languages were developed, such as ETL and EGL [39][38].

Epsilon Generation Language (EGL) is a specific language for transforming a model into text, being this text, executable code, reports, or images. As mentioned, EGL is built over the Epsilon Object Language and reuses its mechanism for controlling the program flow, navigation and model inspection. Since EGL is a template-based code generator, it has some ways to ease that generation. For instance, it includes a merging engine that preserves hand-written sections and a

[23] Taken from http://www.eclipse.org/epsilon/doc/

system that allows the text to be generated from different sources. It also provides a formatting algorithm for producing readable code and a mechanism for traceability of the text to the model [39][40].

Epsilon Transformation Language (ETL) is a task specific language for model transformation, built on top of EOL, with the purpose to define rules and execute schemes. From the EOL, it receives its imperative characteristic for dealing with more complex models. Instead of turning one model into another, ETL can transform an arbitrary number of inputs, into an arbitrary number of outputs in different modeling languages. In order to do so, ETL transformations are grouped in modules, where each module can contain any number of rules and operations. In addition to transformation rules, an ETL module can also specify blocks of EOL statements that can be executed before or after the transformation [39][35].

Epsilon Validation Language (EVL) main objectives are to evaluate constraints on metamodels, checking their intra and inter-model consistency and giving some quick forms to repair their inconsistencies. Epsilon Validation Language is also combined with EMF/GMF, thus the constraints evaluation can be done inside their editors, generating also error messages [39]

4.1.1.2 Tools

Epsilon different tools allows code generation, model transformation and others. One of this tools is EuGENia, responsible for the models needed for the GMF Editor (.gmfgraph, .gmftool and .gmfmap). EuGENia generates these models through an Ecore metamodel, lowering the complexity of GMF by offering a high-level form of making annotations.

4.1.1.3 EMF Support

Epsilon is not bound to the Eclipse Modeling Framework (EMF), however provides a strong support, by the implementation of a driver for the Epsilon Model Connectivity (EMC) layer. This driver allows for all types of EMF, such as reflective, generated and non-XMI models that are specified using XText or EMFText.

The EMF, as describe in its official page, is framework for modeling and code generation based on structured data model. The EMF produces a set of Java classes for the model, specified using a XMI. Models can also be specified using annotated Java, XML documents [41] or using an Ecore metamodel. In order to specify a graphically concrete syntax, for a model defined with EMF, the usage of tools like the Graphical Modeling Framework (GMF) is needed [42].

4.1.2 A Tool for Multi-Paradigm Modeling (AToMPM)

AToMPM, the successor of AToM^3, is a tool that is being developed by several people, namely Eugene Syriani and Hans Vanghelawe. They present an open-source framework that allows design Domain Specific Modeling Languages, modeling transformations and managing models [43][44]. This framework was projected to run on the web, with the purpose to be independent from the operating system or the device, but also because it allows co-workers to share and cooperate with each other in the same project at the same time. This type of real-time collaboration can be done in two ways, by Screenshare or by Modelshare [45][43][46]. The first one, offers the possibility of multiple users share the same canvas and any alteration will be updated in the other canvas, while the second one, indicates that only the abstract syntax of the model is shared [43].

Its designers considered it to be a modern and versatile way for develop all of the parts of the system with the most appropriate level of abstraction and formalism, since it can be modeled using a combination of UML Class Diagram and Statecharts. AToMPM is also a bootstrap, since it does all the modeling by itself.

This tool provides an interface where the actions like defining metamodels, establishing rules and executing transformations are made graphically, by the usage of mouse clicks or movements. However, it is also possible to use textual commands [43][44].

4.1.2.1 Modeling

AToMPM main functionality is modeling and metamodeling, thus a meta-metamodel is defined to be followed by the metamodels. This meta-metamodels can be an Entity Relationships Model or a basic version of UML Class Diagram [45][43]. In this tool, when a meta-model is defined it is automatically compiled, but in order for than to happen the parallelism between concrete syntax and abstract syntax is needed [45].

4.1.2.2 Transformations

AToMPM transformations are based in the Transformation Core (T-Core) framework developed by the same authors of AToMPM, with the purpose to combine or reuse transformations implemented in different languages. This type of framework was needed since there are many forms to execute transformations, besides, with this framework it is possible to plan transformations more suitable for the current task [47].

The transformations can be performed in two ways, release and debug. The first one, where the model is sent to the server, transformed and then returned to the client's side; or the debug mode, where the transformation in done step-by-step or continuously in the client's side [43].

4.1.3 Choice of the platform

The choice between the two presented platforms for the DSL development, Epsilon and AtomPM, was based on advantages that the tool could bring to the language development. In the end, the Epsilon tool was the one that was considered the most appropriated for implementing this thesis prototype, due to its several years of development and improvements, and also because of its great community. Opposing to the AtomPM that is a very recent tool with a reduced support community.

4.2 DATA STORAGE AND MANIPULATION

To complement the developed DSL, it was necessary to store relevant data, manage and query it. As mentioned, the proposed solution allows the users to query the real-time and historic data, therefore, there is a need to have a Data Stream Management System (DSMS) and a Relational Database Manage System (RDBMS) to handle these types of information.

The DSMS is responsible for the real-time information, while the RDBMS stores all the structural and historical information. The use of these two types is necessary for allowing the user to question his current and past consumptions.

4.2.1 Relational Database Management System

RDBMS is Database Management System (DBMS) based on the relational model and was introduced by E. F. Codd in 1970. This type of system is used in this project for storing the structural and historic data, where the structural data is the characteristics of an equipment or of a space and the historic is about relevant details, as the energy consumption of which day.

For managing this information it was choose the MySQL[24] platform. MySQL is a popular open-source database that is mainly used in web applications. For the purpose of this thesis, was considered that the use of RDMS such as Oracle or MSSQL were excessively complex for the target objective. Therefore, MySQL appears as a viable alternative because it can be used in many platforms and supports the essential SQL commands for storing and retrieving information.

4.2.2 Data Stream Management System

For dealing with the real-time data, RDBMS are not the right approach, since they are used for manage static data and where complex queries are less frequent. DSMS as mentioned in the section 2.3, are systems designed for working with transient data and that perform continuous queries over the data. However, with the need for real-time process event in applications, several DSMS were developed:

- STREAM[25], AURORA[26], Niagara[27], TelegraphQC[28] and PIPES[29] were dissertation projects that implement the notions of DSMS, but their development was dropped once the projects were concluded.
- StreamBase[30] and InfoStream[31] are two types of platforms able to perform the necessary functions, but they are proprietary systems and, because of that, they have a limited support from the community.

Given this reasons, we choose Esper[32] to perform the customized queries over the real-time data. Esper is an open source event stream processing (ESP) and Complex Event Processing (CEP) [48] available for Java as Esper, and for .NET as NEsper. In order for processing real-time event, Esper offers the Event Processing Language (EPL). EPL is a declarative language for perform filtering, aggregation and joins, that can be over sliding windows of multiple event series.

The Complex Event Processing from Esper is considered as a traditional database turned upside-down [48]. As mentioned in the Section 2.3, this type of databases do not perform a query against storage data, they store queries and run the data through it, being able to respond in real-time when the data match the query. Esper also provides a Relational Database Adapter for saving and joining real-time data, with historical data, contained in standard relational database using JDBC.

[24] http://www.mysql.com/
[25] http://www-db.stanford.edu/stream
[26] http://cs.brown.edu/research/aurora/
[27] http://research.cs.wisc.edu/niagara/
[28] http://telegraph.cs.berkeley.edu/telegraphcq/
[29] http://dbs.mathematik.uni-marburg.de/Home/Research/Projects/PIPES
[30] http://www.streambase.com/
[31] http://www-01.ibm.com/software/data/infosphere/streams/
[32] http://esper.codehaus.org/

5

EQUERY LANGUAGE

The main goal of this thesis is to develop a Domain Specific Language (DSL), named EQuery (Energy Query), which allows the definition of energy queries. These questions are applied to the home consumption, providing information that can be useful in the reduction of waste energy.

The EQuery development, schemed in the Figure 8, took into account the DSL implementation phases discussed in the Section 2.2.2. Initially was necessary to comprehend the existing information and their flaws, which allowed to determine the relevant information for the users and, consequently, the necessary grammar that permits the questions to be constructed. With these terms, it was possible to define the language metamodel and validations rules that have to be applied. Having the language metamodel defined in the Eugenia and the definition of an Epsilon Validation Language (EVL) file with the previous rules implemented, the next phase was the development of Epsilon Generation Language (EGL) that converts the question in code for the MySQL or for Esper. The MySQL Database will save the user's consumption history and structural data, such as the characteristics of the spaces, the equipment and of the activities, while the Esper Database is responsible for querying the real-time data received from the sensors.

This chapter organization is based on the four DSL implementation phases. In first section explains the methodology followed to perform the Domain Analysis, which includes meetings with domain experts and a public survey. With the information retrieved, the second part describes the language metamodel, followed by its implementation and validation with real users.

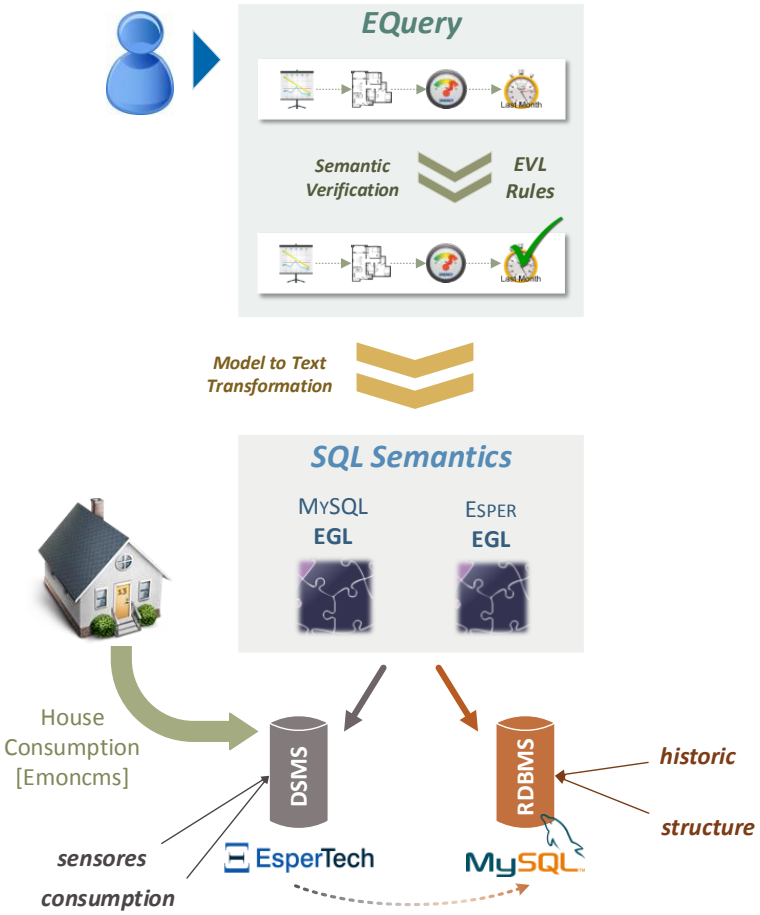


Figure 8 – EQuery Language Overview

5.1 Domain Analysis

In order to start developing the EQuery Language the first step was to comprehend the domain, including his concepts and notions. This study of terms, specific of the domain, permitted the definition of the query elements necessary to build the energy questions.

In order to accomplish this, was necessary to perform the following tasks:

1. Analyzing the information provided by the EMS;
2. Discuss with domain experts the relevant areas of information;
3. Understanding the home user's real concerns;
4. Categorization of the query elements;
5. Definition of language models.

5.1.1 Analyzing the information provided by the EMS

In order to build a language that allows the creation of energy queries it was necessary to first understand the type of information that the existing systems already provide, which was presented in the Table 3 of the Section 3.2.3.

During the analysis of the EMS, it was concluded that these systems offer to all users the information that they consider pertinent. However, this is problematic, since each user has

different concerns, which makes the information that is considered relevant variable from user to user [8]. Some studies also proven that this provided information may not be the most adequate one, since it can lead to demotivation about the product and consequently about the good practices in energy saving [49] [7].

In the mentioned section, it was also discussed that these systems, do not provide the information necessary to induce a behavior change in the consumption of energy, such as the notifications, abnormal equipment consumption and energy recommendations [49] [7].

In order to have a starting point, the Table 4 joins the information that can be relevant for the consumption of energy in a household, joining the information from the EMS and from the EMS studies.

Table 4 – Areas of Information

Areas	Topics
Consumption by period of time	Consumption values from different periods
Consumption Evolution	Consumption comparison with historical values
Average Consumption	The average consumption values from different periods
Consumption by Space	The total amount of energy consumption in a room
Consumption by Category	The total amount of energy consumed by a group of equipment
Consumption by Appliance	Identification of the unnecessary working equipment
	Indication of abnormal consumptions
Tariff	Tariff Comparison
	Budget Setting
	Cost Preview
Notifications	Warning Information
	Saving Suggestions

5.1.2 Discussion of the gathered information with domain experts

In the previous section some flaws about the EMS were kept in mind. One of them is the relevancy of the information, since we are not sure if these systems are based in the user needs. The other one is the lacking of information that may be essential to change the daily routines.

Taking into account the referred flaws, it was necessary to identify if the eight areas of information, present in Table 4, are the must adequate to the home users. In order to accomplish this, several meetings were held with students, professors and facility managers from *Faculdade de Ciências e Tecnologia*[33] and *Instituto Superior Técnico*[34]*.* During these meetings was discussed the adequacy

[33] http://www.fct.unl.pt/
[34] http://tecnico.ulisboa.pt/

of certain areas of information, as the addition of new sub-topics and some questions that could be answered, resulting in the Table 5.

Table 5 – Questions retrieved from the meetings with energy experts

Areas	**Topics**	**Questions**
Consumption by period of time	Consumption values from different periods	*What is the actual consumption?*
		What was the last day consumption?
		What was the last month consumption?
		What was the last year consumption?
Consumption Evolution	Consumption comparison with historical values	*What was the evolution of consumption from the last two years?*
Average Consumption	The average consumption values from different periods	*What is my daily consumption?*
		What is my monthly consumption?
		What is my annual consumption?
Consumption by Space	The total amount of energy consumption in a room	*Which spaces have the higher consumption?*
		What is the consumption of each space?
		Which spaces have consumed more in the peak period?
Consumption by Category	The total amount of energy consumed by a group of equipment	*Which categories of equipment has higher consumption?*
Consumption by Appliance	Consumption of a specific equipment	*What is the consumption of a specific equipment?*
		Which equipment have a higher consumption?
	Identification of the unnecessary working equipment	*Which equipment are in stand-by mode?*
		Which equipment are working when they are not needed?
	Indication of abnormal consumptions	*Which equipment may be damaged?*
		Which equipment are consuming more than they should?
	Identification of energy waste	*Which appliances are using programs with unnecessary power?*
Tariff	Tariff Comparison	*What is the best pricelist for my energy consumption?*
	Budget Setting	*For how much did I exceeded my target budget?*
	Cost Preview	*How much will I pay in the end of the month?*
Notifications	Warning Information	*Warning about equipment are in stand-by mode*
		Warning about equipment that are working when they are not needed?
		Warning about equipment that are consuming more than they should?
	Saving Suggestions	*Which equipment should be replaced?*

Nevertheless, despite the involvement of experts on the fulfillment of these areas, it was realized that it was necessary to confirm these areas with real home users, in order to validate their relevancy and coverage.

5.1.3 Understanding the home users' real concerns

During the meetings with domain experts was noticeable that it was necessary to have the opinion of real home users in the selection of the most relevant areas of information. Therefore, an online survey was constructed and published. This survey was created in the SogoSurvey[35] platform, since from the examined platforms, it was the one that provided less restrictions and more features with the free license. The published survey can be accessed in the Appendix 9.1.

In order to spread this survey we reach out to the MIT program[36] of the *Instituto Superior Técnico,* which provided the contacts of people that already participated in other energy related projects. Besides these contacts, the survey was also published through social networks and to the contacts of the involved authors, including both campus.

5.1.3.1 Work Hypothesis

With this survey we expected to identify distinct patterns of concerns, since they may differ between groups of people, according to the age group, the type of habitation and also the income. Another goal of this survey was to increase the list of interesting questions with the suggestions of the inquired subjects. This is relevant since it allows us to be aware of the people's real concerns that we could have missed. We also expect to know what are the preferred forms of visualization for each type of information.

5.1.3.2 User Profiling

This survey was submitted to the general public, since the energy consumption is a problem that concerns us all. However, some profiling methods were applied when constructing this questionnaire, in order to retrieve some specific information while performing the results analysis.

The first was to determine if the inquired person is encharged of paying the energy bills, because the people who do it are more likely to have different concerns than the other who do not. The other form of profiling the user, comes from the information about his household, since the number, the age and house type of residents can have an influence in the answers. The final method of profiling is by asking if the inquired person has already a system for Home Automation and if this system allows energy savings, which allows to understand the necessities that these people have yet to be solved.

5.1.3.3 Survey Construction

Constructing this survey had to take into account some aspects that were mentioned by Barbara Kitchenam, such as the development phases [50] that are the Goal Definition, the Design and the Testing. The first phase consists in the definition of the purpose, in this case, the validation of the most relevant information about the energy consumption. The design phase is divided in two parts: the survey layout and the formatting rules, where some attention was given for it to be as

[35] http://www.sogosurvey.com/
[36] http://www.mitportugal.org/

perceptible as possible and the type of questions, which were chosen to be the most adequate to each question [51]. Besides these concerns, this phase of development had also taken into account the aspect of being resilient to bias [52].

Before publishing the survey it was necessary to test it, so a test group was formed where the questions were verified and reviewed. Having these phases concluded, the final version of the survey was organized in four parts: Personal Information; Useful Questions; Questions Validation and Other Useful Questions.

The Personal Information retrieves information about the inquired person to be used for user profiling. In this topic the following question were asked:

- If the payment of the bill is their responsibility
- The number of persons living in the household
- The age group of each person living in the household
- The house type
- If the inquired has some kind of home automation
- If the inquired has some kind of consumption sensor, other than the provided by the electric company

The next part is the Useful questions. This part was created to remove the bias, in other words, it was asked to submit at least three questions that concerns the inquired, about his energy consumption. This way, the person is obligated to think beforehand about his needs, and only then the questions gathered in the meetings are presented.

This leads us to the third part, where some of the questions presented in the Table 5 were submitted to validation. The validation of each question is divided in three questions: their usefulness, the ideal form to view the question result and the hardness to accomplish this question with his current system. An example of the validation of the question *"Which categories of equipment has higher consumption?"* would be:

Is useful to you to know which categories of equipment has higher consumption?

Without interest		Indifferent		Very Useful
0	1	2	3	4
O	O	O	O	O

How would you prefer to see which categories of equipment has higher consumption?

O In a house plant, showing the category in each room
O Just the equipment of each category
O In a table, showing each equipment of each category
O Other: _____

Can you current system show the categories of equipment that has higher consumption?

Unable to		Hardly		Easily
0	1	2	3	4
O	O	O	O	O

Figure 9 – Validation of "Which categories of equipment has higher consumption?"

Since it was unpractical to validate all of the questions, it was chosen some of the questions that would illustrate the concerns of the eight areas (Table 5). Therefore, fourteen questions were

selected and as shown, these questions were more focused in the Consumption by Appliance, since it was an area that was poor in the presented EMS (Section 3.2):

- What is the actual consumption (Consumption by period of time)?
- How much will I pay in the end of the month (Tariff)?
- What was the evolution of consumption (Consumption Evolution)?
- What is the consumption of each space (Consumption by Space)?
- Which equipment are in stand-by mode (Consumption by Appliance)?
- Which spaces have consume more in the peak period (Consumption by Space)?
- Which equipment are consuming more than they should (Consumption by Appliance)?
- Which equipment are working when they are not needed (Consumption by Appliance)?
- Which appliances are using programs with unnecessary power (Consumption by Appliance)?
- Which equipment may be damaged (Consumption by Appliance)?
- Which categories of equipment has higher consumption (Consumption by Category)?
- Which equipment should be replaced (Notifications)?
- It is useful to receive warnings about equipment that are working when they are not needed (Notifications)?
- It is useful to receive Warnings about equipment that are consuming more than they should (Notifications)?

To end the survey, it was once more asked the inquired subjects to think about questions about their energy consumption that were not referred before. The purpose of this question was to confirm if there were more concerns about the energy consumption that were not mentioned in the survey.

5.1.3.4 Results Analysis

This survey was available online for about three months and managed to collect a total of 76 participants. As mentioned earlier this survey was submitted to the contacts provided by the MIT Program, which represents 14.47% of the participants. From the remainder, 63.45% are from the social networks and 22.08% from the contacts of the involved authors, including both campus.

5.1.3.4.1 Validation of initial hypothesis

Considering the initial hypothesis, mentioned in the Section 5.1.3.1, some findings were obtained. From Figure 10, was possible to conclude that the preferred way to represent the information is the graphical form. This statement comes from the answers about the ideal form to view the question result. From all of 76 inquired persons, more than 50% answered that they prefer to see the data in a graphical form, perhaps because it is the best way to immediate check the values. This graphical form combines both the results view by charts and by house plant, depending on the question.

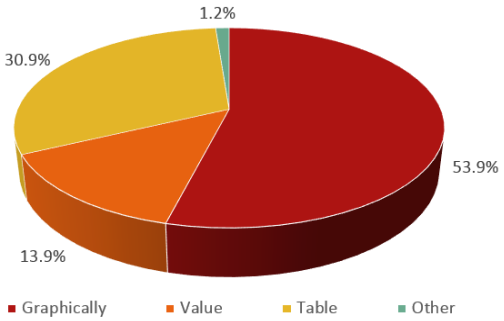


Figure 10 – Prefered way for visualizing the results

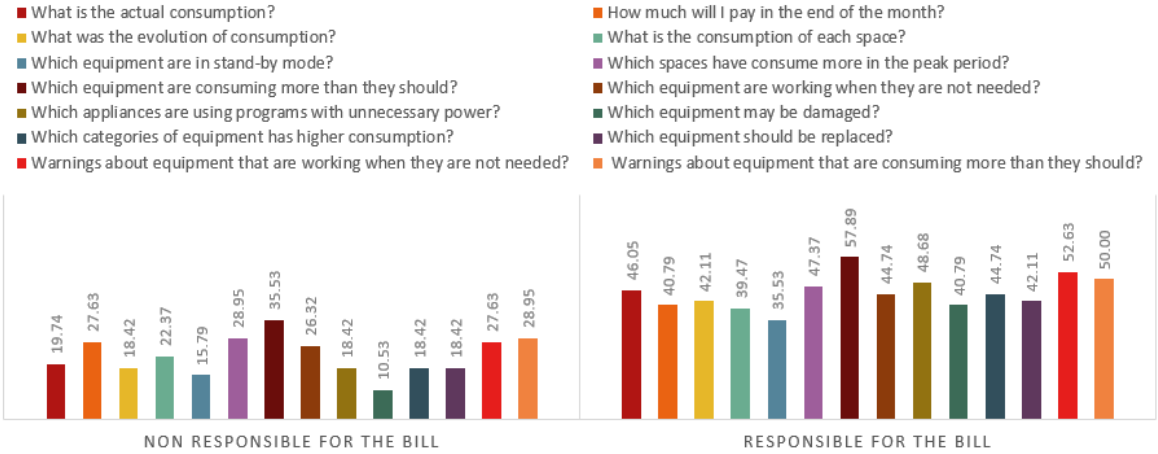


Figure 11 – Relevant Questions for people who pays and do not pays the bill

According to the Figure 11 it was possible to retrieve the questions with more relevancy for the two types of respondents, which are people paying and not paying the bill. Both of them consider relevant to know which equipment is consuming more than it should and also to be alerted about it. However, the first group considers that is important to receive notifications about the equipment that are working when they are not needed, while the second prefers to know about the spaces consumption in peak period.

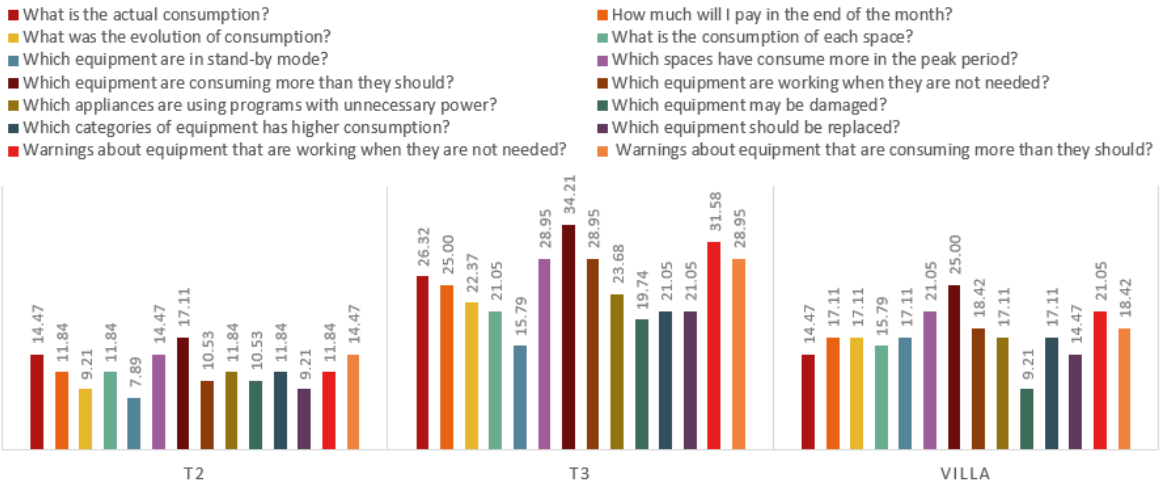


Figure 12 – Relevant Questions for people with different types of household

Other of the control questions was the type of the household, however, we only consider the three more relevant house types, because they represent more than 80% of the answers. These house types are T2 (two bedroom apartment), T3 (three bedroom apartment) and villas. For these three types, it is more relevant to know which equipment are consuming more than they should. They also value the spaces with more consumption in the peak period as an information of interest and also, the notifications that help to reduce the consumption. As can be seen in the Figure 12.

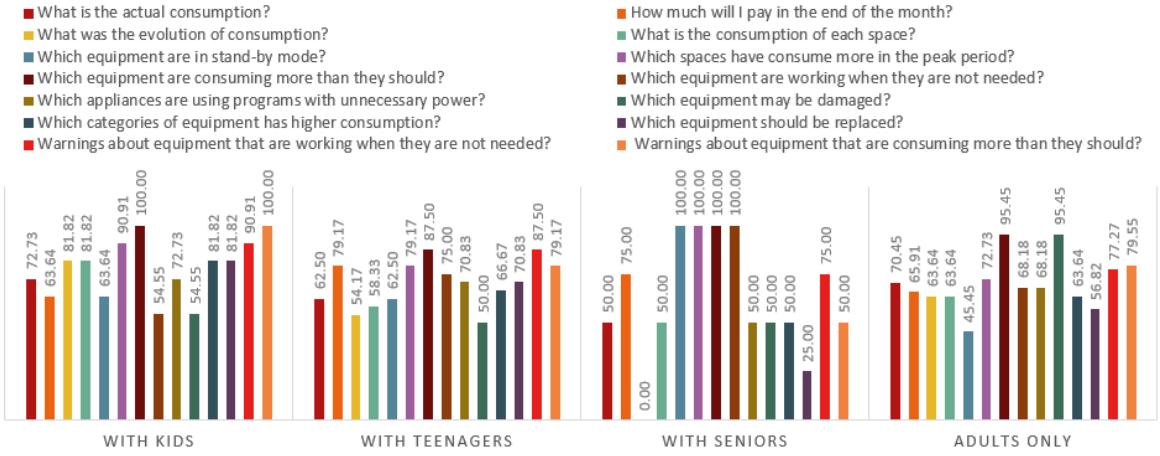


Figure 13 – Relevant questions for the different age groups

As mentioned, the household can have several age groups, which can influence the way that people consume energy. From our survey we identified four group: Household with kids; Household with teenagers; Household with seniors and Household with adults only. As shown in Figure 13, all of these groups have in common concerns about the equipment consuming more than it should and also the spaces with consumption in peak period. Both type of notifications are also appreciated, which means people wanted to be alerts about their consumption. Particularly, there is one main difference between the household with adults only, which is the identification of the equipment that can be damaged.

This survey also managed to fulfill the goal of complement the areas of information, mentioned earlier. The Table 6 emphasize these new additions. The first one is new sub-topics that were

added to the Consumption Evolution, letting the user retrieve, from the historical data, the value of energy that could have been saved and the percentage of energy that was used efficiently. It was also added two new areas, Consumption Comparison and Consumption by Activity. Consumption Comparison allows the user to compare the consumption between equipment or between houses, while the Consumption by Activity permits the user to control the consumption of each activity, such as vacuuming or making a cake. The Consumption by appliance also suffered the joining of a new sub-topic, since it was necessary to have the consumption during a period of time. The final addition was two sub-topics to the area of notification, because some questions about the failures of energy during the day were asked, as some question about the uses that the spent energy could have.

Table 6 – Areas of Information after the Survey

Areas	**Topics**
Consumption by period of time	Consumption values from different periods
Consumption Evolution	Consumption comparison with historical values
	The efficient use of energy
	*Possible sa*vings
Consumption Comparison	Equipment consumption comparison
	House consumption comparison
Average Consumption	The average consumption values from different periods
Consumption by Activity	The amount of energy consumed during a activity
Consumption by Space	The total amount of energy consumption in a room
Consumption by Category	The total amount of energy consumed by a group of equipment
Consumption by Appliance	Consumption of a specific equipment
	Identification of the unnecessary working equipment
	Indication of abnormal consumptions
	Identification of energy waste
	Consumption of a specific equipment during a period of time
Tariff	Tariff Comparison
	Budget Setting
	Cost Preview
Notifications	Warning Information
	Energy Failures
	Other uses of the spent energy
	Saving Suggestions

As can be seen, the brainstorming had managed to collect an interesting group of relevant areas of information, however, it was not enough to reflect all of the user's real concerns. Therefore, with this survey, was possible to receive the public opinion and finished the relevant areas of information.

5.1.3.4.2 Query Patterns

In the survey construction, before determine the relevancy of the gathered questions in the brainstorm, it was included a part where the respondents propose questions, with the intent to reflect their concerns about energy consumption. From a total of 76 respondents, we managed to obtain 251 questions, where 224 were considered valid for the purpose of this study. The Figure 14 represents a chart with these questions' pattern, in other words, the questions cataloged by area of information. As can be seen the areas that most worries the respondents are the Consumption values from different periods and the Consumption by equipment, followed by the Consumption by Space. The following images presents the user concerns on each of the areas sub-topics.

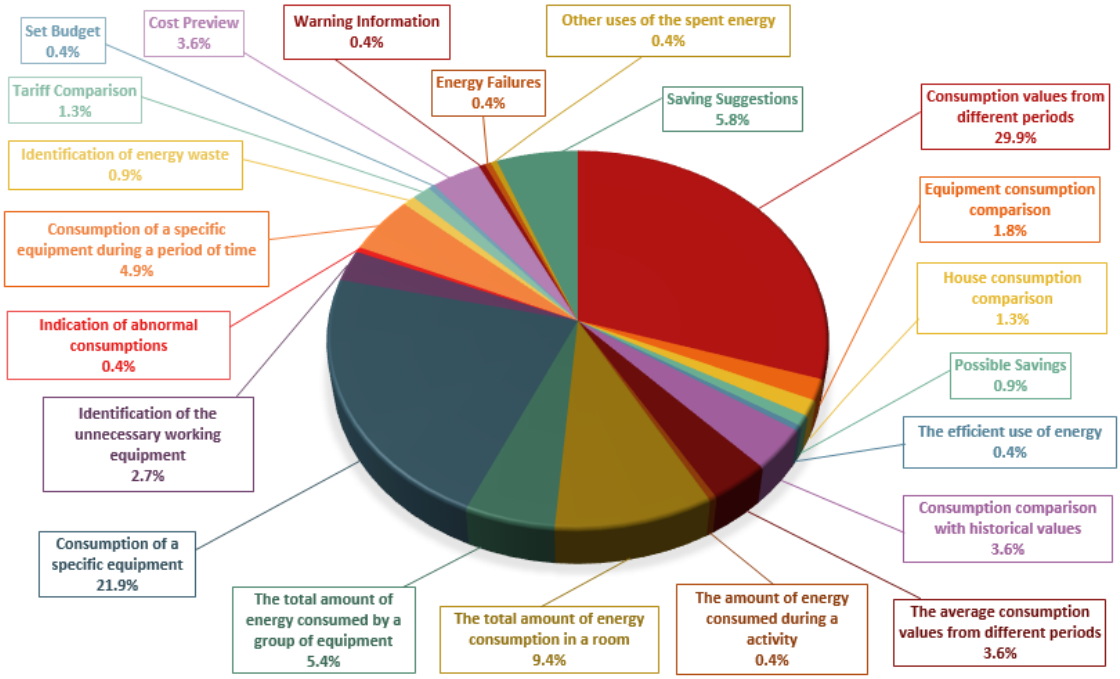


Figure 14 – Respondents concerns by area of information

5.1.3.5 *Discussion*

Considering the results presented and the patterns of questions that are the most relevant, we can confirm three important facts (Figure 15): (1) each respondent is unique; (2) consumption by period is significant, but not as much as the consumption by equipment and (3) notifications are also a central part for the users, since it helps them to be informed.

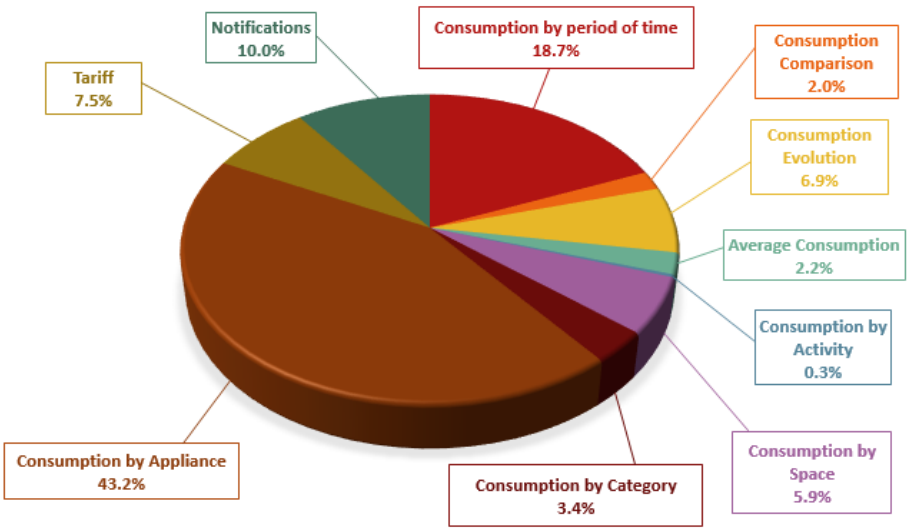


Figure 15 – Respondents concerns by area of information

5.1.3.5.1 Opportunities and Applicability

The accomplishment of this survey was significant to clearly identify the energy patterns and the areas of information relevant to residents. With this information, it is possible not only to understand the questions that the users want to answered, but also to design new systems and languages that are more oriented towards the collection of patterns gathered in this research. Consequently, the development of interfaces can be more focused on what the user wants and on the aspects that he needs to improve at home, to decrease his energy consumption.

5.1.3.5.2 Threats to validity

We have identified two threats to the results of our survey. The first one is the fact that people consider the majority of the questions as very useful. This consists in a problem, since we cannot ascertain if the questions really matter to the respondents or if they were psychologically influenced when completing the survey.

The other threat lies in the domain explored by this survey, since this domain of energy saving can be misunderstood with similar domains. This was clear when analyzing the questions' patterns, where several questions suggested by the respondents were more focused in the domain of Building Automation, such as having a system that turning off the lights when there the room is unoccupied.

5.1.3.5.3 Surprises

In the construction of the survey we asked questions from the majority of the areas of information. The answers to these questions had the expected result, however some new areas, or sub topics, were uncovered. In the open answer question, which was placed in the beginning and ending of the survey, we managed to understand that some information was not included in our gathered areas of information. Themes like Energy Comparison, which allows the user to compare the consumption between equipment and between houses; Consumption by Activity, where the user can register activities and understand their consumption; Notifications, since it would be useful to some users understand their energy failures and other used to the energy that was efficiently used and wasted.

5.1.4 Categorization of the query elements

Having the most relevant areas of information uncover and the question that most concerns the home users (Table 6 and Figure 14), it was achievable to define the necessary elements for constructing a query. These elements (Figure 16) are divided in five groups: Action, Subject, Characteristic, Condition and Timeline.

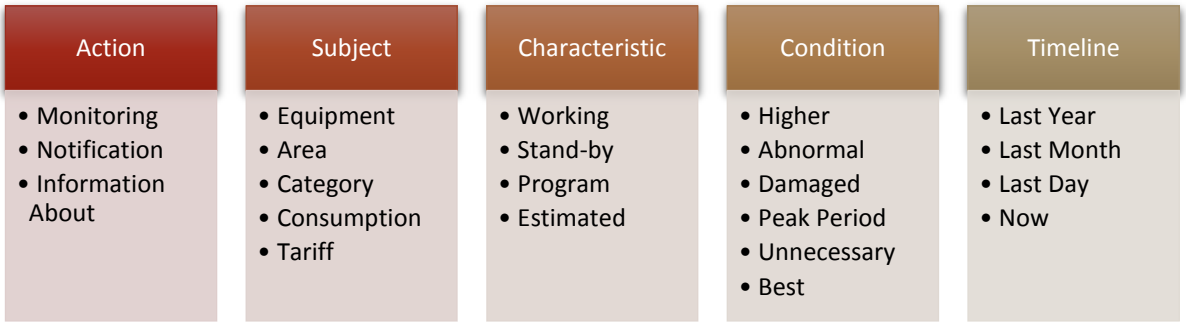


Figure 16 – Categorization of the query elements

The first group allows the user to determine the purpose of the query. In the current version it is allowed three types of actions. The Monitoring action is to be used when we want to observe a subject during a certain amount of time, for instance the room equipment's consumption from the last month. Notification was created to produce alerts through e-mail or text about a certain type of consumption, such as the use of unnecessary equipment and Information About has the goal to provide suggestion or information about the consumption. For example, the estimated value to pay or the best energy tariff.

The Subject describes who is going to be watched in the selected action. The Equipment refers to all house equipment, while the Area gives information about the different rooms in the house and the Category gets the information of all the appliances in each group. The Consumption can be used in order to give the information about total consumption or a specific one if associated to the Equipment, Area or Category.

The third group is used to specify the wanted subject. The Working characteristic filters all of the equipment by the ones that are or were functioning, while the Stand-by outline the ones that are using the stand-by mode and Program indicates that we are interested in equipment's programs, such as the one in a dishwasher. The last one is used to determine the Estimated value of the next month bill.

The Conditions are similar to the Characteristic, since they are also work as filters. The Higher allows to define the number of results wanted sorted from higher to lower. Abnormal gives all the equipment's consumption with unusual values, while Damaged presents the ones that may be broken. The Peak Period condition selects only the ones that have consumptions during the most expensive period, whereas the Unnecessary shows all the equipment that are working when they are not necessary, like having the lights on in a room that is unoccupied. The last is Best, which is used to find the most adequate tariff for the house consumption.

Finally, the Timeline allows to define the time interval for the results visualization. The available intervals are the Last Year, Last Month, Last Day and Now.

5.1.5 Definition of language models

Having identified the terms necessary to construct a query, the next phase is the definition of the language characteristics and the relation between the components required to build a question. As a result, two models were elaborated, the Features Model and the Domain Model.

5.1.5.1 Features Model

The Features Model express the elements that are mandatory and optional, and also their variety in the query creation. As can be seen in the Appendix 9.2, a query has two main components, the query terms and a visualization form, which can be Pie, Bar or Table.

From the terms that constitute a query there are two mandatory and three optional. The mandatory are the Action and the Subject, where one action has to be selected to indicate the query goal, but there can be more than one subject that are under the selected action.

The optional terms in the query building are the Timeline, the Characteristic and the Condition. The Timeline term is optional given the fact that is only used with the Monitoring action, while the Characteristic and the Condition are filters that specifies the chosen subject.

5.1.5.2 Domain Model

The Domain Model represents the conceptual model that describes the definition of the DSL. As shown in the Appendix 9.3, it characterizes the relation between elements and the way that they are connected. The main components are the ones present in the Features Model, however, in this model two new elements appear, the Connection and Container.

The Connection is divided into two different types, the Term Connection and the View Connection. The first one is responsible for connection the query elements, while the second is responsible for associating a visualization form to the query.

The Container is the query holder, meaning that he desired question has to be build inside of it, supporting all of the query element. The visualization form, since it is applied to the all query is place in the outside of the Container and connected to it thought a View Connection

5.2 DESIGN

With the Domain Analysis completed the next step is the definition of the abstract and concrete syntax. The first one is the definition of the language metamodel, which describes the language structure and elements relationships, and the second one is the images chosen to be graphical representation of the elements.

In this stage is also gathered the incompatibilities between elements that are then enforce in the Implementation phase.

5.2.1 Definition of the abstract syntax

In order to define the language abstract syntax was used the Ecore and EMF models. The language metamodel, present in the Appendix 9.4, represents all the different elements, their attributes and relations that were identified in the Domain Model described in the Section 5.1.5.2.

The Figure 17 presents a fragment of the Ecore Model with the main elements to form a query.

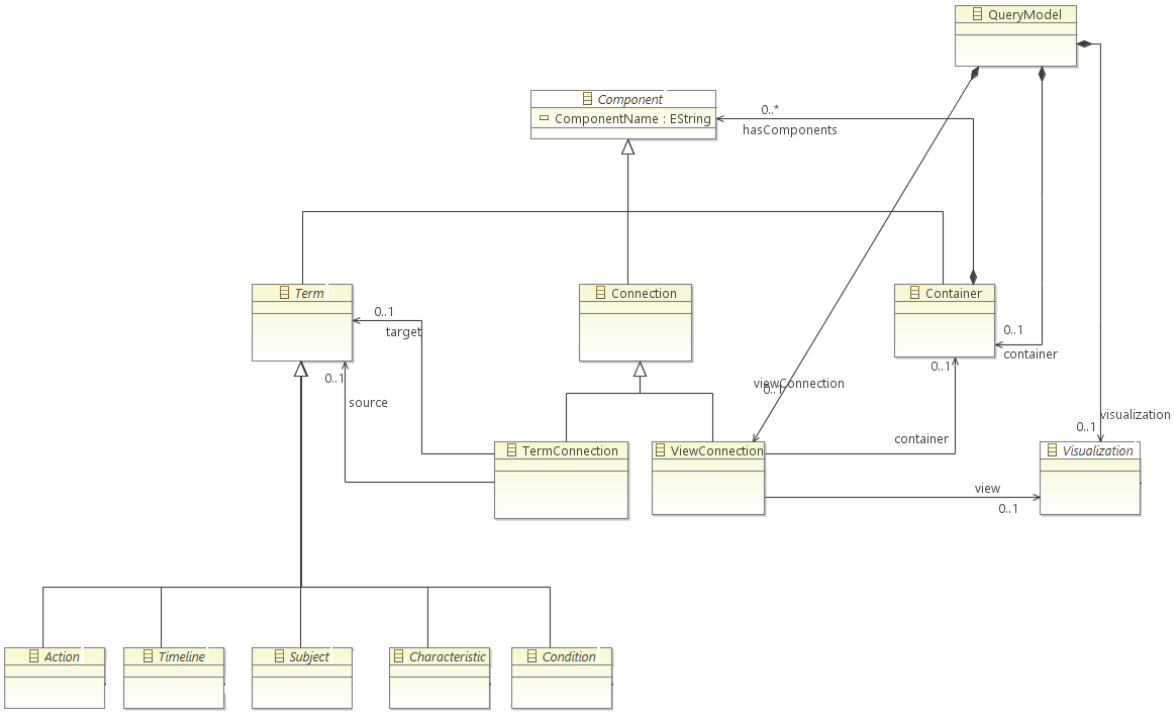


Figure 17 – Fragment of the language metamodel

5.2.2 Definition of the concrete syntax

The visualization is an important part in software engineering [53]. In this language this is particularly essential, since it is more intuitive and simple for the user to relate the icon to the element function [54]. In this language the elements were divided into eight groups, where five of them are the terms necessary to build questions. The Figure 18, represents the first three groups: Objects, Connections and Visualization.

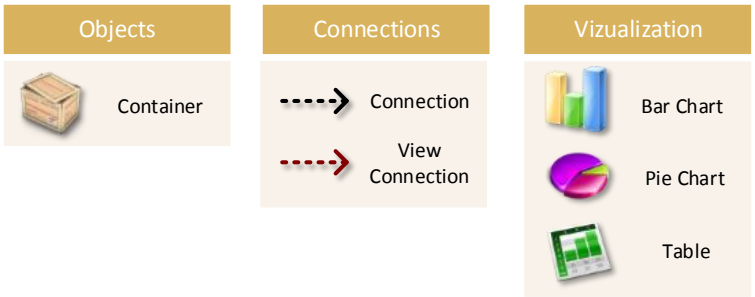


Figure 18 – Support Elements

These groups are responsible for assisting in the definition of questions. The Container assignment is to outline the setting of the query to be constructed. This setting will allow a View Connection of a specific Visualization form chosen by to user to analyze the results, whereas the Connection objective is to connect the different terms that compose a query.

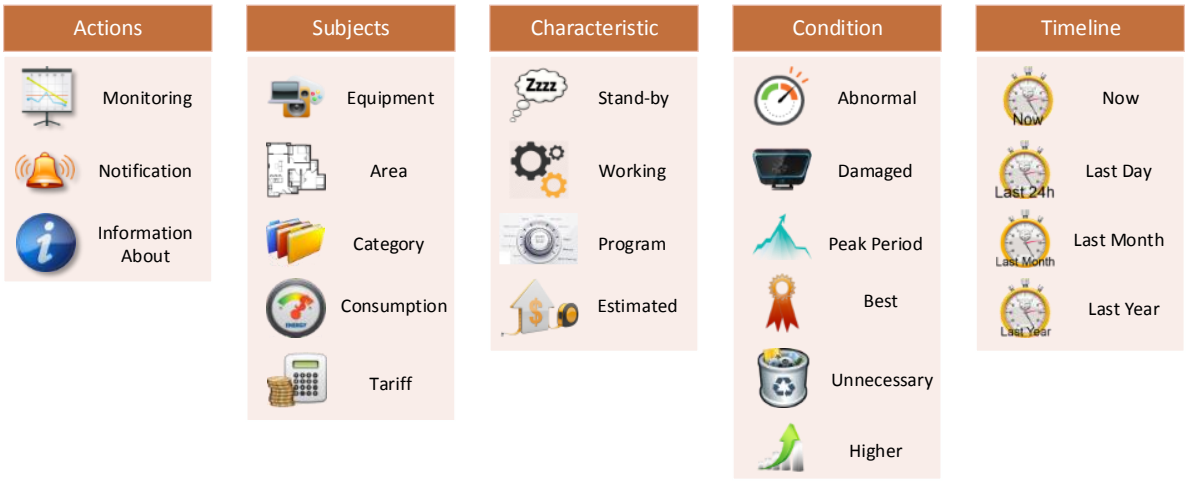


Figure 19 – Query Elements

Represented in the Figure 19 are the visual representations of the query elements, which were presented in the above section. The selection of these images to symbolize their significant was carefully done, however, want the same symbol may stand in for different meanings between two different persons. Given this reason, was necessary to include them in the Usability Test that will be described in the Section 6. With this evaluation it will be possible to understand if they are semantically immediate, semantically opaque or semantically perverse [54].

5.2.3 Elements Incompatibilities

In the definition of the language metamodel several compatibility problems between elements were detected, due to the fact that some elements cannot be combined together. The following table presents the all the found incompatibilities that have to be controlled in the implementation phase.

Table 7 – Elements Incompatibilities

Element	***Inconsistency***	***Description***
Container	Inner Container	The container cannot a hold another container
	Container Name	The container has to have a name
	Query Action	A query needs one Action
	Only One Action	A query can only have one Action
	Query Timeline	The container can only have one Timeline
	Query Subject	The container has to have at least one Subject
	Visualization Form	A visualization form is required to show the query results

Element	*Inconsistency*	*Description*
Term	Isolated Element	All the query element must be connected
Monitoring	Monitoring Timeline	The Monitoring Action requires the usage of a Timeline element
Notification	Notification Timeline	The Notification does not need a Timeline, since it works on real-time
	Unnecessary Visualization Form	The Notification Action does not require a visualization form
Information About	Information About Timeline	The Information About Action does not oblige the usage of a Timeline element
	Tariff Subject	The Information About Action currently can only be used with the Tariff Subject
Stand-By	Stand-By Compatible Subject	The Stand-by Characteristic is only usable with the Equipment Subject
Estimated	Estimated Compatible Subject	The Estimated Characteristic is only usable with the Tariff Subject
Working	Working Compatible Subject	The Working Characteristic is only usable with the Equipment Subject
Program	Program Compatible Subject	The Program Characteristic is only usable with the Equipment Subject
Higher	Higher Compatible Subject	The Higher Condition is only usable with the Consumption Subject
Best	Best Compatible Subject	The Best Condition is only usable with the Tariff Subject
Damaged	Damaged Compatible Subject	The Damaged Condition is only usable with the Consumption Subject
Abnormal	Abnormal Compatible Subject	The Abnormal Condition is only usable with the Equipment and the Consumption Subject
Bar Chart	Bar Compatible Elements	The Bar Chart is only possible for queries involving the consumption of Areas, Categories, Equipment and Stand-by Equipment
Pie Chart	Pie Compatible Elements	The Pie Chart is only possible for queries involving the consumption of Areas, Categories, Equipment and Stand-by Equipment

5.3 IMPLEMENTATION

The Domain Analysis and the Design phases were necessary to define the domain concepts, features and the language metamodel. With these steps concluded it was possible to move through the actual language implementation.

In order to build the EQuery DSL, it was necessary to split the work into four task. The first one is the establishment of the Ecore and EMF file, which are responsible by the definition of the rules and elements from the models mentioned in the Design phase. The second task is the definition of rules

for the connection of elements, for correct query definition. Subsequently comes the parsing from the constructed DSL to the actual SQL query and finally, the communication with the MySQL and Esper Database.

5.3.1 Ecore and EMF Models

As previously mentioned, the Ecore and the EMF are used to represent all the different elements, their attributes, relations, concrete syntax and visualization rules, being then available for the generation of the corresponding Java implementations classes [55].

The Listing 1 presents the definition of the element Higher, a type of Condition, in the EMF file. In this declaration is defined the element concrete syntax and properties, which are the element Name and the default number of results to be seen with higher consumption. The complete file can be found in the Appendix 9.5.

Listing 1 – Definition of the Higher Condition on the EMF

```
@gmf.node(label="Name", figure="figures.Higher", label.icon="false", label.placement="external",
    tool.small.bundle="Thesis_DSLv3.edit",
    tool.small.path="/icons/full/obj16/Higher.png")
class Higher extends Condition {
  readonly attr String Name = "Higher";
  attr int Results = 10;
}
```

The Figure 20 represents the same Condition but defined in the Ecore file.

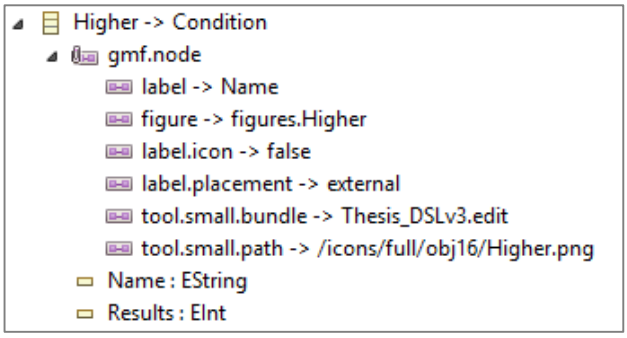


Figure 20 – Definition of the Higher Condition on the Ecore

5.3.2 Well-formed Rules

To prevent the user of creating erroneous questions it was necessary to define some rules. These rules were identified in the Section 5.2.3 and their purpose is to alert the user for some misplaced element or for elements that cannot be combined. For this purpose, it was used the Epsilon Validation Language (EVL), which main objectives are to evaluate constraints on metamodels, checking their intra and inter-model consistency and giving some quick forms to repair their inconsistencies [39]. The complete set of rules is present in the Appendix 9.6, however, the following Listing presents the implementation of one rule that states that in the current prototype, the action Information About can only be used with the Tariff subject.

Listing 2 – EVL rule that states that the Information About action can only be used with the Tariff subject

```
constraint justForTariffs {

    check : (Subject.allInstances.size == 1 and
            Subject.allInstances.exists(t|t.isKindOf(Tariff)))

    message : 'The Action Information About can only be used with the Tariff Subject'
}
```

5.3.3 Language Target Semantics

For converting the questions from the concrete syntax to the SQL code, it was necessary the use of the Epsilon Generation Language (EGL). EGL is a specific language for transforming a model into text, being this text, executable code, reports, or images [13] [14] [19].

In the Listing 3 is possible to see the main EGL file that produces the SQL query for the MySQL. This file resorts to another file in order to retrieve each peach that is necessary to produce a correct SQL query.

Listing 3 – Formulation of a SQL query by the main.egl

```
SELECT [%=getSelectBlock(container)%]
FROM [%=getFromBlock(container)%]
WHERE [%=getWhereBlock(container)%]
[%=getGroupByBlock(container)%]
[%=getOrderByBlock(container)%]
[%=getLimit(container)%];
```

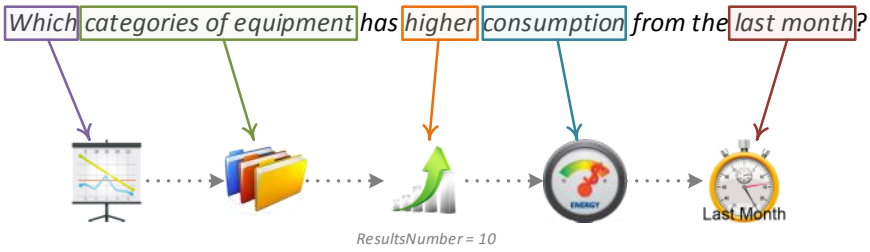
To get this pieces each function reads the several elements that constitute, returning the code associated to it. The Listing 4 present the function that return the group by block.

Listing 4 – Function that defines the group by option in a SQL query

```
[% operation getGroupByBlock(container : Container) : String {
    var groupBy : String = "";
    for (t in container.hasComponents.select(c|c.isKindOf(Term))){
            if(t.isKindOf(Equipment)) {
                groupBy = "GROUP BY equipment.ID";
            }
            if(t.isKindOf(Area)) {
                groupBy = "GROUP BY area.ID";
            }
            if(t.isKindOf(Category)) {
                groupBy = "GROUP BY category.ID";
            }

    }
    return groupBy;
} %]
```

Both of these files are present in the 9.7. Nevertheless, for better understanding of this conversion the Figure 21 presents parsing of the same questions from text to code. In this figure each color represents an element used to construct the wanted questions. Seeing that this example, pretends to perform a monitoring from the last month consumption it will be accessing the stored data in the MySQL database. In the Notification case, the question is converted to Esper.




```
SELECT category.name as Category,getCategoryKwh(category.ID, (now() - INTERVAL
1 MONTH)) as KWh, getCategoryKWhPrice(category.ID, (now() - INTERVAL 1 MONTH))
as 'Cost (€)'
FROM equipment INNER JOIN equipment_category ON (equipment.ID = equipID) INNER
JOIN category ON (category.ID = catID) INNER JOIN historic.activity ON
(equipment.ID = equip)
WHERE historic.activity.start >= (now() - INTERVAL 1 MONTH)
ORDER BY Cost DESC
LIMIT 10;
```

Figure 21 – Conversion example from the concrete syntax to SQL code

5.3.3.1 EQuery Semantics

The purpose of this section is to demonstrate, with more detail, the language transformation from the queries defined in the concrete syntax to the SQL code necessary to present the results to the user. This section displays a few samples in four sets: the consumption by a period; the consumption of a subject; specific queries that can be made and notifications.

5.3.3.1.1 Energy consumption in a specific period

One of the functionalities that the EQuery provides is values of the global consumption from a desired period. The Figure 22 presents an example where is show the last year consumption in a bar chart. This is possible by selecting the Monitoring action, the Consumption subject and the desired timeline.

Figure 22 – "What was the last year consumptions?" in the EQuery

The following listing presents the same question but in the SQL syntax.

Listing 5 – "What was the last year consumptions?" in MySQL

```
SELECT historic.consumption`.consumption as Consumption
FROM historic.consumption`
WHERE date >= (now() - INTERVAL 1 YEAR)
```

5.3.3.1.2 Identification of the subject consumption

The previous question can be more specific if added another subject. For instance, the Figure 23 presents a question where the five areas with higher consumption are retrieved, however, the subject Area could be replace by the one of the others. The Higher condition could also be removed for the question for returning all the equipment.

Figure 23 – "Which areas have higher consumption?" in the EQuery

The Listing 6 shows the same query in its SQL representation.

Listing 6 – "Which areas have higher consumption?" in MySQL

```
SELECT area.name as Area, getAreaKWh(area.ID, (now() - INTERVAL 1 DAY)) as KWh,
getAreaKWhPrice(area.ID, (now() - INTERVAL 1 DAY)) as 'Cost (€)'
FROM equipment INNER JOIN myhouse.area ON (area = area.ID) INNER JOIN
`historic.activity` ON (myhouse.equipment.ID = equip)
WHERE `historic.activity`.start >= (now() - INTERVAL 1 DAY)
GROUP BY area.ID
ORDER BY Cost DESC
Limit 5;
```

5.3.3.1.3 Specific questions

In this section it is presented six questions that have very specific purposes. Each question has its representation in the EQuery and in SQL.

Which spaces have consumed more in the peak period?

For executing this question it is necessary to sum the consumption of the different appliances in each room, since the aim is to determine the consumption of individual spaces. However, by performing this question the consumption of each space will be filtered by the ones that occurred during the peak period. The usage of the Higher element allows to define the number of spaces with higher consumption to be visualized.

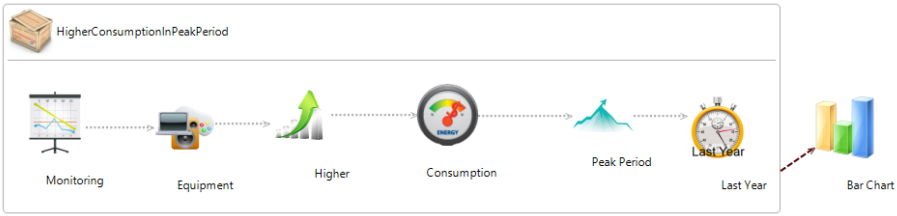


Figure 24 – "Which appliances have consumed more in the peak period?" in the EQuery

Listing 7 – "Which appliances have consumed more in the peak period?" in MySQL

```
SELECT equipment.name as Equipment, area.name as Area,`historic.activity`.start,
`historic.activity`.end, getPeakKWh(equipment.ID, (now() - INTERVAL 1 YEAR), 1) as
KWh, getPeakKWhPrice(getPeakKWh(equipment.ID, (now() - INTERVAL 1 YEAR)) as 'Cost
(€)'
FROM equipment INNER JOIN area ON (area = area.ID) INNER JOIN `historic.activity`
ON (equipment.ID = equip) INNER JOIN tariff_schedule ON (tariff_period =
tariff_schedule.ID)
WHERE typeOfPeriod = 'P' AND (`historic.activity`.start >= (now() - INTERVAL 1
YEAR))
GROUP BY equipment.ID
ORDER BY Cost DESC
Limit 10;
```

Which equipment are in stand-by mode?

With this question, it is possible to retrieve the consumption made by the equipment in the last month during their functioning in stand-by mode. The result is the amount of consumption (kWh) and money these equipment wasted in the past month in this mode.

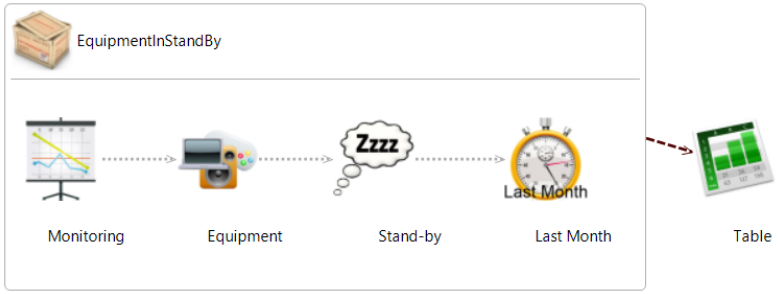


Figure 25 – "Which equipment are in stand-by mode?" in the EQuery

Listing 8 – "Which equipment are in stand-by mode?" in MySQL

```
SELECT equipment.name as Equipment, area.name as Area,
getEquipmentKWh(AVG(consumption), equipment.ID, (now() - INTERVAL 1 MONTH), 0) as
KWh, getEquipmentKWhPrice(equipment.ID, (now() - INTERVAL 1 MONTH), 0) as 'Cost
(€)'
FROM equipment INNER JOIN area ON (area = area.ID) INNER JOIN
`historic.unused_equipment` ON (equipment.ID = equip)
WHERE `historic.unused_equipment`.start >= (now() - INTERVAL 1 MONTH)
GROUP BY equipment.ID
```

Which equipment may be damaged?

This question is compares the equipment's consumption with their expected consumption to retrieve the ones that may be damaged. The transformation of this query is the same for the abnormal consumptions.

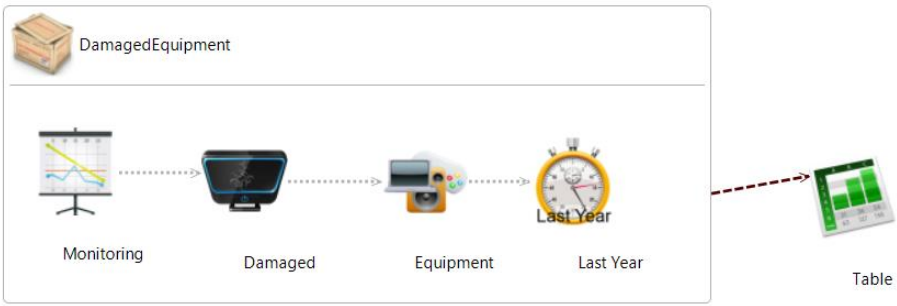


Figure 26 – "Which equipment may be damaged?" in the EQuery

Listing 9 – "Which equipment may be damaged?" in MySQL

```
SELECT equipment.name as Equipment, area.name as Area, expectedConsumption as
'Expected Consumption', standbyConsumption as 'Expected Stand-by Consumption',
consumptionAvg as 'Average Consumption', consumptionPeak as 'Consumption Peak',
`historic.equipment_consumption`.date as Date
FROM myhouse.equipment INNER JOIN area ON (area = area.ID) INNER JOIN
`historic.equipment_consumption` ON (equipment.ID = equip)
WHERE expectedConsumption IS NOT NULL AND (((consumptionAvg > (expectedConsumption
+ (expectedConsumption/4))) OR (consumptionAvg < (expectedConsumption -
(expectedConsumption/4)))) OR (standbyConsumption IS NOT NULL AND ((consumptionAvg
> (standbyConsumption + (standbyConsumption/4))) OR (consumptionAvg <
(standbyConsumption - (standbyConsumption/4)))))) AND (`historic.activity`.start >=
(now() - INTERVAL 1 YEAR))
GROUP BY equipment.ID
ORDER BY consumptionAvg DESC
```

What is the best pricelist for my energy consumption?

This question consists in providing the information about the most adequate tariff for the user consumption. In order to give this information, it is necessary to simulate the user's average consumption in each available tariff. This will be then presented to the user sorted from the most economical to the most expensive.

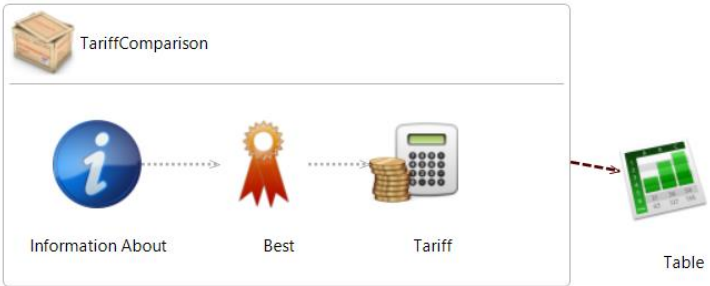


Figure 27 – "What is the best tariff for my consumption?" in the EQuery

Listing 10 – "What is the best tariff for my consumption?" in MySQL

```
SELECT company as Company, type as Type, averagedCost(ID, 3) as 'Cost per Month
(€)', selected as 'Select Tariff'
FROM Tariff
WHERE powerKVA = (SELECT powerKVA FROM Tariff WHERE selected = 1)
ORDER by averagedCost(ID, 3)
```

How much will I pay in the end of the month?

The result of this question is the average consumption of the last months in every tariff available from each company. Due to complexity of this query in SQL was necessary to create a function to perform the all query.

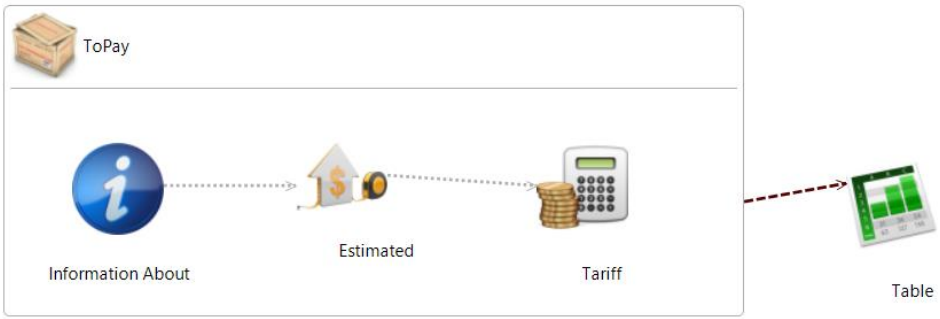


Figure 28 – "How much will I pay in the end of the month?" in the EQuery

Listing 11 – "How much will I pay in the end of the month?" in MySQL

```
SELECT getEstimatedConsumption() as 'Value to Pay (€)'
```

5.3.3.1.4 Warnings

This final section present an example of a Notification for equipment that working when they are not needed. Currently the notification only deals with the presence sensor, which means that the user is alerted, through the designated form (SMS or E-mail), when some appliance is working and there is no presence in the room.

Figure 29 – "Warning about equipment that are working when they are not needed?" in the EQuery

Listing 12 – "Warning about equipment that are working when they are not needed?" in Esper

```
SELECT equipName, areaName
From EquipmentConsumption, Occupation
Where divisionID = area and isOccupied = false
```

5.3.4 Language Architecture

To summarize the developed language allows the user to define a query in the EQuery Eclipse Editor, which can be from three different actions. Each one of this action interacts in a different form with the stored data to generate the necessary SQL code. In order for this generation to work was necessary to have a Relational Database Management System (RDBMS) and a Data Stream

Management System (DSMS). The Figure 30 represents the connection between the EQuery, the RDBMS (MySQL), the DSMS (Esper) and the information provider (Emoncms), which was implemented in one of the Research Laboratories from *Faculdade de Ciências e Tecnologia.*

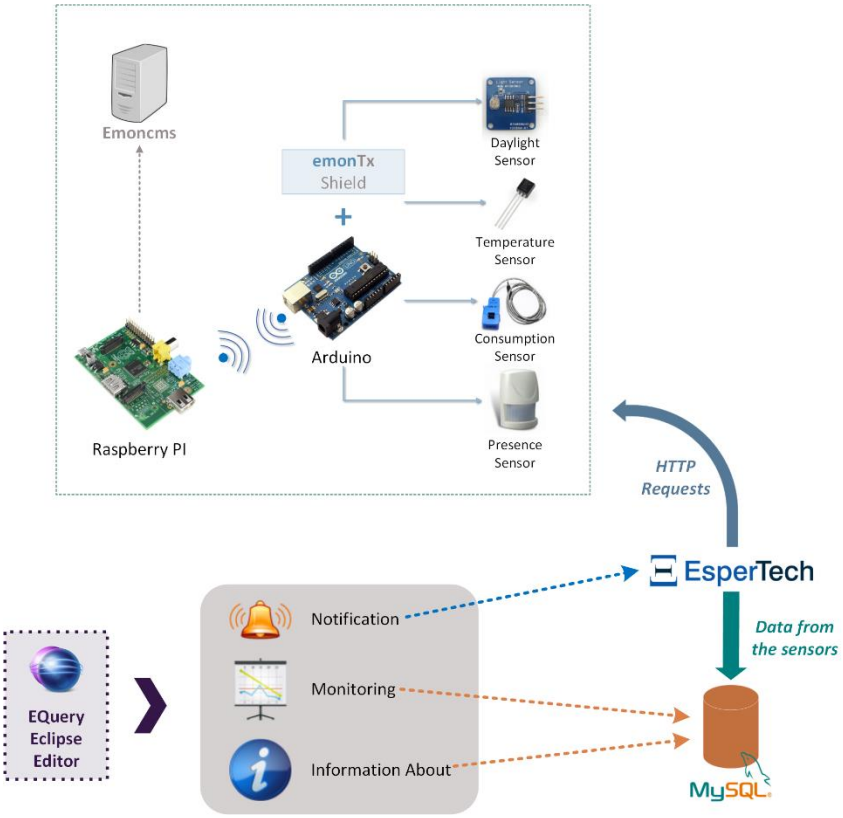


Figure 30 – Language Architecture

The Emoncms is the web application offered by the OpenEnergyMonitor, mentioned in the section 3.2.2, that allows the definition of data feeds for the information that is being retrieved from the sensors. This information is then retrieved by the Esper through HTTP requests for real-time processing, being then passed to the MySQL for historic storage.

The Entity-Relationship Model that describe the MySQL database can be found in the Appendix 9.8 and is divided in three types of information: structure, historic and support. The structure Information is used to establish the home configuration, in other words, the different rooms, appliances and categories. The data provided by the Esper belongs to the historic and in the support information is stored the knowledge that is commonly known, such as the tariff prices and periods, that are necessary for answering the user question.

The execution of the users question is performed according to the query action, if it is a Monitoring or an Information About action the query is performed on the MySQL and is converted as seen in the above section. However, if it is a Notification is necessary to registers it in the Esper. In order to register the Notification is necessary to create a Statement with the wanted query and the selected Listener, which would send a SMS or an email. The Figure 31 presents a Notification example.

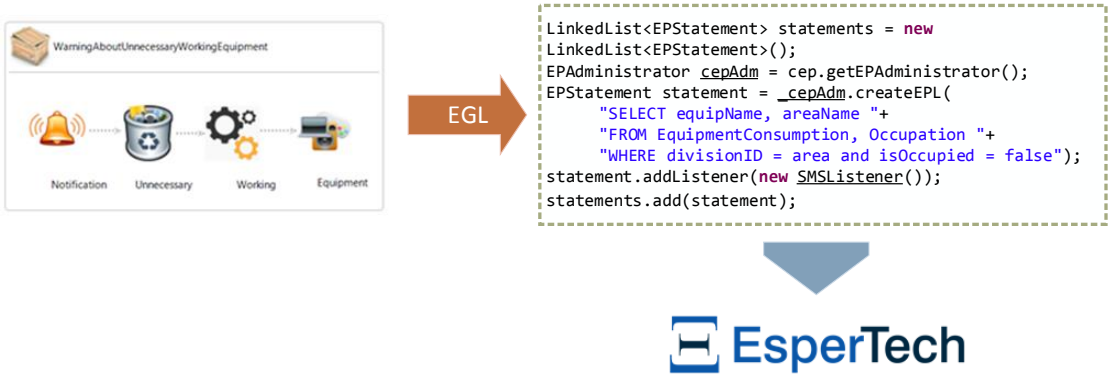


Figure 31 – Definition of a Notification in Esper

The Figure 32 presents the EQuery Eclipse Editor that allows the definition of queries. The editor is composed by four parts: the drawing area, the palette, the properties and the output generation. The first part is where the user constructs his query with the help of the palette, which have all the necessary elements. The properties area is used to give more detail, in this case, to define the number of results. The last one is responsible for the transformation of the query to SQL.

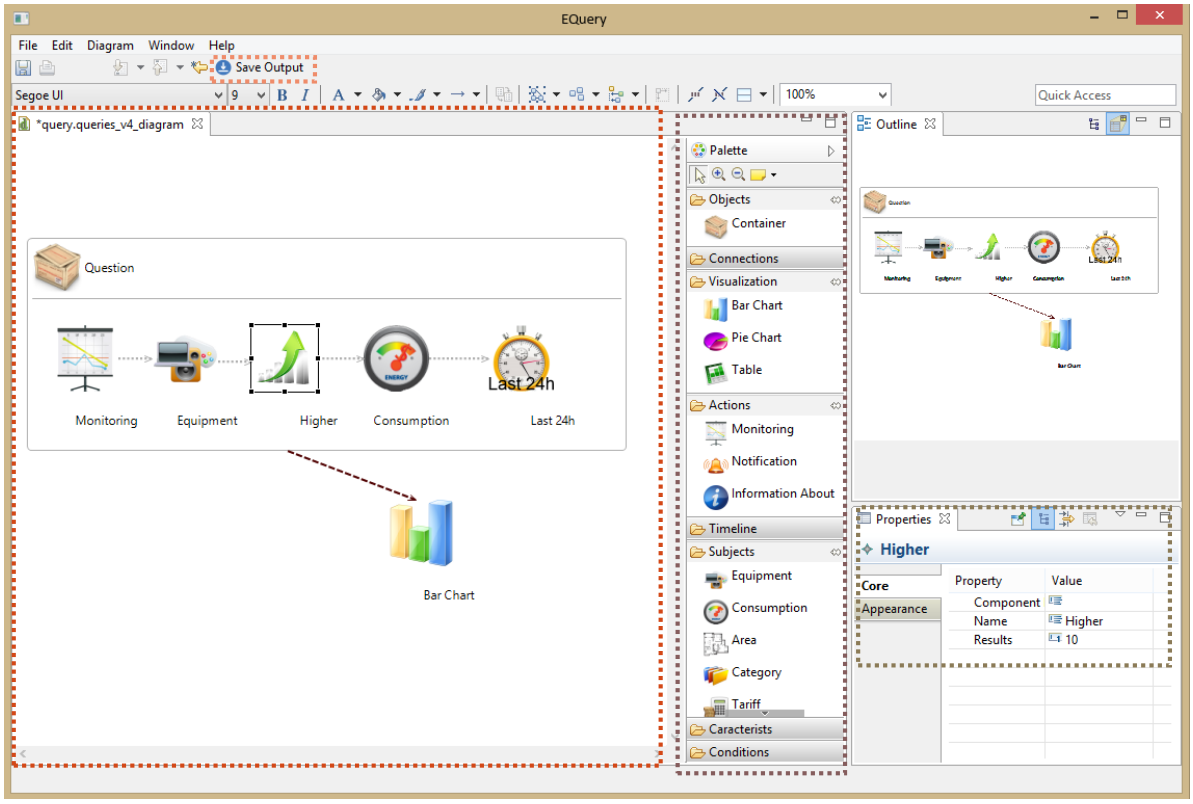


Figure 32 – EQuery Eclipse Editor

6

EQUERY VALIDATION

According to the development process of a DSL, mentioned in the Section 2.2.2, the final step in each iteration is the language validation. With this phase it is possible to determine the language weakness and correcting them in the next iteration, making the language sturdier and comprehensive.

The validation of the develop language is composed by two parts. The first part is the evaluation of the language expressiveness, which is based on the areas of information collected in the Domain Analysis. The second part of the validation consists in usability tests performed with three types of real users, with the purpose to detected problems in the query definition, as well as the user efficiency.

6.1 LANGUAGE EXPRESSIVENESS

With the completion of the EQuery Language it was necessary to establish the language level of expressiveness. For this purpose it was created the Table 8, based on the Table 5 from the Section 5.1. The choice of this table was due to the fact that was created with the assistance of domain experts, containing the relevant questions from the different areas of information.

Table 8 – Language Expressiveness of the pre-established queries (The symbol ○ indicates that is unavailable and the ● that is available in the DSL.)

Areas	***Questions***	***Accomplished***
Consumption by period of time	*What is the actual consumption?*	●
	What was the last day consumption?	●
	What was the last month consumption?	●
	What was the last year consumption?	●

Areas	*Questions*	*Accomplished*
Consumption Evolution	*What was the evolution of consumption from the last two years?*	○
Average Consumption	*What is my daily consumption?*	○
	What is my monthly consumption?	○
	What is my annual consumption?	○
Consumption by Space	*Which spaces have the higher consumption?*	●
	What is the consumption of each space?	●
	Which spaces have consumed more in the peak period?	●
Consumption by Category	*Which categories of equipment has higher consumption?*	●
Consumption by Appliance	*What is the consumption of a specific equipment?*	○
	Which appliances have a higher consumption?	●
	Which equipment are in stand-by mode?	●
	Which equipment are working when they are not needed?	●
	Which equipment may be damaged?	●
	Which equipment are consuming more than they should?	●
	Which appliances are using programs with unnecessary power?	●
Tariff	*What is the best pricelist for my energy consumption?*	●
	For how much did I exceeded my target budget?	○
	How much will I pay in the end of the month?	●
Notifications	*Warning about equipment are in stand-by mode*	●
	Warning about equipment that are working when they are not needed?	●
	Warning about equipment that are consuming more than they should?	●
	Which equipment should be replaced?	○

Due to the language extensibility by the possibility to combine several elements the EQuery only managed to define about 70% of the questions, where their graphical representation is presented in the Section 5.3.3.1. Although being considered important, the missing 30% of the questions were not implemented due to their complexity, since they require specific information, or due to their lack of relevancy to the user as seen in the Figure 14.

6.2 USABILITY STUDIES

Having the Language expressiveness determine, the following step in the validation of the EQuery Language is it usability by real user. Therefore it was conducted a usability test to comprehend the details that may have been overlooked, as well as the user opinion regarding the language usefulness. Thus, all of the language aspects must be evaluated, which includes the following points:

1. Evaluation of the concrete syntax;
2. Identification of the difficulties in the query construction;
3. Comparison of the query construction difficulties using an Excel Spreadsheet as alternative;
4. Readability assessment;
5. Language comprehension;
6. Willingness to use the EQuery

Owing to the fact that the user's performance has also to be evaluated, some metrics from the ISO 9241-11 were added. These metrics are the effectiveness, the efficiency and the user satisfaction, where the effectiveness corresponds to the number of correct questions, the efficiency to the task duration and the satisfaction to participant's opinion.

6.2.1 Evaluation Process

The execution of this usability test was performed in three sessions, where two were located in the *Faculdade de Ciências e Tecnologia* and the one in the *Instituto Superior Técnico*, having a total of 12 participants, with an average duration of 60 minutes. From these 12 participants it was possible to extract three groups of diverse people: Beginners, Domain Experts and DSL Users. The first group was composed by students from a different field of study, being unusual for them to deal with this kind of language. Inserted in the second group were some of the students that participate in the brainstorming, mentioned in the Section 5.1.2, and some that were not involved, but that are knowledge of this domain. The last group was composed by users and developers of Domain Specific Languages (DSL).

These three session were accomplished following the same approach, depending on the four steps shown in the Figure 33.

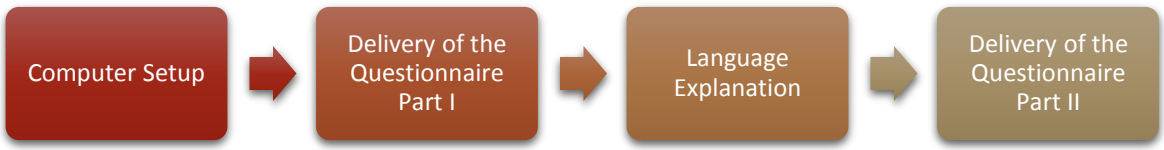


Figure 33 – Evaluation Procedure

The first step consisted in the preparation of each computer, which consisted in the installation of a recording software, the EQuery language and the Excel Spreadsheet with a sample of energy consumption data. After concluding this step, was asked to the participants to answer the Part I of the questionnaire, which is detailed in the Section 6.2.1.2. Once finishing this Part I, a brief explanation about the language is given, through a set of slides with examples, followed by the Part II of the questionnaire.

6.2.1.1 Presentation

With the intent to simplify the language explanation a set of slides were prepared to be showed between the Part I and II of the questionnaire. The first idea presented in these slides is the language purpose, which is showing examples of the most used energy questions that are useful in the daily routines. Having the language goal settled, the next notion was the existing element necessary to build a question, finishing with the presentation of some examples of queries constructed in the EQuery.

6.2.1.2 Questionnaire

As mentioned in the above section, the usability test is structured in two parts. The Part I of the usability test has the objective to retrieve information about the user that is realizing the test, acquiring the following information:

- General Information (Age, Sex and Country);
- Education (Education level and Field of Study);
- Experience in Databases (SQL) and in Spreadsheet (Excel);
- The bill payment responsibility and monitoring of the energy consumption.

This part of the test also ascertains the expressiveness of each icon, mentioned in the concrete syntax (Section 5.2.2). To do this the user can give to each image the classification of Immediate Association, Logical Association or No Association, which corresponds to being Semantically Immediate, Semantically Opaque or Semantically Perverse [54].

In the Part II the user is challenged perform different activities. The first one is to create three questions, with increasing levels of difficulty, in the EQuery Language and in an Excel Spreadsheet, specifying its difficulty and certainty of being correct. With these questions is possible to compare the two tools for gathering information about the energy consumption in the household. The second activity is to write the questions being asked in the displayed figures, which is helpful for determining the level of readability. The final task inquires if it is possible to perform the three presented questions, with the objective to comprehend if the user understood the extent of the language.

The last two questions are about the user satisfaction using the EQuery Language and his ideas for replacing the concrete syntax images that he considered poorly choose.

The main reason for dividing the usability test in two parts was to not influence the participant regarding the concrete syntax. This way, it was possible to retrieve the participant's opinion before introducing the language.

6.2.2 Results Analysis

The subjects that have participated in this study have an average age of 24, having different education level and experience with the tools like the SQL and Excel. The following table presents the profiling of the 12 participants, group by their classification.

Table 9 – Information about the Usability Test subjects

		Sex		Education Level			Experience Average	
	Total	Male	Female	Secondary	Bachelor	Master	SQL	Excel
Beginner	5	2	3	4	0	1	Basic	Medium
Domain Experts	4	4	0	0	2	2	Medium	Basic
DSL Users	3	3	0	0	2	1	Medium	Basic

All of these participants had an individual opinion about the images used in the concrete syntax, as can be seen in the Figure 34. However, all agreed that the images for Bar Chart, Pie Char and Area are the ones that semantically immediate, which means that infers its meaning simply by looking at it. In order to retrieve the semantically opaque and semantically preserve was considered the images with six or more votes. Therefore, the Container, the View Connection, the Monitoring and the Program were deliberated as semantically opaque, meaning that they are associated to its meaning in a subjective form. The one only one that is thought to be semantically perverse, giving a different meaning than expected, is the Damaged.

Having this information is possible to understand that these images have to be improved. Nevertheless, it was also considered poorly choose the one that have six or more votes when joining the semantically opaque and perverse, which includes the Higher and the Unnecessary.

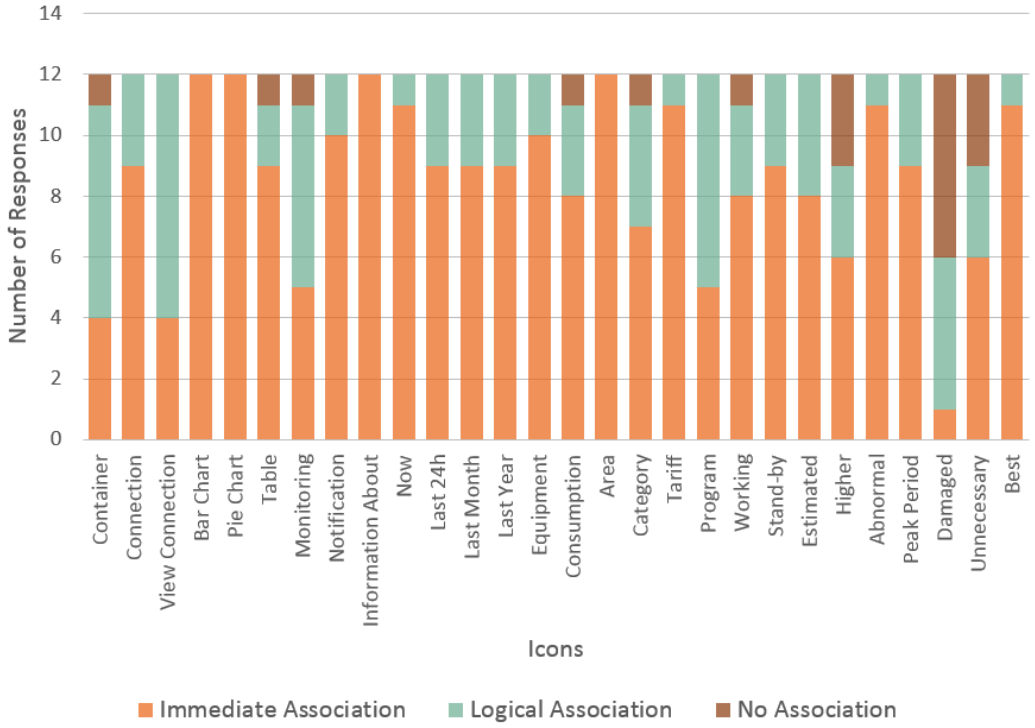


Figure 34 – Concrete Syntax classification by the participants

In the second part of the Usability Test the subject perform three question in the EQuery Language and using an Excel spreadsheet. The Table 10 and Table 11 presents the statistic from these two tools. The first column indicates the average time that a participant needed to complete the task, while the second shows the time variability, in other words, revels how closely the participant time is around the average time. The third and fourth column shows the number of participants that had formed the correct questions and the respectively percentage. The last two columns indicate the number of participants, and corresponding percentage, that were able to answer correctly and within the average time, thereby determining a successful performance.

From the Table 10 is possible to conclude that the most difficult task was the Task 2, since only 42% of the participants were able to produce a completely correct question and merely 17% were able to perform it within the average time, this is, in an efficient way. Although the Task 1 is the most successful, with 92%, only 33% accomplish it within the average time, seeing that it was the first contact that the subjects had with the language. Despite the majority of the subjects agree that the Task 3 was the most difficult, the Task 2 was the one with less successful answers.

Table 10 – Statistics from the participants' effectiveness and efficiency, using EQuery Language in the tasks elaboration

Queries in the DSL	Time Average	Standard Deviation	Correct Answers	% of Correct Answers	Correct Answers within the time benchmark	% Correct Answers within the time benchmark
Task 1	4.32	1.29	11	92%	4	33%
Task 2	4.50	2.58	5	42%	2	17%
Task 3	5.39	3.38	10	83%	9	75%

With the Table 11 and Table 12, it is possible to conclude that the implementation in an Excel Spreadsheet is more complex than using the EQuery Language, since only two subject were able to perform the first question correctly. Due to the complexity of dealing with several information spread by several spreadsheets the subjects started to give up in line with the task's difficulty.

Table 11 – Statistics from the participants' effectiveness and efficiency, when using an Excel Spreadsheet to perform the given tasks

Queries in the Excel	Time Average	Standard Deviation	Correct Answers	% of Correct Answers	Correct Answers within the time benchmark	% Correct Answers within the time benchmark
Task 1	7.85	4.58	2	17%	2	17%
Task 2	11.51	6.48	0	0%	0	0%
Task 3	6.18	2.85	0	0%	0	0%

Table 12 – Participants withdraw statistics

Queries in the Excel	Total of participants that abandoned the Task	% of participants that abandoned the Task	Average Trying Time
Task 1	6	50%	4.60
Task 2	7	58%	2.38
Task 3	8	67%	1.39

The last part of the test had the purpose of gathering the language level of readability, which was achieved through an exercise where the participants had to write the question presented in two images. The Figure 35 shows a pie chart with the level of correctness in the query identification, that defines the language level of readability. As can be seen, in the first chart all of the participants successfully recognized the question exposed. However, in the second chart, 8.3% have committed an interpretation mistake, while the other 8.3% had only the idea right.

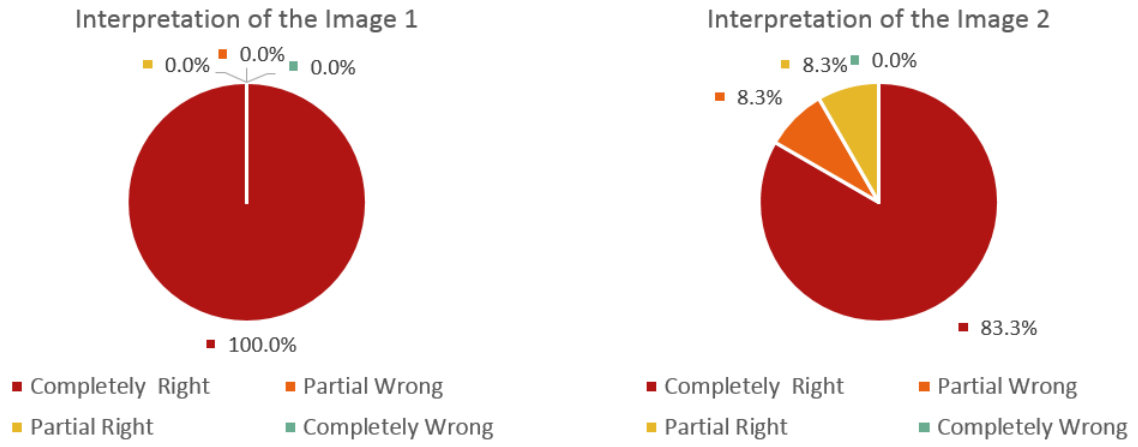


Figure 35 – EQuery readability chart

6.2.3 Discussion

The execution of this Usability Test was very helpful to recognize the weakness and strength in the developed language. From the participants' feedback it was possible to distinguish two vulnerabilities, where the first one is the language flexibility, since the query elements can be connected in different ways to produce the same question. The second is the misunderstanding between the action Monitoring and Information About. Although these ones, in the results analysis was also identified that some images of the create syntax are not very meaningful for the users.

As strengths, it was discovered that the EVL rules (Section 5.3.2) were useful in the questions construction, teaching the user to not make the same mistake in another question. Subsequently, the participants agreed that this tool would be advantageous to monitoring the energy consumption in a house, since it was easier and more intuitive than the most common approaches, the usage of a Spreadsheet. As can be seen in the Table 13, the questions were well performed by the different groups, even by the beginners, which have no training in this field.

Table 13 – EQuery statistics by group

	Total of Elements	% of Elements	Correct Answers Task 1	% Correct Answers Task 1	Correct Answers Task 2	% Correct Answers Task2	Correct Answers Task 3	% Correct Answers Task3
Beginners	5	41.67%	4	80%	1	20%	3	60%
Domain Experts	4	33.33%	4	100%	2	50%	4	100%
DSL Users	3	25.00%	3	100%	2	67%	3	100%

6.2.3.1 Treats to Validity

With the execution of this usability test, three threats to the results have been identified. The first one is the fact that five of the participants are from a different country, which means that due to the language barrier some concepts might not have been completely understood, leading also to more explanation from the test monitors. The second threat is about the user feedback, since the participants may hold back in the flaws detected in the language, in order to provide good opinions to the developers. The last detected threat involves the subject concentration, because two of the session were performed in the afternoon, interrupting the activities that were being accomplished.

7

CONCLUSION

This thesis purpose was the development of a Domain Specific Language (DSL) that allows the creation of queries about the energy consumption in a household. In order to begin the DSL development, the first step was the study of the existing solutions for performing queries and for monitoring the energy consumption (Section 3). The solutions for query definition were considered inadequate, because we are targeting every kind of user and these solutions require some training on database technologies. Subsequently, the studied EMS are already very informative, but are limited in providing information about the equipment, areas and category consumption. Additionally, these systems are formatted to monolithic query wizards, which treats all the users in the same way. This is a serious limitation in expressiveness, which can be confirmed by the existing research demonstrating that each user has different concerns.

In order to understand the users' actual concerns about their consumption of energy, an online survey was performed (Section 5.1.3). With this survey it was possible to uncover the relevant areas of information, and also the most problematic areas for the user, which are the Consumption by equipment and the Consumption by Space.

Having recognized the significant questions about the energy consumption, it was possible to determine the elements needed to define the language metamodel and the implementation of the EQuery Language (Section 5.3). With the purpose of determining the language's weakness and improvement potential, a validation phase was conducted with real users to evaluate the efficacy, effectiveness and satisfaction.

7.1 CONTRIBUTIONS

With the development of this thesis it was possible to elicit user centered query patterns and categorize them. To achieve this, we have performed a Domain Analysis (Section 5.1), which included a brainstorming with domain experts and the publishing of an online survey, where several question were gathered allowing the identification of elements present in a query.

We were also able to develop a language that in its first version allows for the construction of roughly 70% of the collected questions and that, according to usability tests (Section 6), is usable by both programmers and non-programmers.

7.2 FUTURE WORK

The domain explored in this thesis is a recognized problem in the modern society, mainly due to the population growth and to the scarcity of fossil fuels. Therefore, as we have seen, there is a necessity for systems and tools that empowers the home user with awareness mechanisms, which can help to reduce the consumption and help in the protection of the environment.

Given the domain extension, we would like to perform a more complete validation phase involving more subjects. The objective is to perceive if there exists more flaws in the language, motivating another iteration on the design language development phases. The finality is to correct the imperfections that were eventually detected and to allow the formulation of questions in all the gathered areas of information.

Furthermore, it would also be interesting to extend the language using the same approach to facility managers. This would expand the language domain to buildings, institutions, and universities, where the energy management is more complicated and the concerns are more field specific.

8

REFERENCES

[1] C. Fischer, “Feedback on household electricity consumption: a tool for saving energy?,” *Energy Effic.*, vol. 1, no. 1, pp. 79–104, Feb. 2008.

[2] J. M. Abreu, F. C. Pereira, and P. Ferrão, “Using pattern recognition to identify habitual behavior in residential electricity consumption,” *Energy Build.*, vol. 49, no. 0, pp. 479–487, 2012.

[3] J. E. Petersen, V. Shunturov, K. Janda, G. Platt, and K. Weinberger, “Dormitory residents reduce electricity consumption when exposed to real-time visual feedback and incentives,” *Int. J. Sustain. High. Educ.*, vol. 8, pp. 16–33, 2007.

[4] R. P. Leslie, “Capturing the daylight dividend in buildings: why and how?,” *Build. Environ.*, vol. 38, no. 2, pp. 381–385, 2003.

[5] W. Anderson and V. White, “Exploring consumer preferences for home energy display functionality,” 2009.

[6] P. Bonnet, J. Gehrke, and P. Seshadri, “Querying the physical world,” *Pers. Commun. IEEE*, vol. 7, no. 5, pp. 10–15, Oct. 2000.

[7] K. Buchanan, R. Russo, and B. Anderson, “Feeding back about eco-feedback: How do consumers use and respond to energy monitors?,” *Energy Policy*, no. 0, p. -, Jun. 2014.

[8] H. A. He, S. Greenberg, and E. M. Huang, “One Size Does Not Fit All: Applying the Transtheoretical Model to Energy Feedback Technology Design,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010, pp. 927–936.

[9] M. Voelter, *DSL Enginnering*, vol. 2, no. 65. 2013, p. 560.

[10] A. Kleppe, *Software Language Engineering: Creating Domain-Specific Languages Using Metamodels*, 1st ed. Addison-Wesley Professional, 2008.

[11] K. C. Welch and C. K. Harnett, “A review of electricity monitoring and feedback systems,” *2011 Proc. IEEE Southeastcon*, vol. 1, pp. 321–326, Mar. 2011.

[12] A. Faruqui, S. Sergici, and A. Sharif, “The impact of informational feedback on energy consumption—A survey of the experimental evidence,” *Energy*, vol. 35, no. 4, pp. 1598–1608, Apr. 2010.

[13] A. Bariic, V. Amaral, and M. Goulao, “Usability Evaluation of Domain-Specific Languages,” *2012 Eighth Int. Conf. Qual. Inf. Commun. Technol.*, vol. 0, pp. 342–347, 2012.

[14] S. Kelly and J.-P. Tolvanen, *Domain-Specific Modeling: Enabling Full Code Generation*. Wiley-IEEE Computer Society Pr, 2008.

[15] P. J. Clemente, J. M. Conejero, J. Hernández, and L. Sánchez, “HAAIS-DSL: DSL to Develop Home Automation and Ambient Intelligence Systems,” in *Proceedings of the Second Workshop on Isolation and Integration in Embedded Systems*, 2009, pp. 13–18.

[16] C. Atkinson and T. Kühne, “Model-Driven Development: A Metamodeling Foundation,” *IEEE Softw.*, vol. 20, no. 5, pp. 36–41, Sep. 2003.

[17] E. Rios, T. Bozheva, A. Bediaga, and N. Guilloreau, “MDD Maturity Model: A Roadmap for Introducing Model-driven Development,” in *Proceedings of the Second European Conference on Model Driven Architecture: Foundations and Applications*, 2006, pp. 78–89.

[18] J. Mansell, A. Bediaga, R. Vogel, and K. Mantell, “A Process Framework for the Successful Adoption of Model Driven Development,” in *Proceedings of the Second European Conference on Model Driven Architecture: Foundations and Applications*, 2006, pp. 90–100.

[19] A. van Deursen, P. Klint, and J. Visser, “Domain-specific Languages: An Annotated Bibliography,” *SIGPLAN Not.*, vol. 35, no. 6, pp. 26–36, Jun. 2000.

[20] M. Mernik, J. Heering, and A. M. Sloane, “When and How to Develop Domain-specific Languages,” *ACM Comput. Surv.*, vol. 37, no. 4, pp. 316–344, Dec. 2005.

[21] A. van Deursen and P. Klint, “Domain-Specific Language Design Requires Feature Descriptions,” *J. Comput. Inf. Technol.*, vol. 10, p. 2002, 2001.

[22] F. Hermans, M. Pinzger, and A. Van Deursen, “Domain-specific languages in practice: A user study on the success factors,” in *Model Driven Engineering Languages and Systems*, Springer, 2009, pp. 423–437.

[23] P. Gabriel, M. Goulão, and V. Amaral, “Do Software Languages Engineers Evaluate their Languages?,” *Proc. XIII Congr. Iberoam. en "Software Eng. (CIbSE’2010), Univ. del Azuay*, vol. abs/1109.6, 2011.

[24] E. Liarou, R. Goncalves, and S. Idreos, “Exploiting the Power of Relational Databases for Efficient Stream Processing,” in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, 2009, pp. 323–334.

[25] K. Ramamritham, “Real-time databases,” *Distrib. parallel databases*, vol. 1, no. 2, pp. 199–226, 1993.

[26] L. C. DiPippo and V. F. Wolfe, “Real-time databases,” *Database Syst. Handbook, Multiscience Press*, 1997.

[27] W. C. Ogden and S. R. Brooks, “Query Languages for the Casual User: Exploring the Middle Ground Between Formal and Natural Languages,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1983, pp. 161–165.

[28] V. M. M. do Amaral, “Increasing productivity in high energy physics data mining with a domain specific visual query language.,” University of Mannheim, 2005.

[29] L. Androutsopoulos, “Natural Language Interfaces to Databases - An Introduction,” *J. Nat. Lang. Eng.*, vol. 1, pp. 29–81, 1995.

[30] L. Jiang, M. Mandel, and A. Nandi, “GestureQuery: a multitouch database query interface,” *Proc. VLDB Endow.*, vol. 6, no. 12, pp. 1342–1345, 2013.

[31] P. Mäder and J. Cleland-Huang, “A visual language for modeling and executing traceability queries,” *Softw. Syst. Model.*, pp. 1–17, 2012.

[32] M. M. Zloof, “Query by Example,” in *Proceedings of the May 19-22, 1975, National Computer Conference and Exposition*, 1975, pp. 431–438.

[33] R. Baeza-Yates, G. Navarro, J. Vegas, and P. de la Fuente, “A model and a visual query language for structured text,” in *String Processing and Information Retrieval: A South American Symposium, 1998. Proceedings*, 1998, pp. 7–13.

[34] Eurostat, *Manual for statistics on energy consumption in households*. 2013, p. 170.

[35] D. S. Kolovos, R. F. Paige, and F. A. Polack, “The Epsilon Transformation Language,” in *Proceedings of the 1st International Conference on Theory and Practice of Model Transformations*, 2008, pp. 46–60.

[36] R. Wei, L. M. Rose, and D. S. Kolovos, “An Epsilon Solution to the Flowgraphs Case,” in *TTC2013*, 2013.

[37] D. S. Kolovos, R. F. Paige, L. M. Rose, and F. A. C. Polack, “Implementing the interactive television applications case study using epsilon,” in *Model-Driven Development Tool Implementers Forum, Zurich, Switzerland*, 2007.

[38] D. S. Kolovos, R. F. Paige, and F. A. C. Polack, “The Epsilon Object Language (EOL),” in *Proceedings of the Second European Conference on Model Driven Architecture: Foundations and Applications*, 2006, pp. 128–142.

[39] D. Kolovos, L. Rose, A. García-domínguez, and R. Paige, *The Epsilon Book*. 2013, p. 213.

[40] L. M. Rose, R. F. Paige, D. S. Kolovos, and F. A. Polack, “The Epsilon Generation Language,” in *Proceedings of the 4th European Conference on Model Driven Architecture: Foundations and Applications*, 2008, pp. 1–16.

[41] S. Kelly, “Comparison of Eclipse EMF/GEF and MetaEdit+ for DSM,” in *OOPSLA*, 2004.

[42] L. M. Rose, D. S. Kolovos, and R. F. Paige, “EuGENia Live: A Flexible Graphical Modelling Tool,” in *Proceedings of the 2012 Extreme Modeling Workshop*, 2012, pp. 15–20.

[43] E. Syriani, H. Vangheluwe, R. Mannadiar, C. Hansen, V. Mierlo, and H. Ergin, “AToMPM : A Web-based Modeling Environment,” in *MODELS’13 Demonstrations*, 2013.

[44] E. Syriani, “AToMPM Solution for the Petri Net to Statecharts Case Study,” in *TTC2013*, 2013.

[45] R. Mannadiar, “A Multi-Paradigm Modelling Approach to the Foundations of Domain-Specific Modelling Acknowledgments,” School of Computer Science McGill University Montreal, Quebec, Canada, 2012.

[46] R. Mannadiar, “AToMPM User’s Manual,” 2012.

[47] E. Syriani, H. Vangheluwe, and B. LaShomb, “T-Core: A Framework for Custom-built Transformation Languages,” *Softw. Syst. Model.*, 2013.

[48] I. O’Reilly Media, “Esper: Event Stream Processing and Correlation,” 2014. [Online]. Available: http://www.onjava.com/pub/a/onjava/2007/03/07/esper-event-stream-processing-and-correlation.html.

[49] D. Allen, K. Janda, and O. College, “The effects of household characteristics and energy use consciousness on the effectiveness of real-time energy use feedback: a pilot study,” in *Proceedings of the 2006 ACEEE Summer Study on Energy Efficiency in Buildings*, 2006, pp. 1–12.

[50] S. L. S. L. Pfleeger and B. A. Kitchenham, “Principles of survey research: part 1: turning lemons into lemonade,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 26, no. 6, pp. 16–18, 2001.

[51] B. A. Kitchenham and S. L. Pfleeger, “Principles of survey research: part 3: constructing a survey instrument,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 27, no. 2, pp. 20–24, 2002.

[52] B. A. Kitchenham, S. L. Pfleeger, and S. Software, “Principles of survey research part 2: designing a survey,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 27, no. 1, pp. 18–20, 2002.

[53] D. L. Moody, “The physics of notations: a scientific approach to designing visual notations in software engineering,” in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*, 2010, pp. 485–486.

[54] D. L. Moody, “The ‘physics’ of notations: toward a scientific basis for constructing visual notations in software engineering,” *Softw. Eng. IEEE Trans.*, vol. 35, no. 6, pp. 756–779, 2009.

[55] D. Steinberg, F. Budisnsky, M. Paternostro, and E. Merks, *EMF : Eclipse Modeling Framework*, Second Edi. Addison-Wesley, 2009.

9

APPENDIX

9.1 SURVEY

In this section is presented the published online survey for the gathering the home user energy concerns.

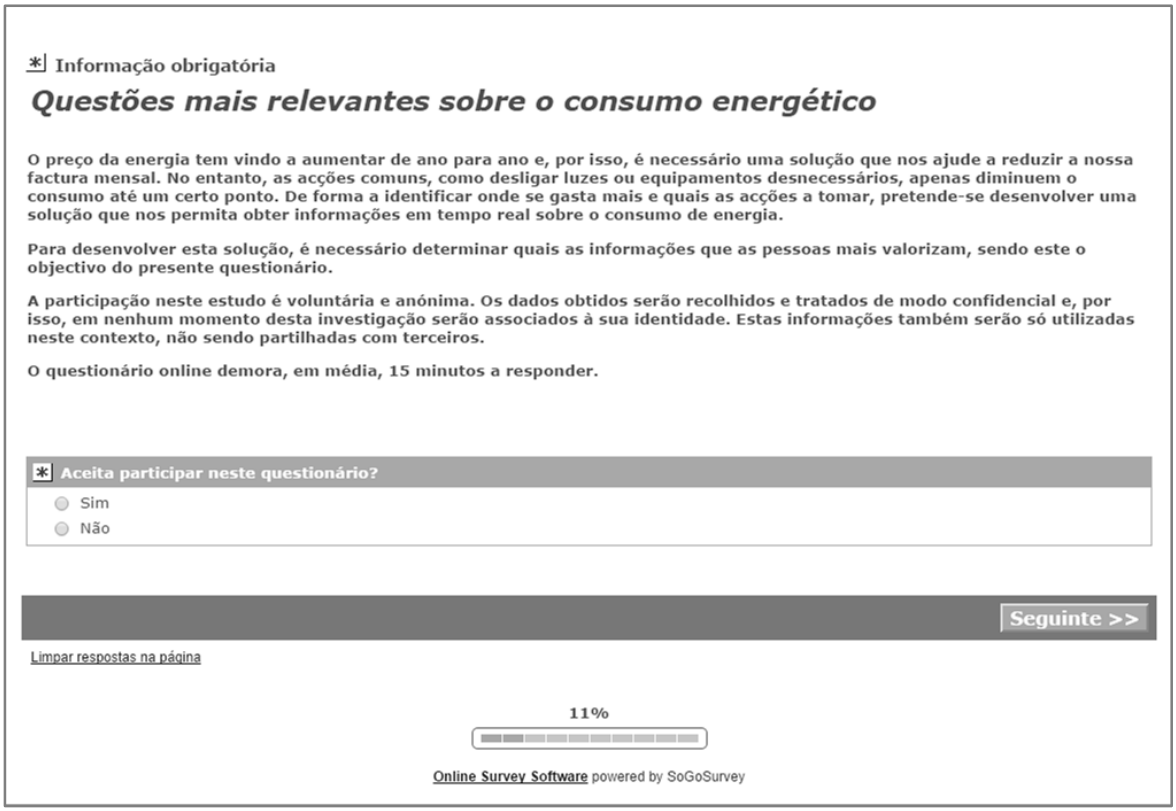


Figure 36 – Online Survey for gathering the concerns about the energy consumption (Page 1)

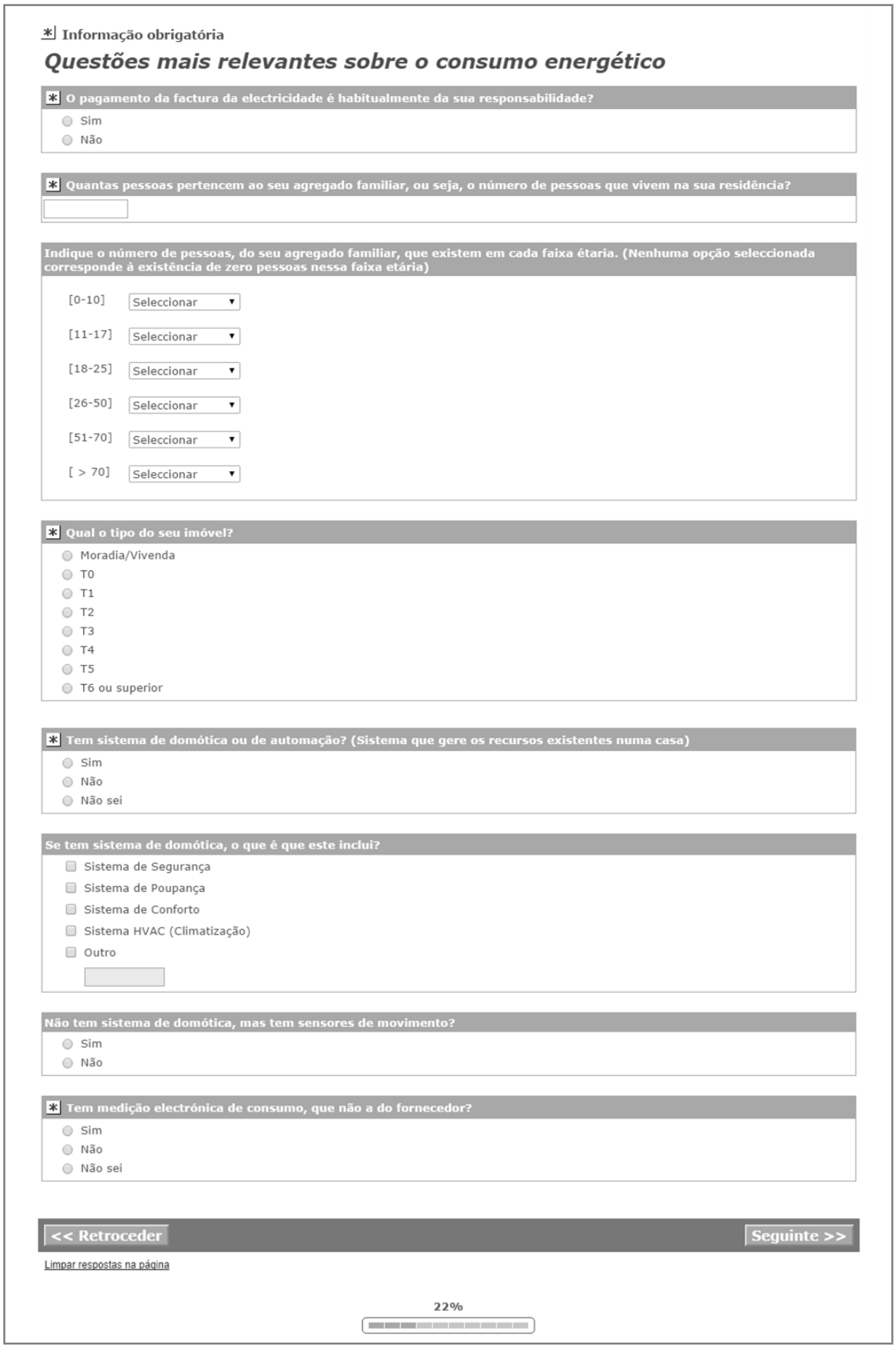


Figure 37 – Online Survey for gathering the concerns about the energy consumption (Page 2)

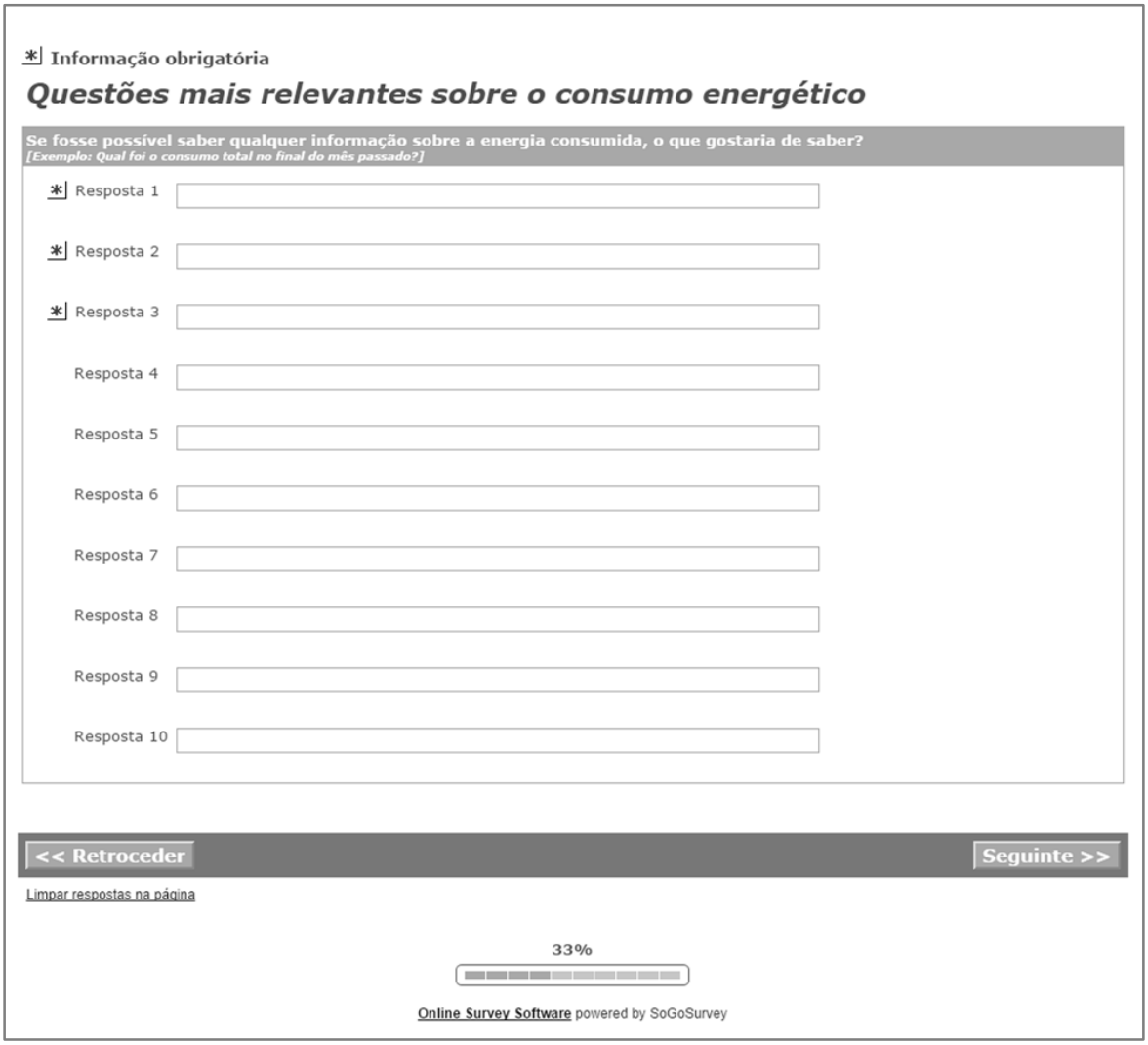


Figure 38 – Online Survey for gathering the concerns about the energy consumption (Page 3)

* Informação obrigatória

Questões mais relevantes sobre o consumo energético

O seguinte grupo de perguntas refere-se ao seu consumo actual:

*** É útil para si saber o seu consumo actual?**

Sem Interesse 0	1	Indiferente 2	3	Muito útil 4
○	○	○	○	○

*** Como gostaria de ver o seu consumo actual?**

- ○ Sobre a forma de gráfico, mostrando a evolução
- ○ Apenas o valor
- ○ Numa tabela, ilustrando também os valores anteriores
- ○ Outra forma

*** O seu sistema permite saber o seu consumo actual?**

Não consegue 0	1	Dificilmente 2	3	Facilmente 4
○	○	○	○	○

O seguinte grupo de perguntas refere-se à previsão do seu consumo:

*** É útil para si saber qual será o valor a pagar no final do mês?**

Sem Interesse 0	1	Indiferente 2	3	Muito útil 4
○	○	○	○	○

*** Como gostaria de ver o valor a pagar no final do mês?**

- ○ Sobre a forma de gráfico, mostrando a evolução
- ○ Apenas o valor
- ○ Numa tabela, ilustrando também os valores anteriores
- ○ Outra forma

*** O seu sistema permite prever o valor a pagar?**

Não consegue 0	1	Dificilmente 2	3	Facilmente 4
○	○	○	○	○

O seguinte grupo de perguntas refere-se à evolução do seu consumo:

*** É útil para si saber qual foi a evolução do consumo, face ao do mês anterior?**

Sem Interesse 0	1	Indiferente 2	3	Muito útil 4
○	○	○	○	○

*** Como gostaria de ver a evolução do seu consumo?**

- ○ Sobre a forma de gráfico, mostrando a evolução
- ○ Apenas o valor
- ○ Numa tabela, ilustrando também os valores anteriores
- ○ Outra forma

*** O seu sistema mostra a evolução do consumo em relação a períodos anteriores?**

Não consegue 0	1	Dificilmente 2	3	Facilmente 4
○	○	○	○	○

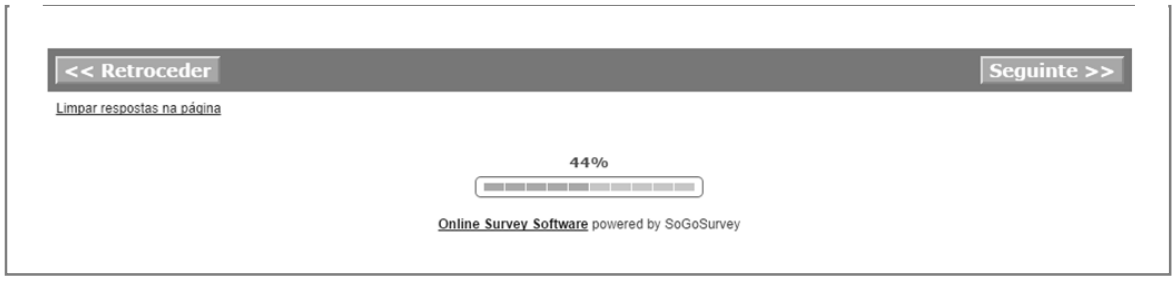
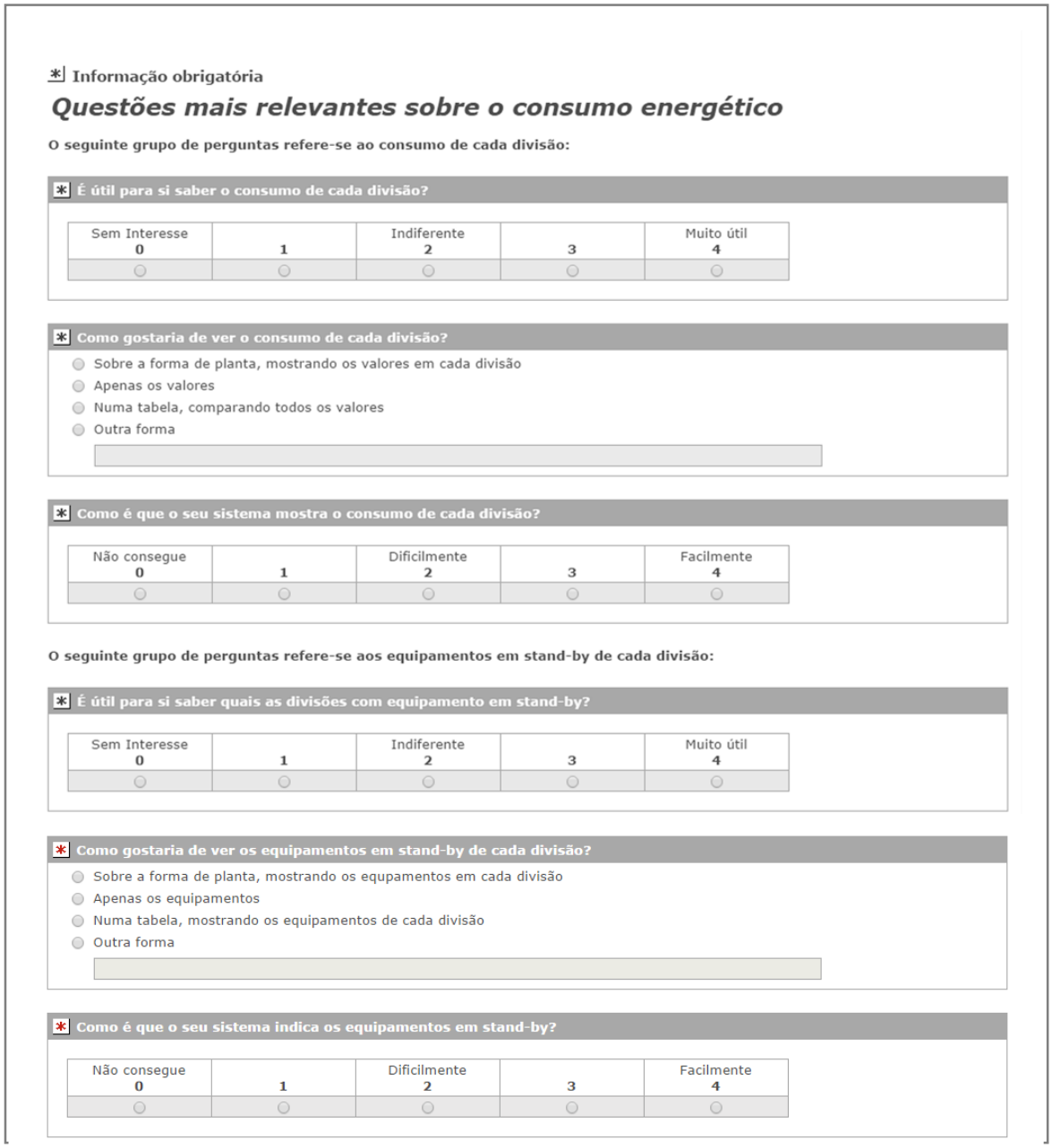
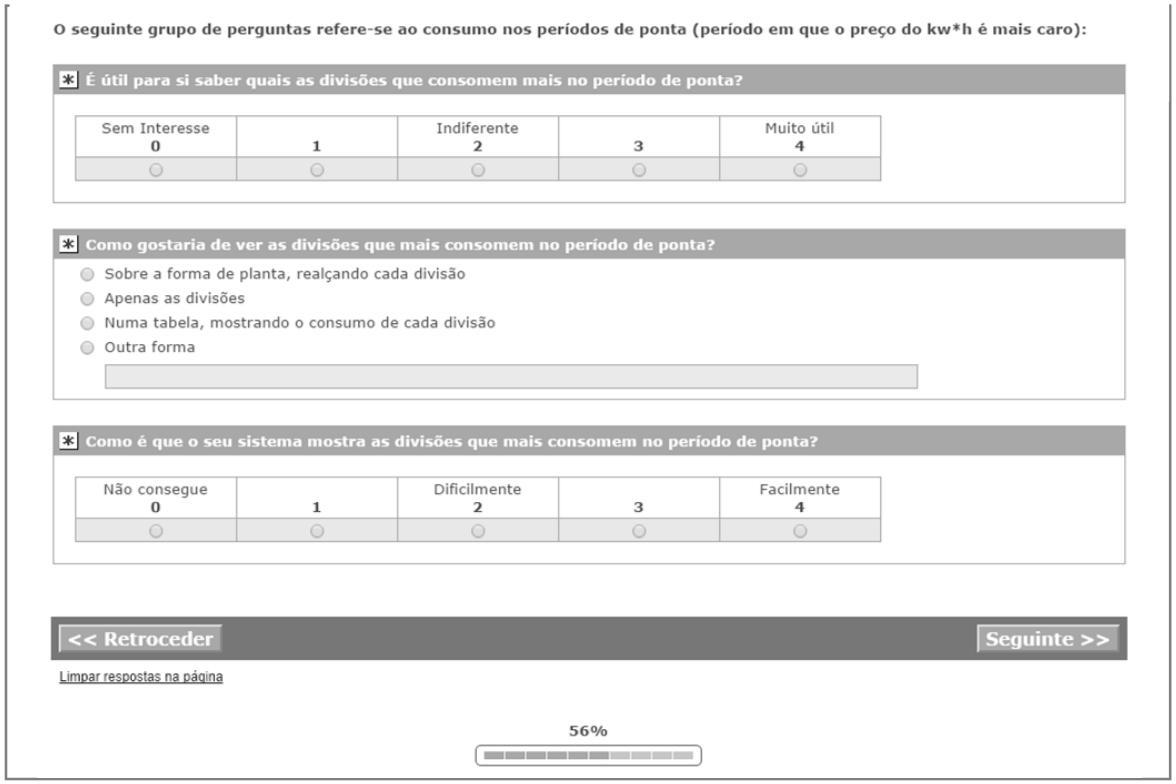


Figure 39 – Online Survey for gathering the concerns about the energy consumption (Page 4)




O seguinte grupo de perguntas refere-se ao consumo nos períodos de ponta (período em que o preço do kw*h é mais caro):

* É útil para si saber quais as divisões que consomem mais no período de ponta?

Sem Interesse 0	1	Indiferente 2	3	Muito útil 4
○	○	○	○	○

* Como gostaria de ver as divisões que mais consomem no período de ponta?

- Sobre a forma de planta, realçando cada divisão
- Apenas as divisões
- Numa tabela, mostrando o consumo de cada divisão
- Outra forma

* Como é que o seu sistema mostra as divisões que mais consomem no período de ponta?

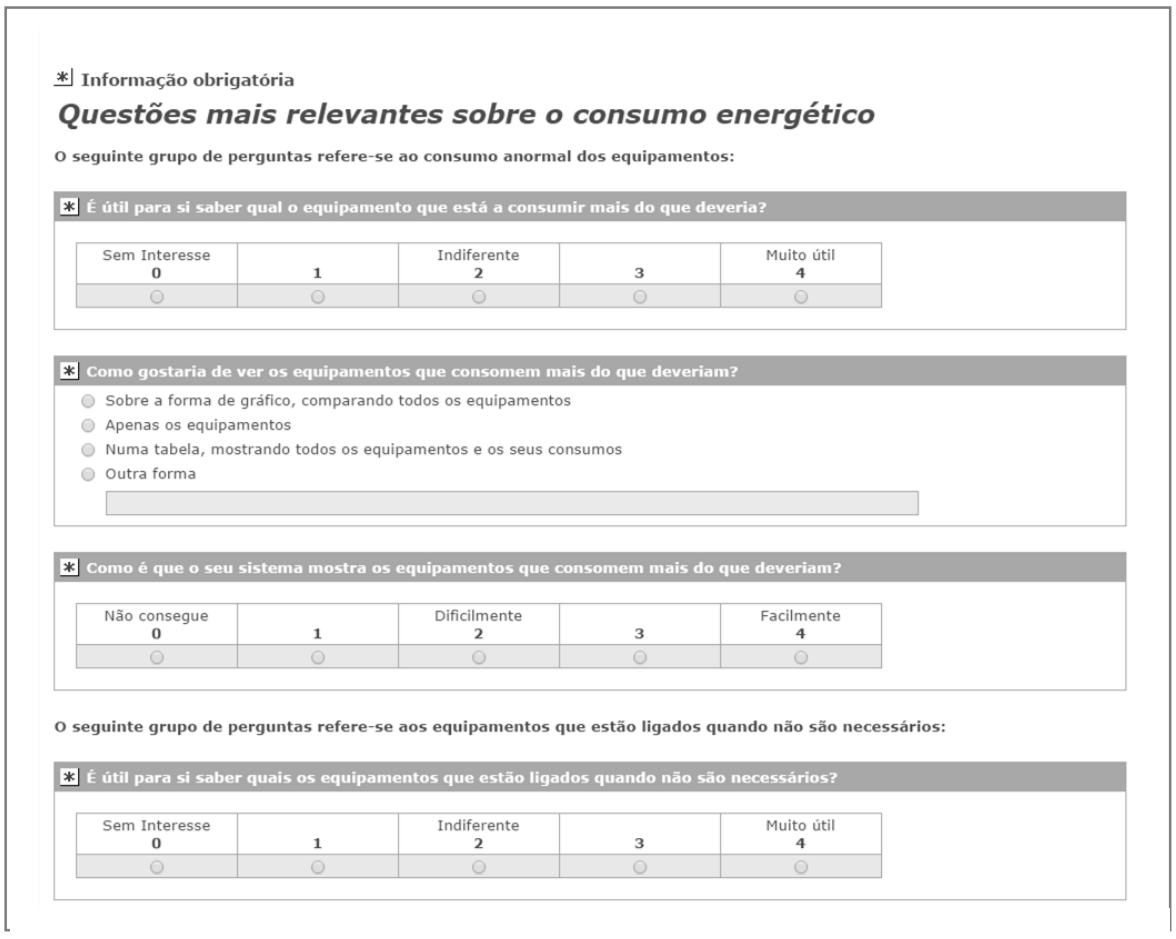
Não consegue 0	1	Dificilmente 2	3	Facilmente 4
○	○	○	○	○

<< Retroceder | Seguinte >>

Limpar respostas na página

56%

Figure 40 – Online Survey for gathering the concerns about the energy consumption (Page 5)


* Informação obrigatória

Questões mais relevantes sobre o consumo energético

O seguinte grupo de perguntas refere-se ao consumo anormal dos equipamentos:

* É útil para si saber qual o equipamento que está a consumir mais do que deveria?

Sem Interesse 0	1	Indiferente 2	3	Muito útil 4
○	○	○	○	○

* Como gostaria de ver os equipamentos que consomem mais do que deveriam?

- Sobre a forma de gráfico, comparando todos os equipamentos
- Apenas os equipamentos
- Numa tabela, mostrando todos os equipamentos e os seus consumos
- Outra forma

* Como é que o seu sistema mostra os equipamentos que consomem mais do que deveriam?

Não consegue 0	1	Dificilmente 2	3	Facilmente 4
○	○	○	○	○

O seguinte grupo de perguntas refere-se aos equipamentos que estão ligados quando não são necessários:

* É útil para si saber quais os equipamentos que estão ligados quando não são necessários?

Sem Interesse 0	1	Indiferente 2	3	Muito útil 4
○	○	○	○	○

*** Como gostaria de ver os equipamentos que estão ligados quando não são necessários?**

- ○ Sobre a forma de planta, indicando os equipamentos de cada divisão
- ○ Apenas os equipamentos
- ○ Numa tabela, mostrando todos os equipamentos
- ○ Outra forma

*** Como é que o seu sistema mostra os equipamentos que estão ligados quando não são necessários?**

Não consegue 0	1	Dificilmente 2	3	Facilmente 4
○	○	○	○	○

O seguinte grupo de perguntas refere-se aos equipamentos que têm programas com elevado consumo:

*** É útil para si saber quais os equipamentos que estão a utilizar programas com maior consumo?** [ex: programa da máquina de lavar louça]

Sem Interesse 0	1	Indiferente 2	3	Muito útil 4
○	○	○	○	○

*** Como gostaria de ver os equipamentos que estão a utilizar programas com maior consumo?**

- ○ Sobre a forma de planta, indicando os equipamentos de cada divisão
- ○ Apenas os equipamentos
- ○ Numa tabela, mostrando todos os equipamentos
- ○ Outra forma

*** Como é que o seu sistema mostra os equipamentos que estão a utilizar programas com maior consumo?**

Não consegue 0	1	Dificilmente 2	3	Facilmente 4
○	○	○	○	○

O seguinte grupo de perguntas refere-se aos equipamentos em fim de vida:

*** É útil para si saber quais os equipamentos em fim de vida?**

Sem Interesse 0	1	Indiferente 2	3	Muito útil 4
○	○	○	○	○

*** Como gostaria de ver os equipamentos em fim de vida?**

- ○ Sobre a forma de planta, indicando os equipamentos de cada divisão
- ○ Apenas os equipamentos
- ○ Numa tabela, mostrando todos os equipamentos
- ○ Outra forma

*** Como é que o seu sistema mostra os equipamentos em fim de vida?**

Não consegue 0	1	Dificilmente 2	3	Facilmente 4
○	○	○	○	○

O seguinte grupo de perguntas refere-se às categorias dos equipamentos (conforto, entertenimento, cozinha, ...):

*** É útil para si saber quais as categorias dos equipamentos que consomem mais?**

Sem Interesse 0	1	Indiferente 2	3	Muito útil 4
○	○	○	○	○

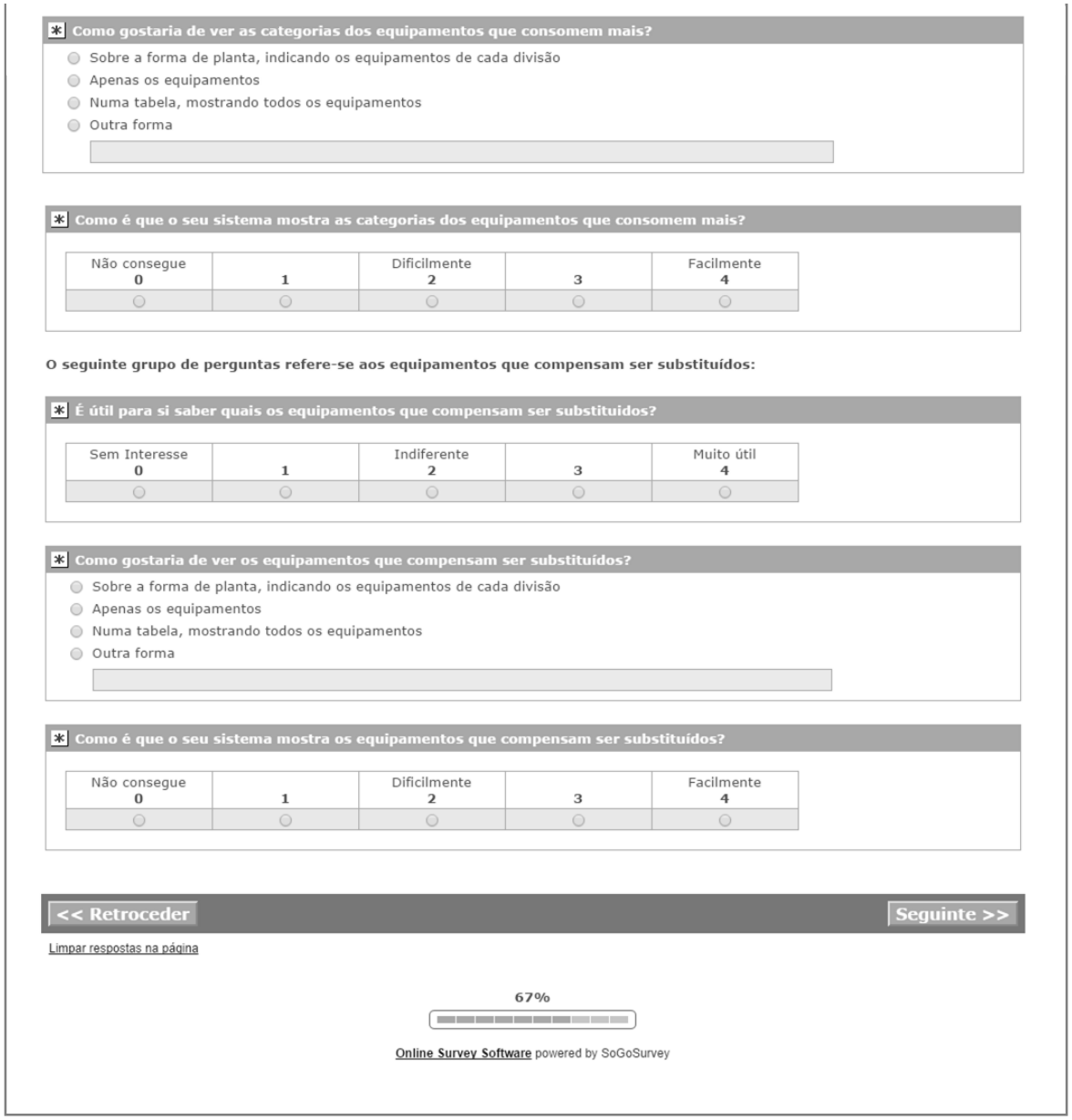
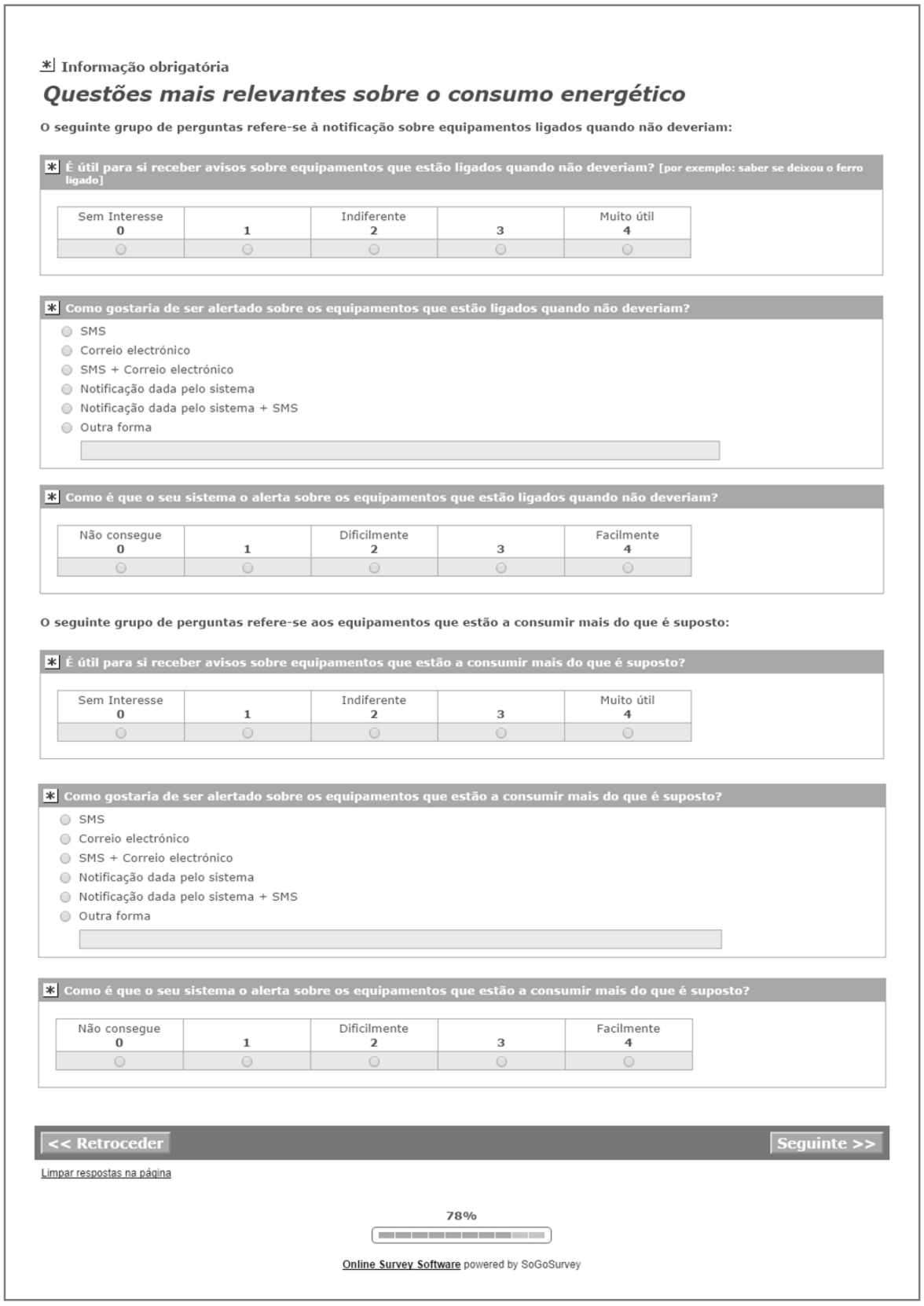


Figure 41 – Online Survey for gathering the concerns about the energy consumption (Page 6)



* Informação obrigatória

Questões mais relevantes sobre o consumo energético

O seguinte grupo de perguntas refere-se à notificação sobre equipamentos ligados quando não deveriam:

*** É útil para si receber avisos sobre equipamentos que estão ligados quando não deveriam? [por exemplo: saber se deixou o ferro ligado]**

Sem Interesse		Indiferente		Muito útil
0	1	2	3	4
○	○	○	○	○

*** Como gostaria de ser alertado sobre os equipamentos que estão ligados quando não deveriam?**

- ○ SMS
- ○ Correio electrónico
- ○ SMS + Correio electrónico
- ○ Notificação dada pelo sistema
- ○ Notificação dada pelo sistema + SMS
- ○ Outra forma

*** Como é que o seu sistema o alerta sobre os equipamentos que estão ligados quando não deveriam?**

Não consegue		Dificilmente		Facilmente
0	1	2	3	4
○	○	○	○	○

O seguinte grupo de perguntas refere-se aos equipamentos que estão a consumir mais do que é suposto:

*** É útil para si receber avisos sobre equipamentos que estão a consumir mais do que é suposto?**

Sem Interesse		Indiferente		Muito útil
0	1	2	3	4
○	○	○	○	○

*** Como gostaria de ser alertado sobre os equipamentos que estão a consumir mais do que é suposto?**

- ○ SMS
- ○ Correio electrónico
- ○ SMS + Correio electrónico
- ○ Notificação dada pelo sistema
- ○ Notificação dada pelo sistema + SMS
- ○ Outra forma

*** Como é que o seu sistema o alerta sobre os equipamentos que estão a consumir mais do que é suposto?**

Não consegue		Dificilmente		Facilmente
0	1	2	3	4
○	○	○	○	○

<< Retroceder | Seguinte >>

Limpar respostas na página

78%

Online Survey Software powered by SoGoSurvey

Figure 42 – Online Survey for gathering the concerns about the energy consumption (Page 7)

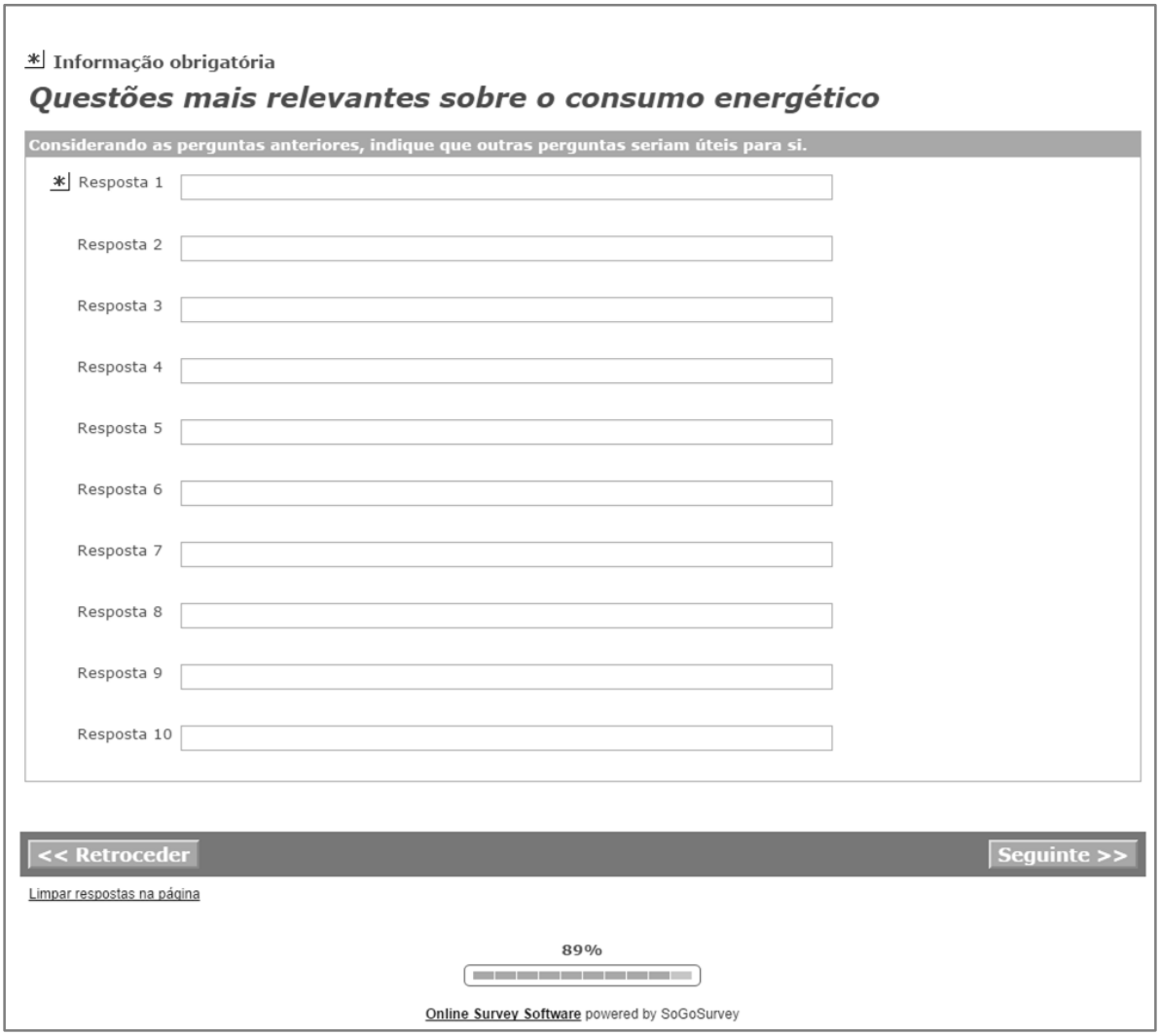


Figure 43 – Online Survey for gathering the concerns about the energy consumption (Page 8)

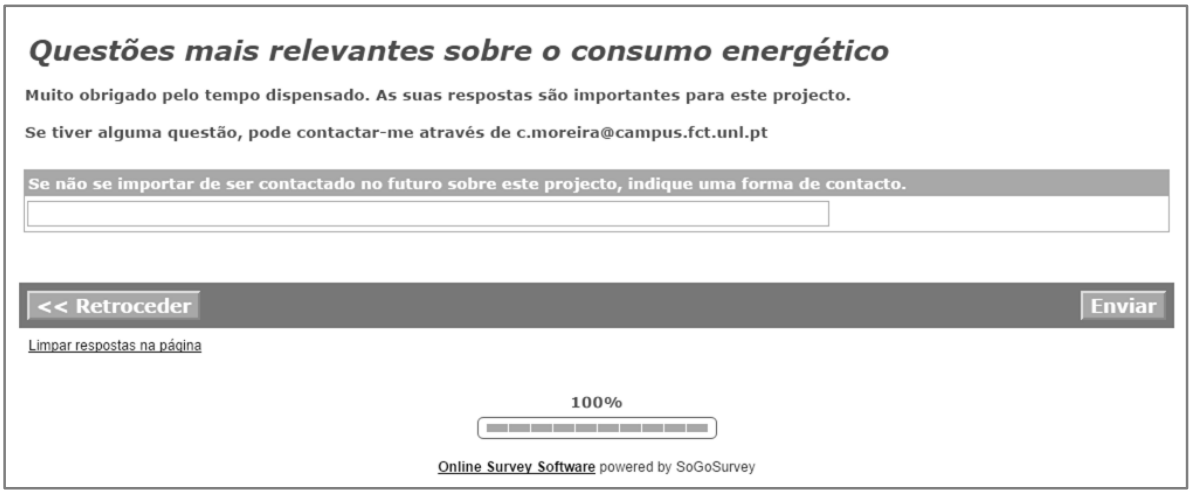


Figure 44 – Online Survey for gathering the concerns about the energy consumption (Page 9)

9.2 FEATURES MODEL

The Figure 45 represents the language variability.

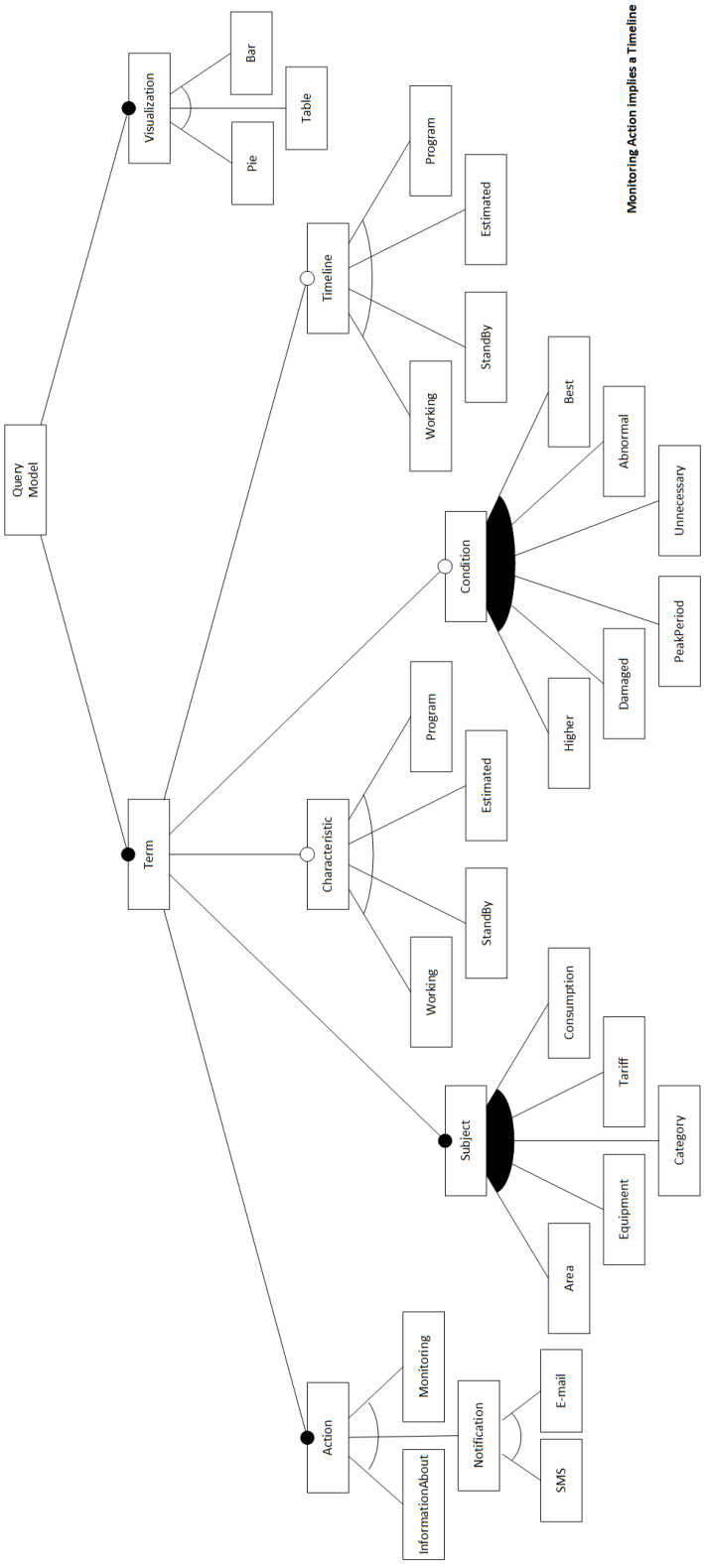


Figure 45 – Features Model

9.3 DOMAIN MODEL

This image represents the language elements and relationships.

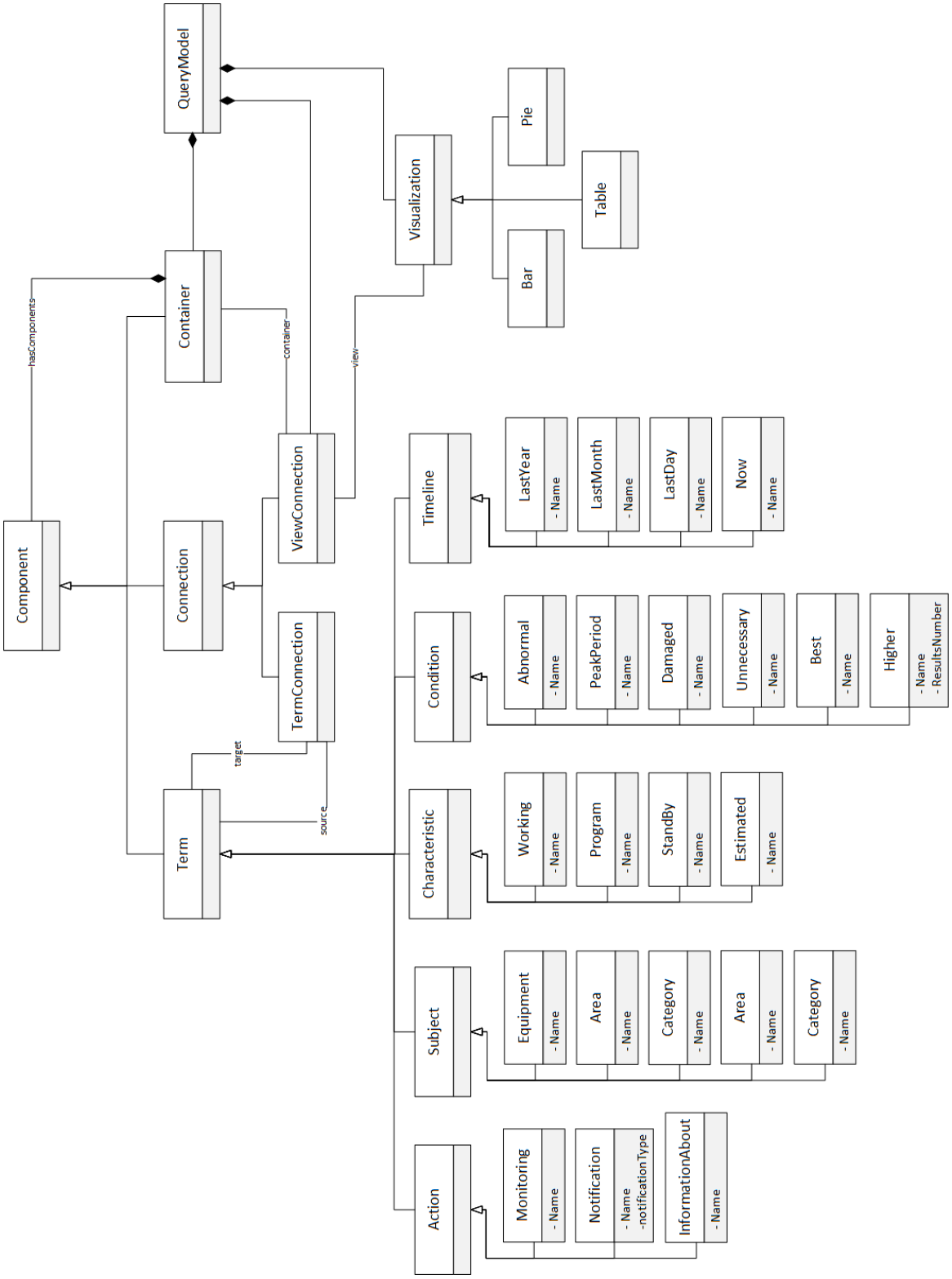


Figure 46 – Domain Model

9.4 LANGUAGE METAMODEL

The Figure 47 presents the language metamodel defined in the Ecore Model.

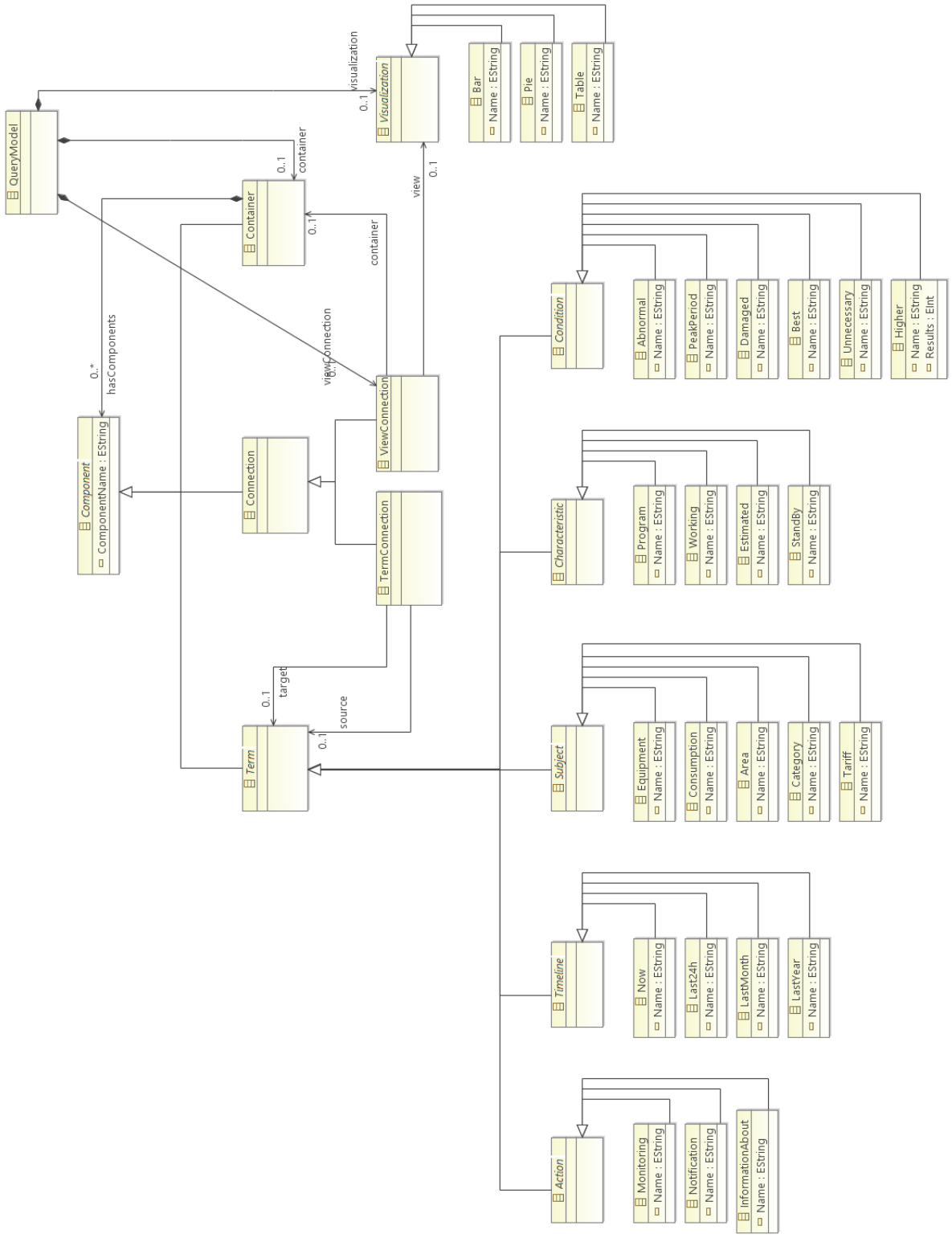


Figure 47 – Ecore Model

9.5 EMF

The EMF file objective is the description of the metamodel concepts, rules and properties in a textual form. The Listing 13 presents this description for the EQuery Language.

Listing 13 – Elements representation in the EMF file

```
@namespace(uri="queries_v3", prefix="queries_v3")
package queries_v3;

@gmf.diagram
class QueryModel {
  val Container container;
  val Visualization visualization;
  val ViewConnection viewConnection;
}

@gmf.node(label="ComponentName")
abstract class Component {
  attr String ComponentName;
}

@gmf.node(label="ComponentName", label.icon="true",
tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Container.png")
class Container extends Component {

  @gmf.compartment
  val Component[*] hasComponents;
}

abstract class Term extends Component {
}

@gmf.link(source="source", target="target", style="dot", width="2",
target.decoration="arrow", label.icon="true",
tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Link.png")
class TermConnection extends Connection {
  ref Term source;
  ref Term target;
}

@gmf.link(source="container", target="view", style="dash", width="2",
color="82,4,4", target.decoration="arrow", label.icon="true",
tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/LinkView.png")
class ViewConnection extends Connection {
  ref Container container;
  ref Visualization view;
}

abstract class Action extends Term {
}
```

```
abstract class Timeline extends Term {
}

abstract class Subject extends Term {
}

abstract class Characteristic extends Term {
}

abstract class Condition extends Term {
}

@gmf.node(label="Name", figure="figures.Now", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Now.png")
class Now extends Timeline {
  readonly attr String Name = "Now";
}

@gmf.node(label="Name", figure="figures.Last24h", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Last24h.png")
class Last24h extends Timeline {
  readonly attr String Name = "Last 24h";
}

@gmf.node(label="Name", figure="figures.LastMonth", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/LastMonth.png")
class LastMonth extends Timeline {
  readonly attr String Name = "Last Month";
}

@gmf.node(label="Name", figure="figures.LastYear", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/LastYear.png")
class LastYear extends Timeline {
  readonly attr String Name = "Last Year";
}

@gmf.node(label="Name", figure="figures.Monitoring", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Monitoring.png")
class Monitoring extends Action {
  readonly attr String Name = "Monitoring";
}

@gmf.node(label="Name", figure="figures.Notification", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Notification.png")
class Notification extends Action {
  readonly attr String Name = "Notification";
}
```

```
@gmf.node(label="Name", figure="figures.InformationAbout", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/InformationAbout.png")
class InformationAbout extends Action {
  readonly attr String Name = "Information About";
}

@gmf.node(label="Name", figure="figures.Equipment", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Equipment.png")
class Equipment extends Subject {
  readonly attr String Name = "Equipment";
}

@gmf.node(label="Name", figure="figures.Consumption", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Consumption.png")
class Consumption extends Subject {
  readonly attr String Name = "Consumption";
}

@gmf.node(label="Name", figure="figures.Area", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Area.png")
class Area extends Subject {
  readonly attr String Name = "Area";
}

@gmf.node(label="Name", figure="figures.Category", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Category.png")
class Category extends Subject {
  readonly attr String Name = "Category";
}

@gmf.node(label="Name", figure="figures.Tariff", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Tariff.png")
class Tariff extends Subject {
  readonly attr String Name = "Tariff";
}

@gmf.node(label="Name", figure="figures.Program", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Program.png")
class Program extends Characteristic {
  readonly attr String Name = "Program";
}

@gmf.node(label="Name", figure="figures.Working", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Working.png")
class Working extends Characteristic {
  readonly attr String Name = "Working";
}

@gmf.node(label="Name", figure="figures.StandBy", label.icon="false",
```

```
@gmf.node(label="Name", figure="figures.StandBy", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/StandBy.png")
class StandBy extends Characteristic {
  readonly attr String Name = "Stand-by";
}

@gmf.node(label="Name", figure="figures.Estimated", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Estimated.png")
class Estimated extends Characteristic {
  readonly attr String Name = "Estimated";
}

@gmf.node(label="Name", figure="figures.Higher", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
      tool.small.path="/icons/full/obj16/Higher.png")
class Higher extends Condition {
  readonly attr String Name = "Higher";
  attr int Results = 10;
}

@gmf.node(label="Name", figure="figures.Abnormal", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Abnormal.png")
class Abnormal extends Condition {
  readonly attr String Name = "Abnormal";
}

@gmf.node(label="Name", figure="figures.PeakPeriod", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/PeakPeriod.png")
class PeakPeriod extends Condition {
  readonly attr String Name = "Peak Period";
}

@gmf.node(label="Name", figure="figures.Damaged", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Damaged.png")
class Damaged extends Condition {
  readonly attr String Name = "Damaged";
}

@gmf.node(label="Name", figure="figures.Best", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Best.png")
class Best extends Condition {
  readonly attr String Name = "Best";
}
```

```
@gmf.node(label="Name", figure="figures.Unnecessary", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Unnecessary.png")
class Unnecessary extends Condition {
  readonly attr String Name = "Unnecessary";
}

abstract class Visualization {
}

@gmf.node(label="Name", figure="figures.Pie", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Pie.png")
class Pie extends Visualization {
  readonly attr String Name = "Pie Chart";
}

@gmf.node(label="Name", figure="figures.Bar", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Bar.png")
class Bar extends Visualization {
  readonly attr String Name = "Bar Chart";
}

@gmf.node(label="Name", figure="figures.Table", label.icon="false",
label.placement="external", tool.small.bundle="Thesis_DSLv3.edit",
tool.small.path="/icons/full/obj16/Table.png")
class Table extends Visualization {
  readonly attr String Name = "Table";
}

class Connection extends Component {
}
```

9.6 EPSILON VALIDATION LANGUAGE

The Listing 14 presents the validation rules defined in the EVL file. The description of these rules can be found in the Section 5.2.3.

Listing 14 – Definition of the validation rules in the EVL file.

```
context Container  {

constraint checkContainer {
	check : not self.hasComponents.exists(t|t.isKindOf(Container))
	message : 'A Container can not have a container'
	fix {
		title : 'Removing Container'
		do {
			for(p in
self.hasComponents.select(t|t.isKindOf(Container))){
				delete p;
			}
		}
	}
}

constraint hasToHaveAName {
	check : self.ComponentName.isDefined()
	message : 'A Container has to have a name'
}

constraint onlyOneAction {
	check : not (self.hasComponents.select(t|t.isKindOf(Action)).size() > 1)
	message : 'A Container can only have one Action'
}

constraint hasToHaveAnAction {
	check : not (self.hasComponents.select(t|t.isKindOf(Action)).size() < 1)
	message : 'A Container has to have an Action'
}

constraint onlyOneTimeline {
	check : not (self.hasComponents.select(t|t.isKindOf(Timeline)).size() >
1)
	message : 'A Container can only have one Timeline'
}

constraint atLeastOneSubject {
	check : (self.hasComponents.select(t|t.isKindOf(Subject)).size() >= 1)
	message : 'A Container has to have at least one Subject'
}

constraint visualizationType {
	check : (Visualization.allInstances.size() = 1) and
(ViewConnection.allInstances.exists(i|i.container = self))
	message : 'A Container has to have one visualization type'
}
}
```

```
context Term {

constraint noIsolations {
	check : TermConnection.allInstances.exists(i|i.Source = self) or
TermConnection.allInstances.exists(i|i.Target = self)
	message : 'All the elements must be connected'
}
}

context Monitoring {

constraint monitoringActionHasToHaveATimeline {
	check : Term.allInstances.exists(t|t.isKindOf(Monitoring)) and
Term.allInstances.exists(t|t.isKindOf(Timeline))
	message : 'The Monitoring Action has to have a Timeline'
}
}

context Notification{

constraint doesNotNeedTimeline {
	check : not Term.allInstances.exists(t|t.isKindOf(Timeline))
	message : 'The action Notification does not need a Timeline'
}
}

context InformationAbout {

constraint justForTariffs {
	check : (Subject.allInstances.size == 1 and
			Subject.allInstances.exists(t|t.isKindOf(Tariff)))
	message : 'The Action Information About can only be used with the Tariff
Subject'
}

constraint doesNotNeedTimelime {
	check : not Term.allInstances.exists(t|t.isKindOf(Timeline))
	message : 'The action Information About does not need a Timeline'
}
}

context StandBy {

constraint justForEquipment {
	check : Subject.allInstances.exists(t|t.isKindOf(Equipment))
	message : 'The Caracteristic Stand-by is only used with the Equipment
Subject'
}
}

context Estimated {

constraint justForTariffs {
	check : Subject.allInstances.exists(t|t.isKindOf(Tariff))
	message : 'The Caracteristic Estimated is only used with the Tariff
Subject'
}
```

```
context Working {

constraint justForEquipment{
	check : Subject.allInstances.exists(t|t.isKindOf(Equipment))
	message : 'The Caracteristic Working is only used with the Equipment
Subject'
}
}


context Program {

constraint justForEquipment{
	check : Subject.allInstances.exists(t|t.isKindOf(Equipment))
	message : 'The Caracteristic Program is only used with the Equipment
Subject'
}
}

context Higher {

constraint justForConsumption{
	check : Subject.allInstances.exists(t|t.isKindOf(Consumption))
	message : 'The Condition Higher is only used with the Consumption
Subject'
}
}

context Best {

constraint justForTariffs {
	check : Subject.allInstances.exists(t|t.isKindOf(Tariff))
	message : 'The Condition Best is only used with the Tariff Subject'
}
}

context Damaged {

constraint justForEquipments{
	check : Subject.allInstances.exists(t|t.isKindOf(Equipment))
	message : 'The Condition Damaged is only used with the Equipment
Subject'
}
}

context Abnormal {

constraint justForEquipmentConsumption{
	check : Subject.allInstances.exists(t|t.isKindOf(Equipment))
		and Subject.allInstances.exists(t|t.isKindOf(Consumption))
	message : 'The Condition Abnormal is only usable with the Equipment and
Consumption Subject'
}
}
```

```
context Bar{

constraint justForGroupAndEquipment{
	check : (Subject.allInstances.exists(t|t.isKindOf(Area)) or
Subject.allInstances.exists(t|t.isKindOf(Category)) or
		(Subject.allInstances.exists(t|t.isKindOf(Equipment)) and
		(Condition.allInstances.select(t|t.isKindOf(Abnormal)).size == 0
and
		Condition.allInstances.select(t|t.isKindOf(Damaged)).size == 0)))
	message : 'The Bar Chart is only possible for query envolving the
consumption of Areas, Categories, Equipments and Equipments in Stand-by'
}
}

context Pie{

constraint justForGroupAndEquipment{
	check : (Subject.allInstances.exists(t|t.isKindOf(Area)) or
Subject.allInstances.exists(t|t.isKindOf(Category)) or
		(Subject.allInstances.exists(t|t.isKindOf(Equipment)) and
		(Condition.allInstances.select(t|t.isKindOf(Abnormal)).size == 0
and
		Condition.allInstances.select(t|t.isKindOf(Damaged)).size == 0)))
	message : 'The Pie Chart is only possible for query envolving the
consumption of Areas, Categories, Equipments and Equipments in Stand-by'
}
}
```

9.7 EPSILON GENERATION LANGUAGE

In order to convert the questions made in the concrete syntax it was necessary to develop two EGL files (Listing 15 and Listing 16). The first file, defines the SQL semantics and uses the functions implemented in the second one.

Listing 15 – The EGL file with the SQL semantics

```
[%
import "query_util.egl";
%]

[%var container = Container.allInstances().first();%]

--
[[%=getQueryType(container)%]]<[%=getVisualization(container)%],[%=getObjectTyp
e(container)%]>[%=container.ComponentName%]

SELECT [%=getSelectBlock(container)%]
FROM [%=getFromBlock(container)%]
WHERE [%=getWhereBlock(container)%]
[%=getGroupByBlock(container)%]
[%=getOrderByBlock(container)%]
[%=getLimit(container)%];
```

Listing 16 – The EGL file that associates each element to the necessary SQL code

```
[% operation getSelectBlock(container : Container) : String {
var select : String = "";
var time : String = getTimeline(container);
for (t in container.hasComponents.select(c|c.isKindOf(Term))) {
      if(t.isKindOf(Area)) {
            return "myhouse.area.name as Area, getAreaKWh(area.ID, "+ time +")
as KWh, getAreaKWhPrice(area.ID, "+ time +") as 'Cost (€)'";
      }
      if(t.isKindOf(Equipment) and container.hasComponents.select(t |
t.isKindOf(Abnormal)).size() == 0 and container.hasComponents.select(t |
t.isKindOf(Damaged)).size() == 0) {
            select = "myhouse.equipment.name as Equipment, myhouse.area.name
as Area";
            if(container.hasComponents.select(t | t.isKindOf(StandBy)).size()
== 1){
                  return select + ", getEquipmentKWh(AVG(consumption),
equipment.ID, "+ time +", 0) as KWh, getEquipmentKWhPrice(equipment.ID, "+ time
+", 0) as 'Cost (€)'";
            }
            else{
                  return select + ", getEquipmentKWh(AVG(consumption),
equipment.ID, "+ time +", 1) as KWh, getEquipmentKWhPrice(equipment.ID, "+ time
+", 1) as 'Cost (€)'";
            }
      }
```

```
	if(t.isKindOf(Consumption) and container.hasComponents.select(t |
t.isKindOf(Subject)).size() == 1) {
		return "myhouse.`historic.consumption`.consumption as
Consumption";
	}
	if(t.isKindOf(Category)){
		return "myhouse.category.name as Category,
getCategoryKWh(category.ID, "+ time +") as KWh,
getCategoryKWhPrice(category.ID, "+ time +") as 'Cost (€)'";
	}
	if(t.isKindOf(Damaged) or t.isKindOf(Abnormal)){
		return  "myhouse.equipment.name as Equipment, myhouse.area.name as
Area, expectedConsumption as 'Expected Consumption', standbyConsumption as
'Expected Stand-by Consumption', consumptionAvg as 'Average Consumption',
consumptionPeak as 'Consumption Peak',
myhouse.`historic.equipment_consumption`.date as Date";
	}
	if(t.isKindOf(Unnecessary)){
		return  "equipment.name as Equipment, area.name as Area,
getStartTime(myhouse.`historic.activity`.start,
myhouse.`sensor.presence`.start) as start,
getEndTime(myhouse.`historic.activity`.end, myhouse.`sensor.presence`.end) as
end, getUnusedEquipmentKwh(equipment.ID,
getStartTime(myhouse.`historic.activity`.start,
myhouse.`sensor.presence`.start), getEndTime(myhouse.`historic.activity`.end,
myhouse.`sensor.presence`.end)) as KWh,
getUnusedEquipmentKwhPrice(equipment.ID,
getStartTime(myhouse.`historic.activity`.start,
myhouse.`sensor.presence`.start), getEndTime(myhouse.`historic.activity`.end,
myhouse.`sensor.presence`.end)) as 'Cost (€)'";
	}
	if(t.isKindOf(PeakPeriod)){
		return  "equipment.name as Equipment, area.name as Area,
myhouse.`historic.activity`.start, myhouse.`historic.activity`.end,
getPeakKWh(equipment.ID, "+ time +", 1) as KWh,
getPeakKWhPrice(getPeakKWh(equipment.ID, "+ time +") as 'Cost (€)'";
	}
	if(t.isKindOf(Best)){
		return  "company as Company, type as Type, averagedCost(ID, 3) as
'Cost per Month (€)', selected as 'Select Tariff'";
	}
	if(t.isKindOf(Estimated)){
		return  "getEstimatedConsumption() as 'Value to Pay (€)'";
	}
}
return select;
} %]

[% operation getFromBlock(container : Container) : String {
var from : String = "";
for (t in container.hasComponents.select(c|c.isKindOf(Term))){
	if(t.isKindOf(Area)) {
		return  "myhouse.equipment INNER JOIN myhouse.area ON (area =
myhouse.area.ID)" +
		" INNER JOIN myhouse.`historic.activity` ON (myhouse.equipment.ID
= equip)";
	}
```

```
	if(t.isKindOf(Equipment) and container.hasComponents.select(t |
t.isKindOf(StandBy)).size() == 0 and container.hasComponents.select(t |
t.isKindOf(Abnormal)).size() == 0 and container.hasComponents.select(t |
t.isKindOf(Damaged)).size() == 0) {
		if(from.length() > 0){
			return "myhouse.equipment INNER JOIN myhouse.area ON (area
= myhouse.area.ID)"
			+ " INNER JOIN myhouse.`historic.activity` ON
(myhouse.equipment.ID = equip) " + from;
		}
		else{
			from = "myhouse.equipment INNER JOIN myhouse.area ON (area
= myhouse.area.ID)"
			+ " INNER JOIN myhouse.`historic.activity` ON
(myhouse.equipment.ID = equip)";
		}
	}
	if(t.isKindOf(Consumption) and container.hasComponents.select(t |
t.isKindOf(Subject)).size() == 1) {
		return  "myhouse.`historic.consumption`";
	}
	if(t.isKindOf(Category)) {
		return  "myhouse.equipment INNER JOIN myhouse.equipment_category
ON (myhouse.equipment.ID = equipID)"
		+ " INNER JOIN myhouse.category ON (myhouse.category.ID = catID)"
		+ " INNER JOIN myhouse.`historic.activity` ON
(myhouse.equipment.ID = equip)";
	}
	if(t.isKindOf(Unnecessary)) {
		if(from.length() > 0){
			return from + " INNER JOIN myhouse.`sensor.presence` ON
(roomID = area)";
		}
		else{
			from = "INNER JOIN myhouse.`sensor.presence` ON (roomID =
area)";
		}
	}
	if(t.isKindOf(StandBy)) {
		return "myhouse.equipment INNER JOIN myhouse.area ON (area =
myhouse.area.ID)"
		+ " INNER JOIN myhouse.`historic.unused_equipment` ON
(myhouse.equipment.ID = equip)";
	}
	if(t.isKindOf(PeakPeriod)) {
		if(from.length() > 0){
			return from + " INNER JOIN tariff_schedule ON
(tariff_period = tariff_schedule.ID)";
		}
		else{
			from = "INNER JOIN tariff_schedule ON (tariff_period =
tariff_schedule.ID)";
		}
	}
```

```
	if(t.isKindOf(Damaged) or t.isKindOf(Abnormal)){
		return  "myhouse.equipment INNER JOIN myhouse.area ON (area =
myhouse.area.ID)"
		+ " INNER JOIN myhouse.`historic.equipment_consumption` ON
(myhouse.equipment.ID = equip)";
	}
	if(t.isKindOf(Best)){
		return "Tariff";
	}
}
return from;
} %]


[% operation getWhereBlock(container : Container) : String {
var where : String = "";
var start : String = "date";

if((container.hasComponents.select(t | t.isKindOf(Abnormal)).size() == 0)){
	start = "myhouse.`historic.activity`.start";
}
if((container.hasComponents.select(t | t.isKindOf(StandBy)).size() == 1)){
	start = "myhouse.`historic.unused_equipment`.start";
}
for (t in container.hasComponents.select(c|c.isKindOf(Term))){
	if(t.isKindOf(Last24h)) {
		if(where.length() > 0){
			where = where + " AND (" + start + " >= (now() - INTERVAL 1
DAY))";
		}
		else{
			where = start + " >= (now() - INTERVAL 1 DAY)";
		}
	}
	if(t.isKindOf(LastMonth)) {
		if(where.length() > 0){
			where = where + " AND (" + start + " >= (now() - INTERVAL 1
MONTH))";
		}
		else{
			where = start + " >= (now() - INTERVAL 1 MONTH)";
		}
	}
	if(t.isKindOf(LastYear)) {
		if(where.length() > 0){
			where = where + " AND (" + start + " >= (now() - INTERVAL 1
YEAR))";
		}
		else{
			where = start + " >= (now() - INTERVAL 1 YEAR)";
		}
	}
```

```
	if(t.isKindOf(Abnormal) or t.isKindOf(Damaged)) {
		if(where.length() > 0){
			where = where + " AND expectedConsumption IS NOT NULL AND
("
			+ " ((consumptionAvg > (expectedConsumption +
(expectedConsumption/4))) OR (consumptionAvg < (expectedConsumption -
(expectedConsumption/4))))"
			+ " OR (standbyConsumption IS NOT NULL AND"
			+ " ((consumptionAvg > (standbyConsumption +
(standbyConsumption/4))) OR (consumptionAvg < (standbyConsumption -
(standbyConsumption/4))))))";
		}
		else{
			where = "expectedConsumption IS NOT NULL AND
(((consumptionAvg > (expectedConsumption + (expectedConsumption/4))) OR
(consumptionAvg < (expectedConsumption - (expectedConsumption/4))))"
			+ " OR (standbyConsumption IS NOT NULL AND"
			+ " ((consumptionAvg > (standbyConsumption +
(standbyConsumption/4))) OR (consumptionAvg < (standbyConsumption -
(standbyConsumption/4))))))";
		}
	}

	if(t.isKindOf(Unnecessary)) {
		if(where.length() > 0){
			where = where + " AND (occupation = 0 AND
((myhouse.`historic.activity`.start between myhouse.`sensor.presence`.start and
myhouse.`sensor.presence`.end) OR (myhouse.`historic.activity`.end between
myhouse.`sensor.presence`.start and myhouse.`sensor.presence`.end)))";
		}
		else{
			where = "(occupation = 0 AND
((myhouse.`historic.activity`.start between myhouse.`sensor.presence`.start and
myhouse.`sensor.presence`.end) OR (myhouse.`historic.activity`.end between
myhouse.`sensor.presence`.start and myhouse.`sensor.presence`.end)))";
		}
	}

	if(t.isKindOf(PeakPeriod)) {
		if(where.length() > 0){
			where = where + "AND typeOfPeriod = 'P'";
		}
		else{
			where = "typeOfPeriod = 'P'";
		}
	}
	if(t.isKindOf(Best)){
		return "powerKVA = (SELECT powerKVA FROM Tariff WHERE selected =
1)";
	}
}
return where;
} %]
```

```
[% operation getGroupByBlock(container : Container) : String {
var groupBy : String = "";
for (t in container.hasComponents.select(c|c.isKindOf(Term))){
            if(t.isKindOf(Equipment)) {
                  groupBy = "GROUP BY equipment.ID";
            }
            if(t.isKindOf(Area)) {
                  groupBy = "GROUP BY area.ID";
            }
            if(t.isKindOf(Category)) {
                  groupBy = "GROUP BY category.ID";
            }
}
return groupBy;
} %]

[% operation getHavingBlock(container : Container) : String {
var having : String = "";

return having;
} %]

[% operation getOrderByBlock(container : Container) : String {
var orderBy : String = "";
for (t in container.hasComponents.select(c|c.isKindOf(Term))){
      if(t.isKindOf(Higher)) {
            orderBy = "ORDER BY Cost DESC";
      }
      if(t.isKindOf(Abnormal) or t.isKindOf(Damaged)) {
            orderBy = "ORDER BY consumptionAvg DESC";
      }
      if(t.isKindOf(Best)){
            orderBy = "ORDER by averagedCost(ID, 3)";
      }
}
return orderBy;
} %]

[% operation getLimit(container : Container) : String {
if(container.hasComponents.exists(c|c.isKindOf(Higher))){
      var h = container.hasComponents.select(c|c.isKindOf(Higher)).first();
      return "Limit " + h.Results;
}

return "";
} %]

[% operation getTimeline(container : Container) : String {
var time : String = "";
for (t in container.hasComponents.select(c|c.isKindOf(Term))){
      if(t.isKindOf(Last24h)) {
            time = "(now() - INTERVAL 1 DAY)";
      }
      if(t.isKindOf(LastMonth)) {
            time = "(now() - INTERVAL 1 MONTH)";
      }
```

```
	if(t.isKindOf(LastYear)) {
		time = "(now() - INTERVAL 1 YEAR)";
	}
}
return time;
} %]

[% operation getQueryType(container : Container) : String {
var t = container.hasComponents.select(c|c.isKindOf(Action)).first();
if(t.isKindOf(Monitoring)) {
	return "M";
}
if(t.isKindOf(Notification)) {
	return "N";
}
if(t.isKindOf(InformationAbout)) {
	return "I";
}
return "Weird";
} %]

[% operation getVisualization(container : Container) : String {
var view = Visualization.allInstances().first();

if(view.isKindOf(Bar)){
	return "Bar";
}
if(view.isKindOf(Pie)){
	return "Pie";
}
return "Table";
} %]

[% operation getObjectType(container : Container) : String {
if(container.hasComponents.exists(c|c.isKindOf(Damaged)) or
container.hasComponents.exists(c|c.isKindOf(Abnormal))){
	return "Damaged";
}
if(container.hasComponents.exists(c|c.isKindOf(PeakPeriod)) or
container.hasComponents.exists(c|c.isKindOf(Unnecessary))){
	return "Wasted";
}
if(container.hasComponents.exists(c|c.isKindOf(Area)) or
container.hasComponents.exists(c|c.isKindOf(Category))){
	return "Group";
}
if(container.hasComponents.exists(c|c.isKindOf(Best))){
	return "Tariff";
}
if(container.hasComponents.exists(c|c.isKindOf(Equipment))){
	return "Equipment";
}

return "Value";
} %]
```

9.8 ENTITY RELATIONSHIP MODEL

The necessity of structuring the MySQL database occured in the parsing from the concrete syntax to SQL semantics. Therefore, the database scheme, shown in the Figure 51, was developed, where the entities are divided in three groups: Structure, Historic and Support.

In Figure 48 presentes the Structure group, which is responsible for defining the house arrangement. In this group there are included the Area, Category, Equipment and Program with the following attributes:

- Area (ID, Name, Description, m^2)
- Category (ID, Name, Description)
- Equipment (ID, Name, Description, Permanent, ExpectedConsumption, StandyConsumption, Area [FK])
- Program (Equipment, Program, Description, Consumption)

The entity Equipment and Program also have the information about the normal consumption of an equipment to be able to detect anomalies.

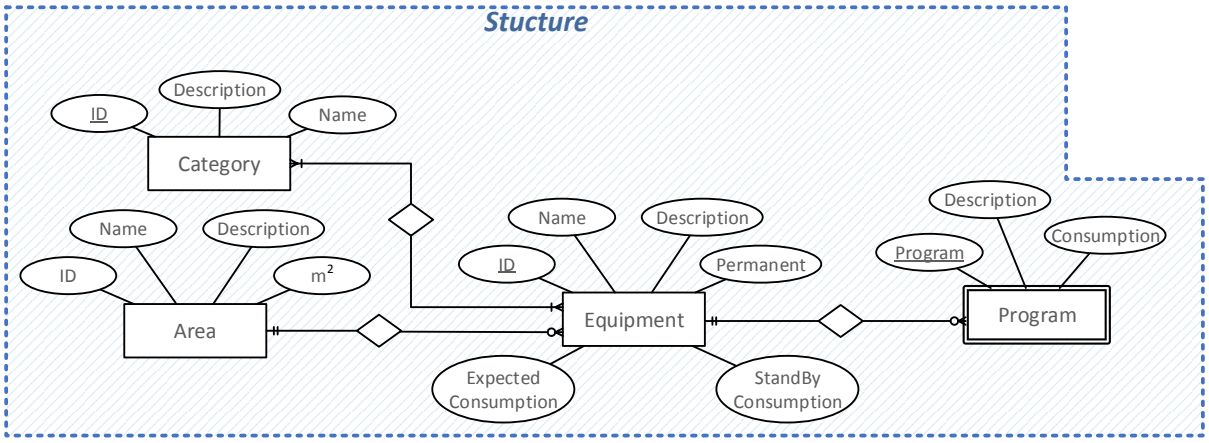


Figure 48 – Structure Group entities in the MySQL database

In the Support group are the entities that assist in the consumption calculations and estimations. Figure 49 represents the entities belonging to this group with the following attributes:

- TariffSchedule (ID, Start, End, Season, DayOfTheWeek, TypeOfPeriod)
- Tariff (ID, Company, PowerKvA, PriceDay, PriceMonth, Type, Selected)
- TariffSimple (Tariff, PriceKwh)
- TariffBi (Tariff, PriceKwhEmptyPeriod, PriceKwhNotEmptyPeriod)
- TariffTri (Tariff, PriceKwhEmptyPeriod, PriceKwhFullPeriod, PriceKwhPeakPeriod)

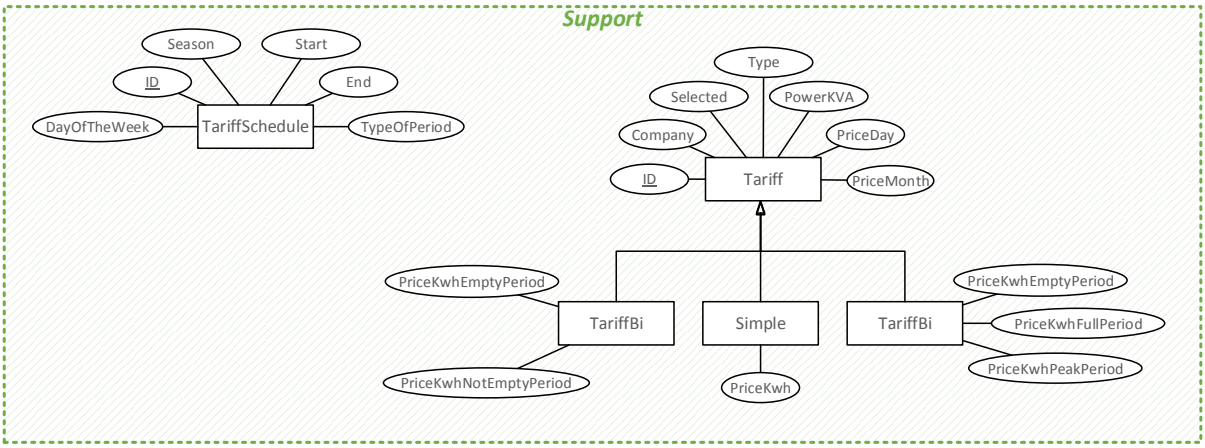


Figure 49 – Support Group entities in the MySQL database

The last group is the one that stores the historic information, which is provided by the Esper that receives it from the sensors. The Figure 50 presents the entities included in this group, which have the following characteristics:

- DaylightSensor (Area, Start, End, LightIntensity)
- TemperatureSensor (Area, Start, End, Temperature)
- PresenceSensor (Area, Start, End, Occupation)
- Activity (Equipment, Start, End, Consumption, Mode)
- EquipmentConsumption (Equipment, Start, End, ConsumptionPeak, ConsumptionAvg, Mode)
- UnusedEquipment (Equipment, Start, End, Consumption)
- Consumption (Date, Consumption)

Some of the mentioned entities requires the structure entity, since the information is connected to it, for instance, the Area is needed in the PresenceSensor to indicate the occupation in a specific room.

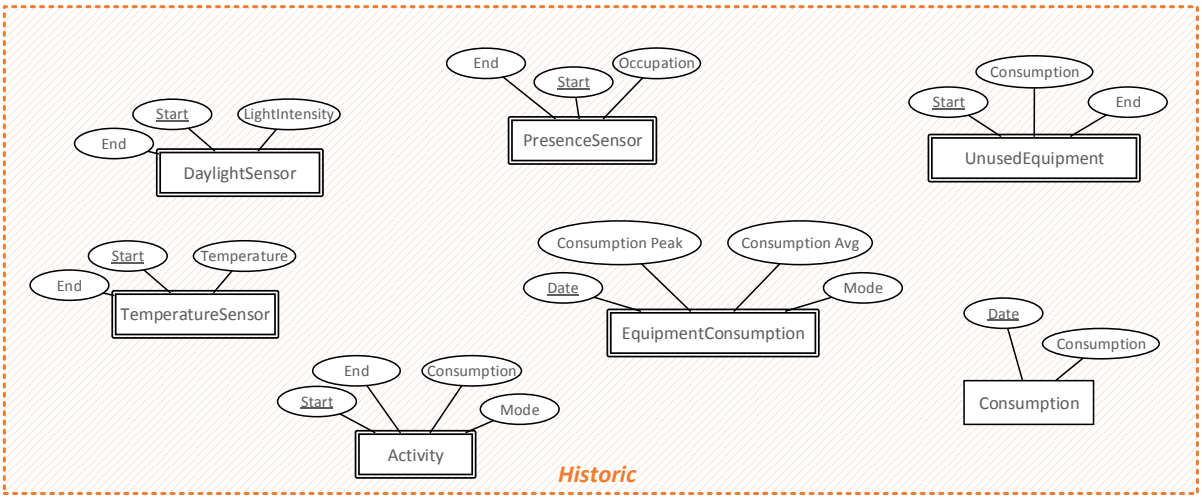


Figure 50 – Historic Group entities in the MySQL database

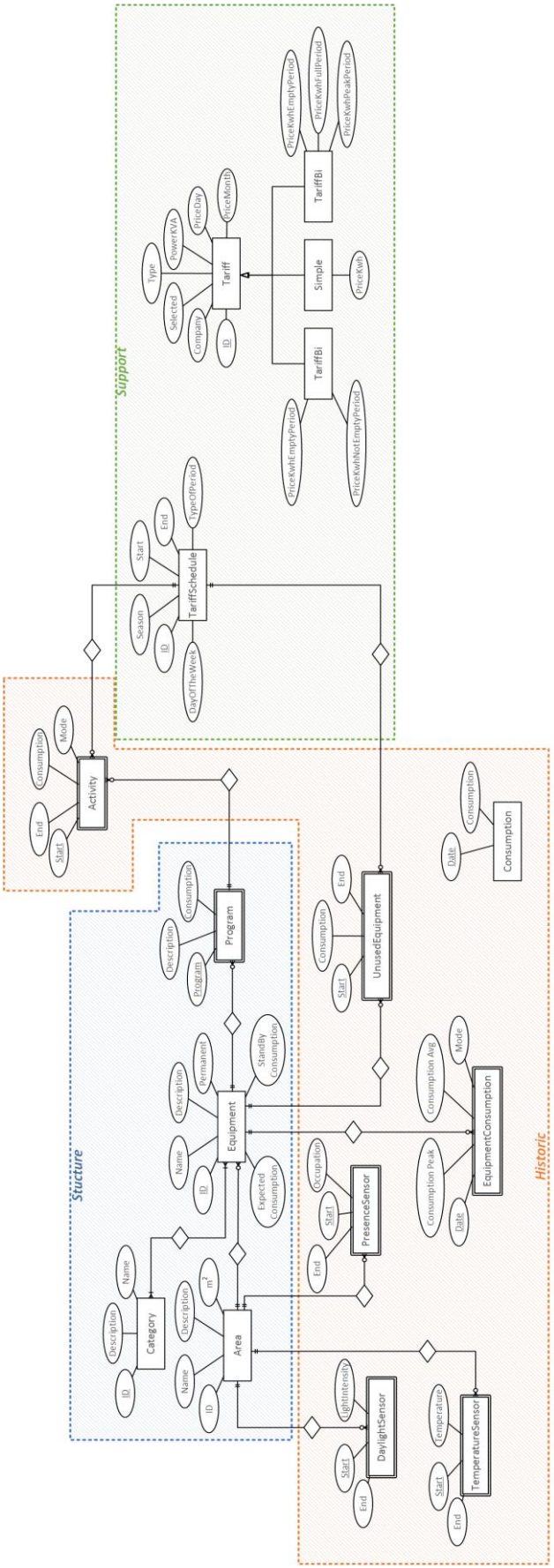


Figure 51 – Entity Relationship Model

9.9 USABILITY TEST

In this section is presented a copy of the Usability Test given to the participants.

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

(Please do not fill these fields)
Date: _____
Number: _____

A DSL for Energy Data Analysis [Part I]

Thank you for your participation, your feedback is important for us. Through this questionnaire, your answers will be helpful in enhancing our language. Your response will be used only for the purpose of this project.

Don't forget we are validating a tool, therefore, there is no wrong answers.

General Information

Age: _____

Sex: ☐ Male ☐ Female

Country: _____

Education

Education Level:

☐ Basic ☐ Secondary ☐ Bachelor
☐ Master ☐ Doctoral

What is your field of study?

_____

Experience

How would you rate your Database (SQL) experience?

None ☐ ☐ ☐ ☐ Expert

How would you rate your Spreadsheet (Excel) experience?

None ☐ ☐ ☐ ☐ Expert

Energy Consumption

Are you responsible for the payment of the energy bill?

☐ Yes ☐ No

Page 1 of 3

Figure 52 – Usability Test (Part I. Page 1)

Do you usually monitor your energy consumption?

☐ No ☐ Yes, through _____

Icons Validation

For each of the following icon, how would you rate the meaning that is associated to it, according to the description?

(Please circle the smiley that you consider most appropriate)

Palette Group	Icon ID	Icon	Icon Name	Icon Description	Immediate Association	Logical Association	No Association
Object	1		Container	Used to surround the energy query			
Connection	2		Connection	Performs the connection between the query elements			
	3		View Connection	Performs the connection between the query and the visualization form			
Visualization	4		Bar Chart	Presents the query result on a bar chart			
	5		Pie Chart	Presents the query result on a pie chart			
	6		Table	Presents the query result on a table			
Action	7		Monitoring	Used for observing an aspect			
	8		Notification	Alerts the user about some situation			
	9		Information About	Used to retrieve information			
Timeline	10		Now	Return the current information			
	11		Last 24h	Returns the information from the last 24 hours			
	12		Last Month	Returns the information from them last month			
	13		Last Year	Returns the information from last year			

Page 2 of 3

Figure 53 – Usability Test (Part I. Page 2)

Subjects	14		Equipment	Represents all the house equipment			
	15		Consumption	Represents the consumption of energy			
	16		Area	Represents the rooms in the house			
	17		Category	Represents the equipment's categories			
	18		Tariff	Represents the energy tariff			
Characteristics	19		Program	Selects only the equipment's programs			
	20		Working	Selects only the ones that are working			
	21		Stand-by	Select only the ones that are on stand-by			
	22		Estimated	Return an estimation			
Conditions	23		Higher	Sorts the information from the highest to the lowest			
	24		Abnormal	Filters the information by abnormal consumptions			
	25		Peak Period	Returns only the ones that work on peak period			
	26		Damaged	Returns only the equipment that are damaged			
	27		Unnecessary	Returns only the equipment that are being used when are not necessary			
	28		Best	Used to determine the best tariff			

Figure 54 – Usability Test (Part I. Page 3)

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

(Please do not fill these fields)
Date: _____

Number: _____

A DSL for Energy Data Analysis [Part II]

Thank you for your participation, your feedback is important for us. Through this questionnaire, your answers will be helpful in enhancing our language. Your response will be used only for the purpose of this project.

Don't forget we are validating a tool, therefore, there is no wrong answers.

Tasks to perform

In this section, we challenge you to use our tool to create some questions about the consumption.

Task 1 – What is the equipment's consumption in standby mode, from the last month?

How would you rate the difficulty of developing this task?

Easy ☐ ☐ ☐ ☐ Hard

How certain are you that the task was completed successfully?

0-20% ☐ 20-40% ☐ 40-60% ☐ 60-80% ☐ 80-100% ☐

Task 2 – What is the best tariff for me?

How would you rate the difficulty of developing this task?

Easy ☐ ☐ ☐ ☐ Hard

How certain are you that the task was completed successfully?

0-20% ☐ 20-40% ☐ 40-60% ☐ 60-80% ☐ 80-100% ☐

Task 3 – What are the areas with higher consumption in the peak period, from the last 24h?

How would you rate the difficulty of developing this task?

Easy ☐ ☐ ☐ ☐ Hard

How certain are you that the task was completed successfully?

0-20% ☐ 20-40% ☐ 40-60% ☐ 60-80% ☐ 80-100% ☐

Page 1 of 5

Figure 55 – Usability Test (Part II. Page 1)

Tasks on the Excel

In this section, we challenge you to create the same questions about the consumption, but using an Excel Sheet.

Task 1 – What is the equipment's consumption in standby mode, from the last month?

How would you rate the difficulty of developing this task?

Easy ☐ ☐ ☐ ☐ Hard

How certain are you that the task was completed successfully?

0-20% ☐ 20-40% ☐ 40-60% ☐ 60-80% ☐ 80-100% ☐

Task 2 – What is the best tariff for me?

How would you rate the difficulty of developing this task?

Easy ☐ ☐ ☐ ☐ Hard

How certain are you that the task was completed successfully?

0-20% ☐ 20-40% ☐ 40-60% ☐ 60-80% ☐ 80-100% ☐

Task 3 – What are the areas with higher consumption in the peak period, from the last 24h?

How would you rate the difficulty of developing this task?

Easy ☐ ☐ ☐ ☐ Hard

How certain are you that the task was completed successfully?

0-20% ☐ 20-40% ☐ 40-60% ☐ 60-80% ☐ 80-100% ☐

Page 2 of 5

Figure 56 – Usability Test (Part II. Page 2)

Query Identification

The following images represents two different queries.

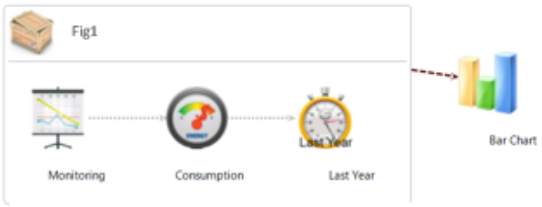


Figure 1

Figure 2

What the user wants to know in the Figure 1?

What the user wants to know in the Figure 2?

Page 3 of 5

Figure 57 – Usability Test (Part II. Page 3)

Query Patterns

Using this tool, do you think that would be possible to determine which categories of equipment have higher consumption?

☐ Yes ☐ No. Why? _____

Using this tool, do you think that would be possible to determine which equipment may be damaged?

☐ Yes ☐ No. Why? _____

Using this tool, do you think that would be possible to determine how much will you pay in the end of the month?

☐ Yes ☐ No. Why? _____

General Opinion

Select which of the previous tasks were the most difficult to perform?

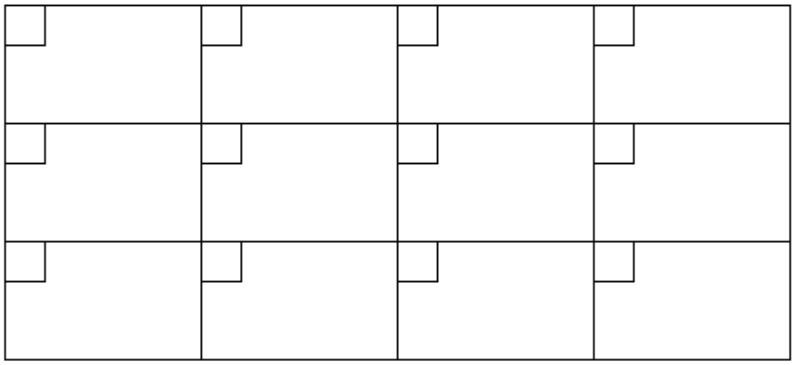
☐ None ☐ Task 1 ☐ Task 2 ☐ Task 3

Would you use this tool to monitor your energy consumption?

☐ Yes ☐ No. Why? _____

Which are the icons that you considered poorly chosen?

(Use the small square to indicate the icon number presented on the icon table and use the box to draw/describe a better option)



Page 4 of 5

Figure 58 – Usability Test (Part II. Page 4)

Additional Feedback

Please list any ideas in which our language could be improved.

Personal Information

Providing the following information is optional.

First Name Last Name

Email

Would you like someone to contact you regarding your responses on this survey?

☐ Yes | ☐ No

Thank you for taking the time to fill out our questionnaire. Your input is greatly appreciated.

Notes (Please do not fill this space)

Page 5 of 5

Figure 59 – Usability Test (Part II. Page 5)