XIS-Analytics

A Domain Specific Language for Data Analysis

Afonso Vitorino Mangas Pereira da Silva

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Alberto Manuel Rodrigues da Silva
Eng. André Filipe Oliveira Pinto Ribeiro

Examination Committee
Chairperson: Prof. António Manuel Ferreira Rito da Silva
Supervisor: Prof. Alberto Manuel Rodrigues da Silva
Member of the Committee: Prof. Bruno Emanuel da Graça Martins

June 2016
There are some people I want to thank for the development of this research work. I believe without their knowledge, expertise, motivation and support, it would not have been possible to complete this important stage of my education.

First, I would like to thank Professor Alberto Rodrigues da Silva, who supervised me throughout this work and always believed in me to accomplish the goals we have defined. His guidance, knowledge and support were a great contribution to the development of this work and, therefore, part of its quality is greatly due to him.

Secondly, I would also like to thank André Ribeiro for its guidance, support and friendship during this past year.

I would also like to thank my friends at IST who accompanied me throughout these years and with whom I have learned a lot and shared many moments of fun.

I would also like to thank my colleagues and students of Professor Alberto Rodrigues da Silva and other people who contributed somehow to the development of this work, namely during its evaluation phase.

I believe the outside of this work environment was really important to help me clear my head and gain new energy to pursue this work. Therefore, I owe that to my closest friends and I want to thank them for cheering me up at times when I was more discouraged.

A special thanks to my family, namely my parents, José and Maria do Carmo, my sister and brother, Mariana and José Pedro, and all my extended family, for allways believing in me despite the hard times and self doubt moments, and helping me throughout this journey. It is to you that I dedicate this work.
Abstract

This research work proposes a model-driven development approach for data processing and analysis, called XIS-Analytics. This approach comprises a domain specific language, defined as a UML profile (the XIS-Analytics language) and a companion software framework (the XIS-Analytics framework).

The XIS-Analytics language captures concepts from the domain of data analytics, more specifically from the data warehouse model, enabling the specification of these systems in understandable way by, even by non developer stakeholders. It divides mapping of the dataset, as well as the design of data visualizations in separate views, promoting a "separation of concerns" principle, reducing the complexity of the process.

The XIS-Analytics framework provides support to the language by automatically generating the source code from the XIS-Analytics visualizations, through Model-to-Text transformations. By doing so, it enables any stakeholder, even non developers, to perceive the information within the dataset.

Some preliminary evaluation of the approach has been made and the results are quite satisfactory. In a group of 12 users, using a scale of 1 to 5, users ranked XIS-Analytics language with a value of 4.38, XIS-Analytics framework with 4.25 and the generated data visualizations with 4.375.

XIS-Analytics is an innovative approach that uses the benefits of a model-driven development, to allow the user to interpret data in search for possible value.

Keywords: Data Analytics, Data Visualization, Domain Specific Language, Model-Driven Development
Resumo

Este trabalho propõe uma abordagem de desenvolvimento conduzido por modelos para o processamento e análise de dados, chamada XIS-Analytics. Esta abordagem compreende uma linguagem específica de domínio, definida como um perfil UML (a linguagem XIS-Analytics) e uma framework associada (a framework XIS-Analytics).

A linguagem XIS-Analytics capta conceitos no domínio da análise de dados, mais especificamente os conceitos do modelo das data warehouses, permitindo a especificação destes sistemas de forma compreensível partes interessadas, mesmo que não sejam developers. As vistas estão separadas por mapeamento dos dados e definição da especificação das visualizações de dados, promovendo o princípio de “separação de conceitos”, reduzindo assim a complexidade do processo.

A framework XIS-Analytics fornece suporte para a linguagem, gerando automaticamente o código-fonte das visualizações XIS-Analytics, através de transformações de modelo para texto. Ao fazer isto, o XIS-Analytics permite que qualquer das partes interessadas, mesmo não developers, possam entender e interpretar as informações contidas do conjunto de dados.

Alguma avaliação preliminar da abordagem foi feita e os resultados são bastante satisfatórios. Recorrendo a um grupo de 12 pessoas, usando uma escala de 1 a 5, o grupo atribuiu em media os resultados de 4.38 a Linguagem XIS-Analytics, 4.25 a framework XIS-Analytics e 4.375 as visualizações geradas.

O XIS-Analytics é uma abordagem inovadora que utiliza um desenvolvimento conduzido por modelos para oferecer ao utilizador a possibilidade de interpretar dados, procurando algum possível valor.

Palavras-Chave: Analise de Dados, Visualização de Dados, Linguagens Específicas de Domínio, Desenvolvimento Conduzido por Modelos
Contents

List of Tables xi
List of Figures xiii
Acronyms xvii

1 Introduction 3
  1.1 Context 4
  1.2 Problem Definition 4
  1.3 Proposed Solution 5
  1.4 Thesis Statement 6
  1.5 Methodology 7
  1.6 Publications 9
  1.7 Outline 10

2 Background and Related Work 11
  2.1 Model-Driven Development 11
    2.1.1 Model Driven Engineering 11
    2.1.2 Model Driven Architecture 12
    2.1.3 Microsoft Software Factories 12
    2.1.4 Modeling language and metamodelling 14
    2.1.5 Domain Specific Language 16
    2.1.6 UML Profile 17
    2.1.7 Model Transformations 17
  2.2 Basics on Data Visualization 19
    2.2.1 Data Visualization Tools 21
  2.3 Related Work 21
    2.3.1 Data Modeling and Data Analytics 22
      2.3.1.1 JSON Discoverer 27
    2.3.2 Similar Works 28

3 XIS-Analytics Approach 31
  3.1 XIS and XIS-Mobile 31
  3.2 Overview 33
  3.3 Case Studies 35

4 XIS-Analytics Language 37
  4.1 Domain Entities View 37
  4.2 Data Analytics View 39
# List of Tables

2.1 Comparison between four template-based code generators\[1\]. . . . . . . . . . . . . . . . 19  
2.2 Comparison between data visualization available on different JavaScript charting frameworks. 22  
2.3 Comparison of the approaches from the Data Modeling perspective\[2\]. . . . . . . . . . . . 26  
2.4 Comparison of the approaches from the Data Analytics perspective\[2\]. . . . . . . . . . . . 27  
5.1 Comparison between four template-based code generators\[1\]. . . . . . . . . . . . . . . . 47  
6.1 Different data types combinations that the charts of XIS-Analytics can process. . . . . . 52  
6.2 Survey’s average score (scale of 1-5) for each XIS-Analytics Framework aspect. . . . . . 56
List of Figures

1.1 Simplified development approach with the XIS-Analytics framework. ........................................ 5
1.2 Action Research Methodology cycle[3] ...................................................................................... 7

2.1 MDE, related terminology and concrete approaches and initiatives[4]. ...................................... 13
2.2 Overview of the MDA approach[5] ......................................................................................... 14
2.3 The relationship between a model and a metamodel[5] ............................................................. 15
2.4 OMG’s MOF four-layered metamodel architecture[6] ............................................................... 16
2.5 Example of a UML profile definition (left) and application (right)[7] .......................................... 17
2.6 Anscombe’s Quartet, to the left the four different data set represented as a table, and to the right the corresponding visual representation[8] .......................................................... 19
2.7 The different data types[9] ....................................................................................................... 20
2.8 Example of three data models at different abstraction levels[2] ................................................. 23
2.9 Star Schema Model example ..................................................................................................... 24
2.10 Representation of cube operations: Slice (top-left), Dice (top-right), Drill Up/Down (bottom-left) and Pivot (bottom-right)[2] ................................................................. 25
2.11 Example of JSON structure used as input for XIS-Analytics .................................................. 26
2.12 Overall view of the JSON Discoverer approach (main phases represented in black boxes and input/output data represented in white boxes) ....................................................... 28

3.1 Multi-view organization of the XIS profile[10] ........................................................................... 32
3.2 Multi-view organization of XIS-Mobile[10] ............................................................................ 33
3.3 Multi-view organization of XIS-Analytics. .............................................................................. 34

4.1 Metamodel of the Domain Entities View ................................................................................... 38
4.2 Domain Entities View of the IMDB dataset from case study A ................................................ 38
4.3 Metamodel of the Data Analytics View .................................................................................... 39
4.4 Data Analytics View of the IMDB dataset from case study A (with the tagged values) .............. 40
4.5 Data Visualization for the XisDataAnalyticsUseCase “IMDB Movies Analysis” from Figure 4.2 40
4.6 Data Visualization for the XisDataAnalyticsUseCase “IMDB Movies Analysis 2” from Figure 4.2 40
4.7 Alternative Data Visualization for the XisDataAnalyticsUseCase “IMDB Movies Analy-

sis” from Figure 4.2 ...................................................................................................................... 41

5.1 Development process with the XIS-Analytics Framework ...................................................... 43
5.2 Profile diagram of XIS-Analytics’s diagrams ........................................................................... 45
5.3 Profile diagram of the toolbox of the Domain Entities View of XIS-Analytics ......................... 45
5.4 Example of the XIS-Mobile project structure making use of case study A. .............................. 46
5.5 Organization of the Acceleo project ........................................................................................... 48
5.6 JSON Discoverer Plug-in adaptation architecture ................................................................... 49
Acronyms

**ATL**  ATLAS Transformation Language. 18

**BPMN**  Business Process Modeling Notation. 17

**CMOF**  Complete MOF. 15

**CSV**  Comma-separated values. 46

**CWM**  Common Warehouse Metamodel. 15

**DSL**  Domain Specific Language. 5

**DSML**  Domain Specific Modeling Language. 14

**DW**  Data Warehouse. 6

**EA**  Enterprise Architect. 5

**EMF**  Eclipse Modeling Framework. 5

**EMOF**  Essential MOF. 15

**ER**  Entity-Relationship. 23

**ETL**  Extract Transform Load. 24

**IDE**  Integrated Development Environment. 12

**IDSS**  Information and Decision Support Systems Lab. 4

**IMDB**  Internet Movie Database. 35

**JET**  Java Emitter Templates. 18

**M2M**  Model-to-Model. 12

**M2T**  Model-to-Text. 5

**MARTE**  Modeling and Analysis of Real-Time and Embedded systems. 17

**MDA**  Model Driven Architecture. 12

**MDD**  Model-Driven Development. 3

**MDE**  Model-Driven Engineering. 3
MDG  Model Driven Generation. 44

MDX  Multidimensional Expressions. 24

MOF  Meta Object Facility. 15

MOFM2T  MOF Model to Text Transformation Language. 15

MSc  Masters degree in the area of Science. 4

NoSQL  Not Only SQL. 24

OLAP  Online Analytical Processing. 23

OLTP  Online Transaction Processing. 23

OMG  Object Management Group. 12

PhD  Doctor of Philosophy. 4

PIM  Platform-Independent Model. 12

QVT  Query-View-Transformation. 18

UML  Unified Modeling Language. 5

XIS  eXtreme modeling of Interactive Systems. 31

XMI  XML Metadata Interchange. 15
Chapter 1

Introduction

We live in a time where the amount of data that is generated every day for the past few years creates massive and complex datasets, which are continually growing larger and larger. In just two days we create as much information as we did from the beginning of the existence of human civilization up until 2003\[11\]. A few years ago, just the storage of all of the data was an issue, challeging all technologies with a scalability crisis\[11\]. Currently, we are able to handle the store of such a volume of data, but the data analysis technologies are not quite there yet. Enterprises and individuals store all the data they can so they can process and analyze all of it, hoping to obtain some edge in the process\[12\][13]. As such, by exploring this big datasets, this enterprises and individuals try to discover facts they did not know before. From this discovery, the benefits can come in many forms, such as faster and better decision making, as well as the having the edge on coming up with new products and services or re-developed ones that can be improved. However, although this is an important task from a business perspective, it represents a big risk to blindly invest time and money in the analysis and exploration of any dataset that we might come across, even more true given the recent economic recession. Even taking the risk, no matter the type of organization, they need the right tools, rigorous processes and some sort of discernment to generate return with the data analysis, since the activity of converting this data analysis projects into knowledge and business value is a hard process where many fail. We currently live in a data-driven world, and the difference between winning and losing on it will be the ability to reduce the costs of, not only the technologies to manage the increasing volume of data, but the capacity to determine if a dataset has value or not.

The more traditional ways of processing data, even do most of them have been adapted to the new paradigm of data (in terms of volume, variety and veracity), have had limited success at tackling this problem. So, new solutions had to be created. Many of these solutions to address data analysis and exploration have been developed in the past few years, but do the the nature of the advantage that this activity can give over business competitors, and the investment it was need to develop them, the best solutions fall into the category of proprietary software. As such, only organizations or individuals that possess budget, capabilities or knowledge tend to risk in exploring the data they might have.

Recently, there have been several solutions being developed, thus attacking this issue, however, not many making use of an emerging area of Software Engineering, Model-Driven Engineering (MDE)\[14\] like the one suggested in this dissertation. MDE, or its development process, Model-Driven Development (MDD)\[15\], that seeks to mitigate the issue of the complexity of some tasks by considering models as the first-citizen artifacts, while also granting the possibility of source code being generated automatically from those models through model transformations.
1.1 Context

This research work has been conducted at the IDSS (Information and Decision Support Systems Lab) group of INESC-ID (Instituto de Engenharia de Sistemas e Computadores - Investigaç~ao e Desenvolvimento) under the supervision of Professor Alberto Rodrigues da Silva, in the scope of the Master Degree in Information Systems and Computer Engineering at Instituto Superior Técnico. This research work results from the interest in the area of MDD and its application to data analysis and exploration. The IDSS is coordinated as a single research group, with nearly 25 senior and associate researchers, about the same number of junior researchers (PhD students), and a dynamic team of MSc students and visiting scholars.

This dissertation was developed in the scope of the DataStorm project, taking part on the task nominated H6-Domain Specific Languages for Largescale-Data Applications. DataStorm is a project that aims to tackle some of the most pressing challenges and societal needs of the present in information technology. The project itself focus on creating a mass of scientist and engineers to address the design, implementation and operation solutions for the current problem with massive data collection systems. The project is composed by horizontal and vertical work-packages. The first address more common research challenges associated with the analysis of large datasets of heterogeneous and imprecise network data. On the other hand, vertical work-packages involve the use of research team have been working. DataStorm aims to provide a very competitive technology to expand the existing competencies in data and knowledge mining.

1.2 Problem Definition

As introduced previously, the rapid growth of data storage generated the need to new solutions for data analysis and exploration to emerge. Data analytics can be a obstacle to many organizations due to, besides possible lack of knowledge or resources, the complexity of the task itself. Due to this impairment, opportunities for some original ideas, innovation or new products might never happen. It is important to supply an alternative to any one who might have the initiative to explore data with the tools to do so. To decide if a set of data is valuable or not, the integration of the human judgement is usually the best way, and to do so, representing those datasets by means of visual representation and interaction techniques in the analysis process the way to address such an issue.

Thus the problem addressed in this dissertation is: Data Analysis and Exploration Complexity. XIS-Analytics intends to work as a solution for this. Given this problem, the following research questions summarize the situation to address:

- **RQ1**: How to specify and map a dataset, in the most possible loosely coupled way relative to the usual database concepts?
- **RQ2**: Which concepts are specific of data visualization design that can be included in a modeling framework?
- **RQ3**: Which views should be used to map the dataset and specify data visualization with the priority on mitigating the complexity of the task?
- **RQ4**: How to generate data visualization, even if the specifications are less correct, i.e., if the user as no idea how to perform the specifications?

\[1\]https://www.inesc-id.pt/group.php?grp=II11

\[2\]http://dmir.inesc-id.pt/project/DataStorm
The ideal scenario would be for an user with little knowledge of the dataset, to specify the data visualization and obtain immediate value at the first processing the data from the interpretation of the visualizations. Also, the data visualizations should be scalable, interactive and responsive, so the user could explore all types of data, interact with the visualizations how he sees fit and adapted to all kinds of devices, respectively. The use of technologies such as HTML5, CSS3 and JavaScript libraries, as well as approaches based on MDD\cite{16}, like the one proposed in this dissertation, are examples of solutions focused on solving these problems.

1.3 Proposed Solution

This research proposed XIS-Analytics, a Domain Specific Language (DSL) and its supporting framework as an innovative solution to the problems presented in the previous section, making use of a MDD approach.

The XIS-Analytics language is a graphical DSL in the form of a UML profile that allows for the visual representation of a dataset, through the use of data visualizations and concepts specific to data analysis and its representation. XIS-Analytics language has a two view organization proposing a solution where the first view is to map the dataset and the second view is to specify how that dataset should be represented for analysis.

The usability of this language has been designed through the implementation of a supporting MDD-based framework developed by using available technologies, such as the Sparx systems Enterprise Architect (EA)\cite{3} and the Eclipse Modeling Framework (EMF)\cite{4}. As it can be seen in Figure 1.1, the XIS-Analytics Framework is comprised by three major modules: (1) Visual Editor, which allows the definition the models to perform both the mapping of the data and the specifications of the visualizations; (2) Model Validator, in charge of checking if the models produced by the user while using the Visual Editor are valid; and (3) the Code Generator, which generates the source code of the data visualizations with the specifications defined by the user, through Model-to-Text (M2T) transformations based on code templates specified using Acceleo\cite{5}, a plug-in compatible with the EMF.

The validity of the data visualizations produced by the user while using XIS-Analytics is entirely in the hands of the user, since they are the ones that decide if they answer to the question or not.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{xis-analytics-framework.png}
\caption{Simplified development approach with the XIS-Analytics framework.}
\end{figure}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Header 1 & Header 2 \\
\hline
Row 1, Column 1 & Row 1, Column 2 \\
\hline
Row 2, Column 1 & Row 2, Column 2 \\
\hline
\end{tabular}
\caption{Example table}
\end{table}

4http://www.eclipse.org/modeling/emf
5https://www.eclipse.org/acceleo}
solution tries to generate the visualization no matter how invalid the specifications might be, and by doing so, it returns to the user something from where conclusions can be taken, allowing him to reiterate the models and the specifications to further continue exploring the data. XIS-Analytics can not determine if a dataset might be valuable or not, and defends that the integration of the human judgement is usually the best way take conclusions on this.

After presenting the proposed solution, it is important to clearly highlight the goals of this dissertation namely:

• **G1**: Collect and analyze the related work done in the scope of data modeling and data analysis with the purpose of visually representing data.

• **G2**: Design and implement a Domain Specific Language that allows the empowers its user with the capabilities to map, analyze and represent the input structured data so he can interpret and seek for conclusions and value.

• **G3**: Implement the tools that support that language, namely: (1) A visual editor that allows the mapping of the dataset as well as the specification of the data visualization, (2) a model validator to guarantee the validity of the produced models, and (3) a M2T generator that parses the models specified in this language and generates automatically the code of the data visualizations.

• **G4**: Evaluate the quality of the proposed solution through the implementation of some case studies of datasets and by an user session assessment.

1.4 Thesis Statement

This dissertation’s thesis states that is possible to analyse a structred dataset, by means of a MDD approach, generating data visualization so the user, no matter the previous knowledge of formal quering data, can take conclusions and assumtions about the dataset, making use of the data visualizations to explore it. An even if the first generated data visualizations do not help concluding anything, repeating the process and generating more and more visualizations could.

In particular, I claim that this can be achieved by relying on the development of the interactive, responsive and scalable data visualizations that the XIS-Analytics language and framework generates. Interactive by the process it self, as well as the properties of teh generated visualization, responsive becuse the visualization adapt to diferent devices and scalable becuse the generated code was desired to handle a big data set. By requesting from the user to map the dataset used as an input, and doing so by modeling views with concepts similar to those found in Data Warehouse [DW][17] models, followed by specifying how the user wants his data to be represented, I claim that this user can explore his dataset. By using XIS-Analytics, the user has immediate insight to the content of the data, and even if it this is not the case, the user will eventualy gain the insight, even if it is do to the repetition of the process. Therefore, i also claim that the use of XIS-Analytics contributes to an initial assessment of the value of a dataset

Furthermore, it was possible to develop two case studies that allow exploration of the dataset by making use of the proposed approach, making the dataset go from JSON, to model until source code of the respective data visualizations. In addition, a user session contributed to assess the acceptance of the development process, as well as data visualization themself, while raising awareness of the participants on what this approach can offer.
1.5 Methodology

As suggested previously in the project presentation, this research work has been conducted in an iterative and gradual way following the Action Research methodology. As illustrated in Figure 1.2, this methodology suggests a cyclical process composed of five steps executed in a certain scope, known as client-system infrastructure (or research environment). These steps are:

- **1 - Diagnosing**: Represents the identification of the problem domain and the motivation for its relevance;
- **2 - Action Planning**: Represents the planning and definition of the proposed solution and the necessary changes to solve/relieve the problem identified in the previous step;
- **3 - Action Taking**: Consists in the implementation of the solution planned in the previous step;
- **4 - Evaluating**: Represents the evaluation/assessment of the solution developed in the previous step, namely analyzing if the actions performed have succeeded to solve/relieve the problem.
- **5 - Specifying Learning**: Represents the lessons learned during the cycle, and in the preparation of the next iteration of the action research cycle if needed.

Particularly, this research work can be subdivided in three iterations of the Action Research methodology:

**Iteration 1 - Month 1 to Month 7**

At the early stage, the first iteration after the first presentation of the dissertation started with redefining the scope and goals of this thesis, since it was pointed out that they should be more specific. After redefining the scope and goals to what was presented in Section 1.3, the rest of this iteration mostly consisted studying XIS and XIS-Mobile as well as reviewing and updating the related literature and work, specially in the fields of MDE, data processing, data analysis and data visualizations. The review task resulted in the publication of a survey regarding the current state of data modeling and data
analytics across the three main approaches: operational databases, decision support databases and Big Data technologies[2]. Then, the available technologies were explored to find which could support or help the definition and use of the XIS-Analytics profile. The choice of the development environment fell on the environment provided by the Eclipse IDE and two plugins, Papyrus and Acceleo. For model definition it was decided to design it similar to the Data Warehouse (DW) star schema model. At last, a data set and a case study were defined (Case Study A, Section 3.3), to evaluate the design of the language and framework. Next the five steps of the Action Research methodology for the first iteration are summarized.

- **Diagnosing**: Review of the XIS and XIS-Mobile, as well as the existing literature and technologies that could help with the definition of the XIS-Analytics language and framework.

- **Action Planning**: Definition of the organization and concepts of a preliminary version of the XIS-Analytics language. Definition of the architecture and components of a preliminary version of the XIS-Analytics framework.

- **Action Taking**: Specification of the preliminary XIS-Analytics language as a UML profile using the Papyrus plugin for the Eclipse IDE. Early stages of the implementation of the XIS-Analytics framework using Papyrus for specifying the XIS-analytics language and Acceleo for defining the initial code templates for the data visualizations.

- **Evaluating**: Explore the options for case studies, choose one, define it and start testing the validity of the preliminary design and implementation. The chosen dataset was the IMDB (Case Study A, Section 3.3).

- **Specifying Learning**: Success on creating a case study dataset using Pentaho’s spoon to build it. Success on modeling the case study with the XIS-Analytics language, using Papyrus. Then, Acceleo offered a good support to perform the Model-to-Text transformations and allowed the generation of dynamic code for the data visualizations, expressing the specifications defined in the models. Even though it was documented by XIS-Mobile, even while using a more advanced version of Papyrus, still came to the same conclusion that it is not suitable to model a language with the complexity of XIS-Analytics, due to still possessing bugs that hinder the modeling process. After the first generations, the need to improve the M2T templates and add more options was a clear need for the next iteration.

**Iteration 2 - Month 5 to 12**

The second step, after the planning and initial design of XIS-Analytics in the previous iteration, consists in the refinement of the XIS-Analytics language, and in the final implementation of the XIS-Analytics Framework. The XIS-Analytics language was subjected to major changes regarding some issues and limitations found in the previous iterations. To solve some of the limitations, it was decided that the input should have a specific structure. Since Papyrus could not provide the desired user experience, it was decided to migrate from the Eclipse environment to Sparx Enterprise Architect (EA) since it provides a more sophisticated and refined environment. Additionally, model validator have been implemented to enhance the quality of the designed models while subsequently improving the quality of the specified data visualizations. This iteration originated the submission of a paper to the QUATIC 16 which is still pending for confirmation. Next, the five steps of the Action Research methodology are summarized for this iteration.

- **Diagnosing**: Analysis of the issues and needs discovered in the previous iteration. Research on improvements to the profile views and data visualizations.
• **Action Planning:** Definition of the concepts that should be fixed and added to the XIS-Analytics language. Review on the architecture of XIS-Analytics framework through the change to EA’s environment and addition of the model validation. Definition of the rules that should be triggered during model validation.

• **Action Taking:** Reorganization and correction of the XIs-Analytics language using EA. Implementation of the model validator feature on the XIS-Analytics framework.

• **Evaluating:** Try different specifications for the Case Study A on both views of the XIS-Analytics, regarding different mapping alternatives, as well as different specifications for data visualizations.

• **Specifying Learning:** Success on applying the Case Study A in its various forms to XIS-Analytics. Success on the implementation of the model validator and the code generator. EA provided a very good environment to use XIS-Analytics. Need to define new case studies that exercise further the concepts of the XIS-Analytics framework. Documented a limitation while using Google Chrome because of the Anti-XSS filter not letting the JSON file be loaded.

**Iteration - Month 11 to 12**

The last iteration essentially consisted in the evaluation of XIS-Analytics by third parties. First, it was defined how the evaluation would occur and the aspect to be analyzed. The evaluation has been performed by conducting a user session and focused on three aspects: Language, Framework and Generated Data Visualizations. For that purpose, a second case study (Case Study B, Section 3.3) has been defined which should be modeled and explored by the users during the session. In addition, a survey has created with the objective of gathering the opinions of the participants. The survey was to be filled at the end of the evaluation session, after all the participants tested XIS-Analytics. After the sessions, it was carried out an analysis of the results obtained from the survey, as well as the issues expressed by the participants midsession. This analysis contributed to immediate improvements to the Model Validator as well as plans for future work.

• **Diagnosing:** Identification of the needs of XIS-Analytics in terms of evaluation

• **Action Planning:** Definition of the aspects to be evaluated. Definition of how the evaluation session would occur and the type of participants involved.

• **Action Taking:** Specification of the user session guide, including the new case study and the session rules. Elaboration of the participation survey.

• **Evaluating:** Realization of the evaluation session with users.

• **Specifying Learning:** Analysis of the results obtained from the survey, as well as the behaviors expressed by the participants during the session. The surveys collected positive results and showed XIS-Analytics’s usefulness and feasibility as proof of concept. Still, XIS-Analytics can still be improved in terms of the application of the concepts and by adding more types of data visualizations.

### 1.6 Publications

During this research one paper was published to an international conference, presented respectively:
• 1. André Ribeiro, Afonso Silva and alberto Rodrigues da Silva. 2016. Data Modeling and Data Analytics: A Survey from a Big Data Perspective. Journal of Software Engineering and Applications. Scientific Research Publishing. This paper describes and analyzes the current state of data modeling and data analytics, while clarifying the main differences between the three main approaches concerned with these aspect, namely: operational databases, decision support databases and Big Data technologies.

Also, one other paper was submitted to the 10th International Conference on the Quality of Information and Communications Technology (QUATIC 16) which is currently pending. This paper describes the XIS-Analytics language and framework. Part of the content of this paper is contained in Chapter 4 and 5.

1.7 Outline

The remainder of this dissertation is organized as follows:

• **Chapter 2:** This chapter provides an overview of the main concepts that underline this research work. Particularly focusing on three main topics: Model-Driven Development (MDD) and Data Visualizations. Some related work on the subjects that XIS-Analytics fit in is presented in this chapter as well.

• **Chapter 3:** This chapter presents an overview of the approach proposed in this dissertation, known as XIS-Analytics approach. Also, the chapter provides an insight on how the research for the solution went about, while making notice of the relevant findings for the planning of the XIS-Analytics approach.

• **Chapter 4:** This chapter presents the XIS-Analytics language, a DSL that allows for a user to explore a dataset through the use of data visualizations.

• **Chapter 5:** The chapter describes the XIS-Analytics framework, explaining the integrated environment that supports the Model-Driven Development (MDD) of the data analysis and data visualizations using the XIS-Analytics language.

• **Chapter 6:** This chapter presents and discusses a twofold evaluation performed to XIS-Analytics.

• **Chapter 7:** This chapter presents the main conclusions of this work along with the future work perspectives.
Chapter 2

Background and Related Work

This chapter provides an overview of the main concepts that underlie this research work and the related work on them. These concepts are divided into three main topics: Model-Driven Development (MDD), basic on Data Visualization and related work regarding those two topics.

First, Section 2.1 describes the basic concepts around the MDD field. As such, Section 2.1 describes MD, Model-Driven Engineering (MDE) and related initiatives, explains concepts such as metamodeling, modeling language, domain-specific language, and UML profile, and details the types of model transformations used in MDD.

Then, Section 2.2 describes the basic concepts around the data visualization in terms of choosing the representation and how to use it, along with some basic concepts regarding the representation of the data itself are given as well. At the end of the section some insight to the most used tools is provided as well.

Finally, Section 2.3 describes the Related Work some of the research done to build the XIS-Analytics as well as presenting some of the solutions that are most similar to the one addressed in this dissertation.

2.1 Model-Driven Development

This section presents some concepts around the Model-Driven Development (MDD) area. Section 2.1.1 describes what is Model-Driven Engineering (MDE) and Model-Driven Development (MDD), while Section 2.1.2 presents initiatives of MDD. Then, Section 2.1.3 goes in depth of concepts of metamodeling and modeling language. In turn, Section 2.1.4 and Section 2.1.5 the concepts of domain-specific language (DSL) and UML profile, are respectively defined. Finally, in Section 2.1.6, the model transformations commonly used in MDD, are explained, even though XIS-Analytics currently only makes use of one of them, as it will be elaborated later on this dissertation.

2.1.1 Model Driven Engineering

Model-Driven Engineering (MDE) is a software development methodology that considers domain models as first-class entities. By recurring to models like such, MDE seeks to move the usual source code development process to a more abstract level of specification with the purpose of mitigating software complexity. To represent a certain domain problem, domain models are built, consisting in abstract representations of concepts specific to said problem. A model can even be classified as descriptive or prescriptive, depending on the use that is made of it. A descriptive model is generally only used as documentation of the system it details, while a prescriptive model besides describing the system, it is
also used to develop the system. Thus, using this terminology, MDE defends the use of prescriptive models.

MDE has as its main goal to make use of the models to guide all the development activities, from system design, code generation and deployment until the system maintenance. By using this abstraction, some results can be observed such as quality improvements, increased productivity and improved communication with domain experts, as well as programmers\textsuperscript{15}. By making use of concepts closer to the domain problem, MDE also tries to reduce the time to market. Other of the greatest benefits of MDE is the ability to specify the structure and the behavior of a software system in a more platform agnostic way than the traditional programming approaches.

Having said that, because it is concerned with other model-based activities of a complete software engineering process, besides the development tasks, MDE is considered a software engineering discipline. For instance, model-driven reverse engineering and model-driven evolution are examples of such activities. Thus, it becomes clear that Model-Driven Development (MDD) is a subset of MDE, because it only involves the model-based development activities, meaning, the generation of the system through models. Furthermore, MDD is materialized through some initiatives. Two of the most popular MDD initiatives are the Model Driven Architecture (MDA), proposed by the Object Management Group (OMG), and the Software Factories, proposed by Microsoft. To contribute to a better understanding of the MDD initiatives, both are briefly described below in sections 2.1.2 and 2.1.3, respectively.

Concluding, Figure 2.1 summarizes the range of action of MDE, MDD, MDA and Software Factories.

### 2.1.2 Model Driven Architecture

Model Driven Architecture (MDA) is the initiative proposed by the Object Management Group (OMG) to develop software through a MDD approach. A set of guidelines and principles to specify a system based on models is provided by MDA. These models can reside at different levels of abstraction, each one emphasizing a certain aspect of the system. MDA considers two main types of models: (1) the Platform-Independent Model (PIM) and (2) the Platform-Specific Model (PSM). A PIM is a specification of a system with a high level of abstraction expressed in a platform-independent way, this is, in a model independent of the technology used. On the other hand, while PSM is also a specification of a system, it is in a platform-specific way, being that a PSM specifies how a system uses a concrete type of platforms. That being said, a PIM can be translated to one or more PSMs through the use of Model-to-Model (M2M) transformations which map the PIM with some implementation language or platform (for example, Java or C++) according to the already predefined rules. The last step consists in the transformation of each of the PSMs in source code of the respective platform. These transformations are designated by Model-to-Text (M2T) transformations. The goal of MDA is to provide system specification and interoperability providing a description of a system in a platform independent way(ref).

### 2.1.3 Microsoft Software Factories

Microsoft Software Factory is a product line, or a collection of software, used to create specific kinds of applications. It may be composed of processes, templates, Integrated Development Environment (IDE) configurations and views.

A Software Factory contains three main ideas: a Software Factory Schema, a Software Factory Template and an Extensible Development Environment. The Schema is like a “recipe” that describes the product line architecture and the key relationships between the components and frameworks of which it is comprised. The Template provides elements, such as DSLs, patterns, templates, editors, frameworks or samples, used to build the final product. Finally, the Extensible Development Environment
Figure 2.1: MDE, related terminology and concrete approaches and initiatives[4].
becomes a software factory when it uses the configurations defined in the Template. Unlike MDA, Software Factories are not so worried with portability and platform independence. Instead, they are more focused in productivity with the goal of reducing the costs and time to market.

### 2.1.4 Modeling language and metamodeling

In MDD, the used models can have different representations and even have different domain concepts defined, but all of them are specified through a modeling language. A modeling language is defined by three main components: (1) the abstract syntax, (2) the concrete syntax and (3) the semantics.

In the abstract syntax is where the set of concepts provided by the modeling language and the relationships among themselves are specified. Like so, the abstract syntax is a model itself, being then considered the metamodel. This means that a metamodel provides a language in which the model is specified. The concrete syntax, also known as notation, defines the representation of the concepts of the modeling language, being that its representation can be either textual and/or visual. Finally, the semantics describe the meaning of each concept defined in the abstract syntax that could not be captured by the metamodel. Semantics is regularly specified through the definition of rules which restrict the possibilities of use between each language element, having as objective the prevention its invalid use. These rules can be specified using a rule specification language, like OCL[18], and using a natural language specification which is more quickly and easily understood by model designers.

Therefore, metamodeling is a crucial activity on MDD, because it defines the metamodel used to create modeling languages. Figure 2.3 summarizes the relationship between these concepts.

A modeling language can be classified as either a General-Purpose Modeling Language (GPML) or a Domain Specific Modeling Language (DSL/DSML). The use of GPMLs or DSMLs is a recurrent topic of discussion in the area of MDE. A GPML is characterized by having a greater number of generic constructs compared to its counterpart, thanks to which, it encourages a wider and widespread use in different fields of application. The Unified Modeling Language (UML) specified by the OMG, is a prime example of that kind of modeling language, which provides a large set of constructs and notations.
These are primarily for specifying and documenting software systems according to the object-oriented paradigm. On the other hand, DSLs provide a better description for the domains in which they are applied. Thus, DSLs use constructs that are closer to the most significant concepts of its application domain. Therefore, by using DSLs it is possible to define models that capture more details of the specific domain problem while simultaneously doing it in a more expressive way. However, UML also provides the Profile mechanism that allows extending its concepts (metaclasses), allowing them to be adapted to specific domain problems. Both the topics of DSL and UML Profile are further described with more detail in sections 2.1.5 and 2.1.6, respectively.

Quite a few metamodeling approaches are available nowadays, namely because MDD does not obliges the use of any specific approach in particular\[7\]. Nevertheless, the use of standardized approaches is of good practice, while eases and fosters the adoption of the modeling language to be developed. The most popular metamodeling approach is the Meta Object Facility (MOF) proposed by the OMG. MOF was conceived with the goal of being able to prove a metamodel in which OMGs standards could be based. Some examples of MOF-based languages are UML, XML Metadata Interchange (XMI), MOF Model to Text Transformation Language (MOFM2T) and Common Warehouse Metamodel (CWM)\[19\]. MOF comprises two variants: Essential MOF (EMOF) and Complete MOF (CMOF). EMOF is a subset of MOF that provides the capabilities that can be found in object-oriented programming languages and XML. For instance, EMOF is the metamodel of the XMI language. On the other hand, CMOF is the result of merging EMOF with its extensions, meaning that CMOF provides the concept of Association. CMOF is the metamodel used to specify UML2.

As shown in Figure [2.4] MOF is designed as a four-layered metamodel architecture where each layer conforms to the one above it:

- The M3 layer corresponds to the meta-metamodel layer, in which MOF is the meta-metamodel, i.e., MOF is defined by itself.

- The M2 layer corresponds to the metamodel layer, in which the metamodel is an instance of MOF. For instance, the most commonly used is the UML metamodel.

- The M1 layer corresponds to the model layer, in which the metamodel concepts are used to define the user model. Taking in consideration that if the UML metamodel was the one defined in the M2 layer, then this layer would use UML concepts (example: Class, Attribute or Operation) to define the model.

- At last, the M0 layer corresponds to the system it self, in which runtime instances are defined using the model elements specified in the M1 layer.
To define its models, as an OMG’s standard, MDA highly relies on MOF (PIMs and PSMs). Specifically, UML is commonly used for specifying the PIM. Doing so guarantees that the models can be stored in a MOF-compliant repository, parsed and transformed by MOF-compliant tools, and finally rendered into XMI for transport over a network. Additionally, Ecore5, another variant of MOF, has been defined in the context of the Eclipse Modeling Framework (EMF). Ecore is similar to the EMOF and is used to represent models and their runtime support in EMF. For instance, the UML implementation used in Eclipse is defined by Ecore.

### 2.1.5 Domain Specific Language

A Domain Specific Language (DSL) is a language which was tailored to attend a specific set of tasks which, with appropriate notations and abstractions (either textual or graphical), depicts the concepts of a particular problem domain[16]. In particular, Domain Specific Modeling Languages (DSMLs), like the one proposed in this dissertation, abstract even more the problem domain, since they usually represent their concepts in a more graphical way. DSLs are usually smaller than the usual programming languages and can usually offer several advantages.

Considering that a DSL is expressed using domain concepts, it is usually easier to read, understand, communicate and validate with, easing the cooperation between programmers and domain experts. Moreover, in the literature is mentioned that DSLs can improve productivity, reliability, maintainability and portability [20][21]). Yet, making use of a DSL can raise some problems, such as the cost of learning, implementing and maintaining the new language created, as well as assure the support tools use to develop with it. Some popular examples of DSLs are SQL for databases, R for statistics and HTML for web pages.
2.1.6 UML Profile

A UML profile[18] is an extension mechanism provided by UML to allow the customization of UML models for a particular domain purpose. A UML profile, instead of allowing the direct edition of the UML metamodel, allows to extend it with new concepts, creating those more specific concepts for the given problem domain, hence the UML profile mechanism also being called a "lightweight approach". UML profiles are defined using the following elements:

- **Stereotypes**: A stereotype is a specific metaclass that extends UML metaclasses, meaning, a new element that extends the elements defined in the UML metamodel. For instance, a stereotype extends metaclasses like Class, Attribute or Operation. A stereotype can also define a custom appearance or representation of the element defined by it;

- **Tagged Values**: A tagged value is a meta-attribute contained by a stereotype. Tagged values are equivalent to Class attributes and offer more expressiveness to stereotypes, since they add more information to them;

- **Constraints**: A constraint is a restriction or rule required in a particular domain that restrains the use and the relationships between stereotypes. The constraints specified by the UML metamodel cannot be weaken by these. Constraints can be defined in OCL.

UML profiles provide a straightforward way to produce DSMLs, since UML is widely known and used and the UML profile mechanism is supported by the majority of UML tools. Therefore, existent MDD technologies based on UML can also be reused to use UML profiles thereby reducing the implementation cost. However, model designer must deal with an infrastructure ready for all UML concepts, since the UML profiles are extensions of UML. Namely, the designer must be aware in order to not mix the predefined UML constructs with the domain-specific constructs, which are defined in UML profiles.

Some examples of UML profiles are MARTE (Modeling and Analysis of Real-Time and Embedded systems), used for specifying real-time and embedded applications, and BPMN (Business Process Modeling Notation), used for specifying business process models. Figure 2.5 provides an example of the definition and use of a simple UML profile.

2.1.7 Model Transformations

Model transformations are key aspects to the MDD approach, because it is by making use of them that the development process can be partially automated. There are two types of model transformations in MDD: (1) Model-to-Model (M2M) and (2) Model-to-Text (M2T)[22].

![Figure 2.5: Example of a UML profile definition (left) and application (right)[7].](image-url)
Model-to-Model Transformations

A M2M transformation consists in a transformation process that receives one or more models as input, producing one or more models as output. The most common case is a one-to-one mapping transformation, which receives one input model and produces one output model. An example of such transformation is the transformation of a class diagram into a relational model for a database. However, there are situations where one-to-many, many-to-one or many-to-many transformations are required

The XIS-Mobile is an example of a MDD framework that uses multiple diagrams as input to generate multiple diagrams as output, meaning it performs many-to-many transformations. Several languages specialized on M2M transformations have been developed over the last decade, but perhaps the most popular and used ones are QVT and ATL.

QVT (Query-View-Transformation) is a standard proposed by the OMG. As its name suggests, QVT comprises three languages based on MOF for developing model transformations: (1) QVT-Relations; (2) QVT-Operational; and (3) QVT-Core. QVT-Relations defines in a declarative manner the correspondences between the source and target metamodels. These correspondences can be both unidirectional and bidirectional. QVT-Relations provides both a graphical and textual concrete syntax. The QVT-Operational language allows the specification of unidirectional model transformations in an imperative manner. QVT-Operation is represented by making use of a textual concrete syntax. Finally, the QVT-Core language is a declarative language designed to be the target of the compiler of the QVT Relations language. Thus, it is not intended to write transformations directly in QVT-Core. Eclipse provides tool support for QVT, namely for the declarative languages (QVT-Relations and QVT-Core) and QVT-Operation.

ATL (ATLAS Transformation Language) is a model transformation language developed by and maintained by OBEO and AtlanMod (formerly ATLAS Group). ATL is a rule-based language which uses OCL, but additionally provides features to support model transformations (e.g. the ability to create model elements). ATL transformations are represented textually and are unidirectional. For instance, if a transformation from model A to model B is required, and vice versa, two ATL transformations have to be specified. ATL transformations only allow reading the source model and writing on the target model. This is done to assure that the result of queries in the source model do not differ based on the execution state of the transformation. Additionally, ATL has been implemented as plug-in for the Eclipse M2M (Model to Model) project.

Model-to-Text Transformations

A M2T transformation consists in a transformation process that receives a model as input, producing text as output. Since MDD intends to generate a running software system from models, the text generated by M2T transformations usually corresponds to the source code. Therefore, M2T transformation can be also known as code generation. M2T transformations can be divided in two approaches: (1) visitor-based approaches and (2) template-based approaches.

In the visitor-based approach, a mechanism based on the Visitor design pattern is used to go over the internal representation of the model and write code to a text stream. A few examples of tools that make use of this approach are Kermeta3 and JAMDA.

In turn, the template-based approaches are more popular and commonly used. They consist in the definition of code templates, meaning, text files that contain great part of the code to be generated and some specific annotations. Those annotations represent the dynamic part of the code template, because in generation time they are replaced by the data of the source model. Using approach, the structure of a template resembles more closely the code to be generated, compared to the visitor-based approaches. Besides, since templates are defined in separate files, it is easier to perform an iterative development. JET (Java Emitter Templates) and Apache Velocity are two examples of simple template engines, while Acceleo and Xpand are two popular examples of template-based code generation frameworks.
Table 2.1: Comparison between four template-based code generators[1].

<table>
<thead>
<tr>
<th>Tool</th>
<th>Template Language</th>
<th>Editor Support</th>
<th>Debugger/Profiler Support</th>
<th>Custom Metamodel</th>
</tr>
</thead>
<tbody>
<tr>
<td>JET</td>
<td>JSP</td>
<td>Yes (Veloedit Eclipse plugin)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Apache Velocity</td>
<td>VTL</td>
<td>Yes (EMFT JET Editor)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Acceleo</td>
<td>MOFM2T</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Xpand</td>
<td>Xpand</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2.1: Comparison between four template-based code generators[1].

Figure 2.6: Anscombe’s Quartet, to the left the four different data set represented as a table, and to the right the corresponding visual representation[8].

provides a brief comparison between these four technologies.

2.2 Basics on Data Visualization

This section presents some basic concepts and insight around Data Visualization while in Section 2.2.1 makes mention of some of the most popular tools to create data visualizations.

Data can be considered anything, from social information to analytics from a website, which can be stored and later processed. The content of the data can possess intrinsic value, but most of the times the, since raw data can be intimidating and deceptive, recognizing that value on this format might be hard to do, and it is impracticable. Data visualization tends to this problem. Data visualization is a general term which is usually applied to the display of data that is produced with the intention of helping people understanding faster and more efficiently, either in terms of significance, dimension or even the subject of it. In text-based data, patterns, trends and correlations might go undetected, and representation that data visually can expose those, making its their recognition easier. Figure 2.6 shows this statement with the famous example of data visualization of Anscombe’s quartet[8]. As it can be seen, with the aid of the visual representation, some relations can be observed instantaneously. The XIS-Analytics language presented on this dissertation intents to capitalize on data visualization for these reasons.

Technology as enabled analysis of text to produce graphical representation of prominent concepts as
Data, at the highest level, can be separated in two main categories: Quantitative and Qualitative. Quantitative data deal with numbers and all things possible to measure objectively (example: height, width and temperature). Qualitative data on the other hand, deal with characteristics and descriptors that are not easily measured, which the interpretation of the values can be done subjectively (example: textures and color). Between this two main categories there are subcategories. For quantitative data there is continuous and discrete (ratio) data, while for qualitative there is binomial, nominal and ordinal data, as it can be seen in Figure 2.7 from the least to the most restrictive.

Depending on the combination of data types that compose the dataset, some types of charts and dashboard should be considered in favor of others, and choosing the right one can improve drastically the quality of the data visualization.

A data visualization is always built with a "story to tell". This story represent the subject of what the data is about, and from what angle it is represented. For example, if the designer of the data visualization intents to utilize a student score chart, a story might be "who as the better grades", for which a simple bar chart suffices, but a line chart, which usually is used to represent time progressions. The data visualizations are built with the intention of either exploration or explanation. In exploration the user does not know what the story is, since it is not well defined to allow the user to make with the data what he wills, while in explanation there is a story to be told, and the goal that the visualization has is
for the user to understand it. Thus, data visualization need to be built with intentional choices, and those choices are made with three main inputs that help guided: (1) the data, (2) the audience/reader, and (3) the designer[27]. (1) The data for it self, depending on how it is structured and types of information that contains should be represented in different ways. (2) The audience/reader since the data is intended to be used by them, so it should have their needs in consideration. (3) And finally the designer, since it is by whom the data is designed, and it is design with his purpose on mind. Having said that, most of the data visualizations are built with their goal already decided by the designer to help the reader. This separation of roles is what XIS-Analytics intents to change. The challenge of XIS-Analytics is to allow to the user the exploration of a dataset, assuming that the structured input was respected. Regardless of the content of the data itself, XIS-Analytics always returns data visualizations that can be useful, thus not requesting from the user a premeditation of how he wants to visualize the data and also not obliging him to have an extensive knowledge on how to explore the dataset. By using XIS-Analytics, the user becomes both the audience/reader and the designer.

2.2.1 Data Visualization Tools

Data visualization tools can be divided in two groups, being they are made focus more on the needs of developers or non-developers, but can be ultimately used by both. This section enumerated some of the most used and known in both groups.

The tools in the form of libraries are mostly used by developers. These libraries are very extensive, and most of them use Javascript, so the visualization can run in all browsers and mobile devices. D3[1] is the most popular and powerful open source JavaScript library for charts and graphs[28]. As such, there are several variations of variants, including amCharts[2], the library of choice for XIS-Analytics. Both of these libraries are used to create interactive and animated data visualization components using large datasets. Companies such as Twitter and Facebook use commercial libraries such as HighCharts[3], these offering more features then the open source D3. For non developers most of the noteworthy tools are comercial. A example on the market that offers the possibility to query data with out knowing what might be in it, while providing dashboards and chart at the end of the pipeline is RJMetrics[4]. This is not a automated service, since it involves constant support. Another good example is Alteryx[5], promoting themselves as “the leading platform for self-service data analytics”, providing both the data analytics capabilities, allowing the user to take decisions on it, while possessing a array of customized charts to choose from to represent it.

Table 2.2 provides a comparison between some of different JavaScript charting frameworks currently to choose from and what kind of chart types the frameworks have available, justifying the choice of using amCharts for XIS-Analytics, which, besides being free, it provides with most of the chart types (having all of the selected ones for XIS-Analytics), giving the possibility the extension of the representation that XIS-Analytics currently offers.

2.3 Related Work

After presenting some of the concepts that are incorporate in XIS-Analytics, before going in depth on how it is built, it is time to present some of the related research work in the different areas that XIS-Analytics worked with. Section 2.3.1 presents an overview of the research made on the subject of data

---

1 https://d3js.org
2 https://www.amcharts.com/
3 http://www.highcharts.com
4 https://rjmetrics.com
5 http://www.alteryx.com
2.3.1 Data Modeling and Data Analytics

The initial take of this MSc was to explore the possibility of the creation of the formerly known XIS-Big Data Framework, which had as a goal to allow developers to build applications that would work with large data sets. After the initial project review, the scope was advised to be redefined, since it was brought upon that the initial goal was too ambitious and maybe unrealistic. It was then decided that this future framework should aim to allow an user to explore and analyze a dataset, making use of data visualization. Since this framework was not tested with a dataset of enough size to be considered Big Data, the name XIS-Big Data was too modest, hence the change to XIS-Analytics. The research was divided in two parts, corresponding to two different questions: (1) how should the data inputted be modeled and (2) how should that data be processed and analyzed. To provide answers to these questions, three main approaches were research on: Operational, Decision Support and Big Data. A bigger focus is provided to the decision support approach since it was the one that XIS-Analytics took more from in while designing it.

Before tackling the first question, a brief introduction to the data modeling process may involve the definition of three data models defined at different abstraction levels, namely Conceptual, Logical and Physical data models[29][30]. Figure 2.8 shows part of the three data models for each of these abstraction levels.

- **Conceptual Data Model:** is used to define, at a very high and platform-independent level of abstraction, the entities or concepts, which represent the data of the problem domain, and their relationships. It leaves further details about the entities (such as their attributes, types or primary keys) for the next steps. This model is typically used to explore domain concepts with the stakeholders and can be omitted or used instead of the logical data model.

- **Logical Data Model:** is a refinement of the previous conceptual model. It details the domain entities and their relationships, but standing also at a platform-independent level. It depicts all the attributes that characterize each entity (possibly also including its unique identifier, the primary key) and all the relationships between the entities (possibly including the keys identifying those relationships, the foreign keys). Despite being independent of any DBMS, this model can easily be mapped on to a physical data model thanks to the details it provides.

Table 2.2: Comparison between data visualization available on different JavaScript charting frameworks.

<table>
<thead>
<tr>
<th>Framework Name</th>
<th>Line</th>
<th>Timeline</th>
<th>Scatter</th>
<th>Area</th>
<th>Pie</th>
<th>Donut</th>
<th>Bullet</th>
<th>Radar</th>
<th>Funnel</th>
<th>Gantt</th>
<th>Network</th>
<th>Grouped</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>amCharts</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Free</td>
</tr>
<tr>
<td>AnyChart</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Proprietary</td>
</tr>
<tr>
<td>FusionCharts</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Commercial</td>
</tr>
<tr>
<td>Highcharts</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Commercial</td>
</tr>
<tr>
<td>JTable</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Free</td>
</tr>
<tr>
<td>OLAPCharts</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Free Basic Edition</td>
</tr>
</tbody>
</table>

modeling and data analytics. This research was made with the focus of making XIS-Analytics function in a Big Data environment, and will be taken in consideration for future work on XIS-Analytics. This research resulted in, as said before, a publication on the subject, as well as the development of a fully functional plugin, both of which will be elaborated upon. This section will also present how this research influenced the architecture and development process of XIS-Analytics. Finally, in Section 2.3.2, some works similar to XIS-Analytics are presented.
• **Physical Data Model**: visually represents the structure of the data as implemented by a given class of DBMS. Therefore, entities are represented as tables, attributes are represented as table columns and have a given data type that can vary according to the chosen DBMS, and the relationships between each table are identified through foreign keys. Unlike the previous models, this model tends to be platform-specific, because it reflects the database schema and, consequently, some platform-specific aspects (e.g. database-specific data types or query language extensions).

Summarizing, the complexity and detail increase from a conceptual to a physical data model. First, it is important to perceive at a higher level of abstraction, the data entities and their relationships using a Conceptual Data Model. Then, the focus is on detailing those entities without worrying about implementation details using a Logical Data Model. Finally, a Physical Data Model allows to represent how data is supported by a given DBMS[29][30].

As said before, three main approaches where studied to research what would be the best approaches for XIS-Analytics to be based on: Operational Databases, Decision Support Databases and Big Data Technologies.

The Operational Databases mostly make use of the Relational Model, since its use overcame the problems of predecessors data models (Hierarchical Model and the Navigation Model)[31]. The usage of the Relational Model caused the emergence of Relational Database Management Systems (RDBMSs), as well as the definition of the Structure Query Language (SQL)[32] as the standard language for defining and manipulating data in RDBMSs. RDBMSs are widely used for maintaining data of daily operations. Considering the data modeling of operational databases there are two main models: the Relational[33] and the Entity-Relationship (ER)[34] models. These systems are designed to handle a high number of interactions that usually perform changes to the operational data. Systems like these are called Online Transaction Processing (OLTP) systems, and these are what RDBMSs are mostly used nowadays. RDBMs have increasingly been optimized to perform well in OLTP systems, namely providing reliable and efficient data processing[30]. The set of operations supported by RDBMSs is derived from the relational algebra and calculus underlying the Relational Model[29] and usually guarantees the traditional ACID (Atomicity, Consistency, Isolation, Durability). SQL is the standard language to perform these operations. SQL can be divided in two parts involving different types of operations: Data Definition Language (SQL-DDL) and Data Manipulation Language (SQL-DML). The SQL-DDL allows the creation of the various database object while SQL-DML allows the manipulation of the those database objects.

Decision Support Databases were the evolution of the relational databases, hereinafter indistinctly referred as "Data Warehouses" (DWs), occurred with the need of storing operational but also historical data, and the need of analyzing that data in complex dashboards and reports. Even though a DW seems to be a relational database, it is different in the sense that DWs are more suitable for supporting query and analysis operations (fast reads) instead of transaction processing (fast reads and writes) operations. DWs contain historical data that come from transactional data, but they also might include other data sources[35]. DWs are mainly used for OLAP (online analytical processing) operations. OLAP is the
approach to provide report data from DW through multi-dimensional queries and it is required to create a multi-dimensional database. Most DWs include a framework which allows the extraction of data from multiple data sources and transform it before loading to the repository. This framework is known as ETL (Extract Transform Load) framework. The data modeling in DW consists in defining fact tables with several dimensions tables, being that most of the used cases fit into either the star or snowflake schema data models category. Figure 2.9 provides with generic example of a star schema model. A star schema has a central fact table linked with dimension tables. Usually, a fact table has a large number of attributes, which can be denormalized, with many foreign keys that are the primary keys to the dimension tables. The dimension tables represent characteristics that describe the fact table. When star schemas become too complex to be queried efficiently they are transformed into multi-dimensional arrays of data called OLAP cubes which represent the aggregation of all the tables. The OLAP cube offer a set of operations to analyze the cube model. Since data is conceptualized as a cube with hierarchical dimensions, its operations have familiar names when manipulating a cube, such as slice, dice, drill and pivot. Figure 2.10 depicts an example of these operations in use. These operation all have as a purpose to query and sort the data among the cube. The most popular language for manipulating OLAP cubes is MDX (Multidimensional Expressions), which is a query language for OLAP databases that supports all the operations mentioned above.

The star schema and the OLAP cube are designed a priori with a specific purpose in mind and cannot accept queries that differentiate much from the ones they were design to respond too. The benefit in this, is that queries are much simpler and faster, and by using a cube it is even quicker to detect patterns, find trends and navigate around the data. This is the reasons why the star schema model and decision of requesting a specification of structured data input was chosen to be included in XIS-Analytics.

Finally, the Big Data Technologies, which is the name given to the technologies that are able to manage the huge amount of data with variable representation, that make the RDBMSs and DWs often impracticable. Thus, there is the need to devise new data models and technologies that can handle this new paradigm of data. NoSQL (Not Only SQL) is one of the most popular approaches to deal with

---

Figure 2.9: Star Schema Model example

---

http://datawarehouse4u.info
this problem. A NoSQL system, consisting in a group of non-relational DBMSs, allows the management and storing of large-scale denormalized datasets, achieving this by compromising consistency in favor of availability and partition-tolerance\[41\]. NoSQL provides BASE guarantees (Basically Available, Soft state and Eventually consistent) instead of the traditional ACID guarantees, in order to greatly improve performance and scalability\[42\]. One of the reasons it was not appropriate to the current version of XIS-Analytics. Big Data analytics can be separated in four categories: (1) Batch-oriented processing; (2) Stream processing; (3) OLTP; and (4) Interactive ad-hoc queries and analysis. The (1) Batch data processing is an efficient way of processing high volumes of data is where a group of transactions is collected over a period of time\[43\]. On the other hand, (2) Stream processing involves a continual input, process and output of data, being that this data must be processed in a small time period\[44\]. (4) OLTP, as said before, mainly used by RDBMSs, are mostly used to deal with transaction processing, being that their goal is to achieve availability, speed, concurrency and recoverability. Finally, (4) interactive ad-hoc queries and analysis consists in a paradigm which allows querying different large-scale data sources and query interfaces with a very low latency, arguing that queries should not need more than a few seconds to execute, no matter the dataset size, so that users are able to react to changes if needed. Thus, XIS-Analytics choose to process its data input in a similar way of batch data processing, by collection JSON files for later analysis. JSON was the chosen format since it is lightweight data-interchange format while also being a simple data format (easy for humans to read and to write). It is a format which is currently widely used, already having multiple REST APIs exporting JSON for personal and consumer use. The chosen format for the JSON file used as an input is shown in Figure 2.11.

From the data modeling perspective, Table 2.3 considers the following features of analysis: (1) the data model; (2) the abstraction level in which the data model resides, according to the abstraction levels (Conceptual, Logical and Physical) of the database design process; (3) the concepts or constructs that compose the data model; (4) the concrete languages used to produce the data models and that apply the previous concepts; (5) the modeling tools that allow specifying diagrams using those languages; and (6) the database tools that support the data model.

On the other hand, in terms of the Data Analytics perspective, Table 2.4 considers six features of analysis: (1) the class of application domains, which characterizes the approach suitability; (2) the common operations used in the approach, which can be reads and/or writes; (3) the operations types most typically used in the approach; (4) the concrete languages used to specify those operations; (5) the abstraction level of these concrete languages (Conceptual, Logical and Physical); and (6) the technology
Figure 2.11: Example of JSON structure used as input for XIS-Analytics.

Table 2.3: Comparison of the approaches from the Data Modeling perspective\[^2\].
Table 2.4: Comparison of the approaches from the Data Analytics perspective\(^2\).

<table>
<thead>
<tr>
<th>Features</th>
<th>Class of Application Domains</th>
<th>Common Operations</th>
<th>Operations</th>
<th>Concrete Languages</th>
<th>Abstraction Level</th>
<th>Technology Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational</td>
<td>OLTP</td>
<td>Read/Write</td>
<td>Select, Insert, Update, Delete, Join, OrderBy, GroupBy</td>
<td>SQL-DML</td>
<td>Logical, Physical</td>
<td>Microsoft SQL Server, Oracle, MySQL, PostgreSQL, IBM DB2</td>
</tr>
<tr>
<td>Decision Support</td>
<td>OLAP</td>
<td>Read</td>
<td>Slice, Dice, Drill down, Drill up, Pivot</td>
<td>SQL-DML, MDX, XMLA</td>
<td>Logical, Physical</td>
<td>Microsoft SQL Server, Oracle, MySQL, PostgreSQL, IBM DB2, Microsoft OLAP provider, Microsoft Analysis Services</td>
</tr>
<tr>
<td><strong>Big Data</strong></td>
<td>Batch-oriented processing</td>
<td>Read/Write</td>
<td>Map-Reduce, Select, Insert, Update, Delete, Load, Import, Export, OrderBy, GroupBy</td>
<td>HiveQL, PigLatin</td>
<td>Logical, Physical</td>
<td>Hadoop, Hive Pig</td>
</tr>
<tr>
<td></td>
<td>Stream Processing</td>
<td>Read/Write</td>
<td>Aggregate, Partition, Merge, Join</td>
<td>SQLstream</td>
<td>Logical, Physical</td>
<td>Storm, S4, Spark</td>
</tr>
<tr>
<td></td>
<td>OLTP</td>
<td>Read/Write</td>
<td>Select, Insert, Update, Delete, Batch, Get, OrderBy, GroupBy</td>
<td>CQL, Java, JavaScript</td>
<td>Logical, Physical</td>
<td>Cassandra, HBase</td>
</tr>
<tr>
<td>Interactive</td>
<td>ad-hoc queries and analysis</td>
<td>Read</td>
<td>Select, Insert, Update, Delete, OrderBy, GroupBy</td>
<td>SQL-DML</td>
<td>Logical, Physical</td>
<td>Drill</td>
</tr>
</tbody>
</table>

support of these languages and operations.

2.3.1.1 JSON Discoverer

This section presents a plugin which is intended to be integrated to the current XIS-Analytics in the brief future. This plugin was implemented by altering the already existing JSON Discoverer tool. The JSON Discoverer is a tool that allows the user to discover the implicit schema on JSON\(^{45}\). One of the goals of JSON Schema Discovery is to assist developers in the composition of application programming interfaces (APIs). Any JSON document includes both metadata (i.e., the schema) and data (i.e., the objects/values conforming to the schema). Given a set of JSON documents, JSON discoverer analyzes the JSON definitions and generates for the user a class diagram showing graphically the implicit JSON schema of those same documents plus an object diagram representing their data. Figure 2.12 presents the overall view of the approach used by JSON Discoverer, JSON Discovery begins by first analyzing the domain behind each involved JSON file and then identifies composition links among them. Then, the discovered information is used to render the graph, presenting the domain information as class diagrams and their relationships. As said previously, at the beginning of this project, the possibility of using JSON Discoverer was explored by creating an adaptation. An adaptation was needed because, since JSON Discoverer only outputted ECore models, and since the Enterprise Architect cannot receive ECore models as input, both technologies were incompatible and the usage of the EA visual editor was a requirement for this dissertation project. This adaptation of the tool is further explained in Section 5.4.
2.3.2 Similar Works

To search for works similar to XIS-Analytics, one should look for ETL (Extract, Transform and Load) systems that are able to provide a similar data processing and data transformations. Thus, one should look for systems that have as purpose the production of data visualizations from given datasets. The purpose of an ETL system is to integrate data from multiple applications. For an example, the REST APIs, similar to the ones address in the case studies for this dissertation (Section 3.3), usually have multiple endpoints from where the data can be Extracted, which after can be Transformed into a specific format or structure for the required querying and analysis purpose, allowing it to be Loaded however is intended to be used[17].

Pentaho’s spoon[46] was the ETL tool used to produce the JSON files that were used as input for XIS-Analytics throughout its development process and testing. Pentaho offers two edition: the enterprise edition and the community edition. The enterprise edition contains extra features not found in the community edition, being that the enterprise edition is not open source, and the community edition is. The spoon used to produce the inputs to test XIS-Analytics belongs to the community edition. However, there are two examples from the community edition that have goals similar to XIS-Analytics: Pentaho Dashboard Designer and the Pentaho Interactive Reporting[46]. Pentaho Dashboard Designer is a commercial plug-in, with a focus on business analytics, which allows its users to create data visualizations in the form of dashboard. This dashboards are collections of other content components displayed together with the goal of providing a centralized view of key performance indicators and other business data movements, letting users monitor them and make decisions. These individual content components come in the form of information graphics, tables, OLAP views or reports, and the creation of the dashboards can be done by drag-and-drop interaction to a layout template of a dashboard, much similar to the Interaction Space view in XIS and XIS-Mobile. The Pentaho Interactive Reporting is similar solution, but allowing for the creation of reports instead of dashboards, and enabling an ad hoc creation, fitting this approach in the category of interactive ad-hoc queries and analysis.

Google Analytics[47] is a web analytics solution offered by Google, and it is currently the most widely used web analytics service on the internet[48], since it is one of the easiest and simpler solutions to learn. It allows for customized data collections, creation of custom reports, importation of data from other sources and can be applied to website, mobile and other digital environment. On top of that, since it is a Google product, it allows as well for cross platform linkage between other Google products. But because it is a free Google product come with three main problems: it is constantly being upgraded and requires a constant learning process, has a somewhat limited option in terms of the types of data visualizations that can be created, and all the user produces can be claimed to be owned by Google. However, there is also the Google Analytics Premium, which is a paid upgraded version of the free solution, which solves the problem of Google owning the created data and data visualizations as well as winding the creation options.
Piwik is a good alternative to Google Analytics in some cases. Piwik is also an open source web analytics application running on MySQL/PHP webserver. It is a solution mostly used to motorize the activity and traffic on a website, providing with analysis regarding visits, goals, downloads, keywords and more. This data is accumulated and then analyzed providing the user with data visualizations so that he can take his own conclusions. It follows a similar process to the XIS-Analytics but focused on website monitoring.

Perhaps the most similar work compared to XIS-Analytics is Caravel. Caravel (by airbnb) is a data exploration platform designed to be visual, intuitive and interactive. Like XIS-Analytics, Caravel’s main goal is to make it easy to slide, dice and visualize data, empowering its users with the capabilities of interpret the results. Providing the user with a large array of visualizations to choose from, and providing the possibility to gather some into interactive dashboards, it can be said that, even if is still a work in progress, it is a very promising one. Originally designed to work on top of the database Druid.io it has already broadened its scope to support other databases by using SQLAlchemy but it is still limited to relational databases.

[8] https://github.com/airbnb/caravel
Chapter 3

XIS-Analytics Approach

This chapter presents an overview of the approach proposed in this dissertation, the XIS-Analytics approach. The XIS-Analytics approach relies in two components: the XIS-Analytics language (a DSL to generate data visualizations of a dataset), and the XIS-Analytics framework (a set of tools that supports the MDD of the process of representing datasets recurrin to data visualizations using the XIS-Analytics).

Thus, Section 3.1 provides with the background from which the XIS-Analytics approach is based. Since XIS-Analytics is based on both XIS-Mobile and XIS profile, a presentation of both is made. XIS-Analytics is linked to XIS and XIS-Mobile because it reuses, extends and materializes the approach and some of the concepts from both.

Section 3.2 presents an overview of the organization of the XIS-Analytics language and the dependencies between its views. Section 3.2 details the design approaches supported by the XIS-Analytics approach. At last, Section 3.3 describes two case studies that were used during this research work, and are referenced throughout this dissertation.

3.1 XIS and XIS-Mobile

This section presents the basic of both the languages that are the background, from which this research work has been based that motivated the development of a UML Profile focused on data analysis. The work presented in this dissertation has as a basis an existing DSL named XIS-Mobile[49][10], which is an extension of XIS(eXtreme modeling of Interactive Systems)[50]. Even though it is based on both this languages concepts, it differs considerably, since for XIS-Analytics some of the concepts were removed and new ones needed to be introduced. Still, the concepts of XIS-Mobile and XIS are presented in this section.

XIS is a UML profile focused on the design of interactive software systems at a PIM level (Platform Independent Level) by following a MDD approach. As shown in Figure 3.1, XIS is organized in three major groups of views: Entities, Use-Cases and User-Interfaces. XIS-Mobile is a DSL that allows the design of mobile applications in a platform-independent way. XIS-Mobile, being a extension of XIS, as shown in Figure 3.2, is organized in four major group views: Entities, Architectural, UseCases and User-Interfaces views. Both XIS and XIS-Mobile define two modeling approaches: the smart approach and the dummy approach[49][10]. In the smart approach, the designer of the models only needs to specify specific set of views: For XIS, the Domain, BussinessEntities, Actors and UseCases views, and for XIS-Mobile, the Domain, BusinessEntities and UseCases views. When using the smart approach, the User-Interfaces views can be automatically generated though M2M transformations (from the Domain and UseCases views) and then extended or refined through direct design. Using the dummy, on the other
hand, the developer needs to define all views manually from scratch. The XIS-Mobile’s Architectural View is a special case, and it is a view that always needs to be specified by hand, but only required if the application interacts with external entities.

The set of Entity Views is composed of the Domain and BusinessEntities view for both DSLs. In the Domain View the relevant classes to the problem domain are represented, together with their attributes and the relationships among them. In turn, the goal of the BusinessEntities View is to define higher-level entities, known as business entities, that intent to aggregate the entities represented in the Domain View or other business entities. Business entities should be manipulated depending on the context of a specific given use case.

The set of Use-Cases View for XIS contains the UseCases View and the Actors View, while for the XIS-Mobile this set is replaced by a single view, the UseCases View. In XIS, the Actors View specifies the entities that can perform actions over the system, while the UseCases View relates the actors defined in the previous view with the operations they can perform over the business entities, when interacting with the system. The UseCases View of XIS-Mobile, both this specifications and relations can be found, being that this view plays an important role during the M2M transformations present in XIS-Mobile.

The XIS-Mobiles’s Architectural View is composed by the representation of the interactions between the mobile application and any external entities. As said before, this view is not always required.

At last, the set of User-Interface views, present in both XIS and XIS-Mobile, composed by the InteractionScape View and the NavigationSpace View. The NavigationSpace View defines the navigation flow between interaction spaces with which the user interacts, while the InteractionSpace View details the elements of the graphical interface contained in each screen and also can specify the access control of the actors to these elements. As state before, these can be generated automatically with the smart approach.
Summarizing, both XIS and XIS-Mobile represent useful solutions to model simple desktop or web interactive applications and mobile applications in a platform-independent way, respectively \[49\][10]. However, the proposed challenge for this dissertation was to explore the possibility to extend on both of this DSLs, reutilizing the concepts present in both, to create a DSL able to query datasets of considerable dimensions, making use of the same or similar technologies. During the execution of this thesis, the scope of the DSL was redefined in terms of complexity and dimension, since it was brought upon that it was very audacious for a master thesis, but never the less, the research work was done, resulting in a publication on the subject, as some concepts from the research were useful and utilized while designing the XIS-Analytics language and framework. This research is presented in the next section, with an emphasis on the concepts used on XIS-Analytics. Thus, this dissertation not only presents the XIS-Analytics language focused on representing data making use of visual representations, but also its supporting framework based on the Sparx Systems Enterprise Architect (EA), a widely used and popular modeling tool.

### 3.2 Overview

XIS-Analytics proposes a DSL, in the form of a UML profile, to generate visual representations of the contents of a given dataset, making use of a MDD approach. XIS-Analytics allows the user to explore and analyze a dataset with the use of a set of data visualization, that among them are able to represent most combinations of datasets, provided with a structure JSON file that represents said dataset, through the use of model specifications. XIS-Analytics proposes a supporting framework that allows the edition of these models and, with the usage of M2T transformations, generates code for each of the different
combinations that might be specified in the models.

To allow these features, XIS-Analytics had been designed having in mind the following four fundamental principles of MDD:

- **Modularization:** The division of systems in modules is important. XIS-Analytics addresses this principle through the use of packages (views) in its organized structure, by using two different views for the data analysis process: one to represent the dataset and data domain while requesting early decisions on how the data is analyzed, and the second one focusing more on the details of the visual representation of the data. Both of this are explained in detail in Chapter 4.

- **Separation of concerns:** XIS-Analytics addresses this principle by using a multi-view organization for each of the concerns of data analysis and representation, while trying to be as minimal as possible, while trying to have the concepts independent of one another. Respecting this principle, XIS-Analytics is built so, in the future, more views that may solve other concerns might be added (e.g. views that support a specific format of file input or even a future interaction space, as suggested in Section 7.2).

- **Use-Case-driven approach:** XIS-Analytics requires for a user to specify a Use Case type of view (the Data Analytics view) for every iteration of the XIS-Analytics process, which is crucial during the M2T transformation stage. For each iteration, depending on the specified Use Case, the generation might vary immensely.

- **Model transformations:** XIS-Analytics proposed the use of M2T transformations to generate source code, allowing the visual representation of a given dataset, from the specifications on the models design by the user.

As shown in Figure 3.3, XIS-Analytics is organized in two packages of views: Entities and Data Analytics views. In turn, the Entities View contains the Domain Entities View. The external package is considered to allow the possibility to extend the XIS-Analytics language in the future. The Domain Entities View does not depend on any other view since it is the starting point, allowing the user to define in what way he wishes to query the dataset. The Data Analytics View depends on the Domain Entities View, since Data Analytics view needs a specified dataset to validate the model itself. Each one of these views is detailed in Chapter 4.
3.3 Case Studies

During this research work, two case studies have been defined to test and evaluate the XIS-Analytics approach. Throughout this dissertation, these case studies are used to exemplify the various aspects and different outputs of XIS-Analytics, contributing for a better understanding and simplicity of explanation.

### Case Study A - IMDB

[IMDB](https://ftp.fu-berlin.de) is an online database for films, television, programs and videos, including cast and all kinds of information. This dataset is composed with information from IMDB Endpoint (ftp.fu-berlin.de) built as JSON file respecting the structure require by XIS-Analytics. The dataset is composed by three main entities: Movie, Actor and Character. It has been built with the intention of using the Movie entity as the fact entity, thus all entities have the movieName acting as a fact key. The Movie entity possess the main attributes regarding the player: "movieName", "year" and "rank". Actor contains the information of the actor that participated in at least a movie, being that information: "actorName" and "actorBirthYear". The Character entity relates the actor with the movie containing information such as: "characterName" and both the "actorName" and "movieName".

### Case Study B - Riot Games League of Legends

Riot Games is a video game publisher and developer, primarily known for the game League of Legends, a real-time strategy video game where players assume the role of a "summoner" and control characters named "champions". This dataset is composed with information from Riot Endpoint (https://developer.riotgames.com) built as a JSON file respecting the structure require by XIS-Analytics. The dataset is composed by three main entities: Summoner, SummonerStatus and TopUsedChampions. It has been built with the intention of using the Summoner entity as the fact entity, thus all entities have the summonerID acting as a fact key. The Summoner entity possess the main attributes regarding the player: "summonerName", "summonerID" and "rank" (ranging from 0-15, being 15 the highest). SummonerStatus contains all the information regarding the games of each summoner as a player, having quantified information such as the number of "wins", "kills", "assists" and "minionKills". The TopUsedChampions contains the most used character of the players with the quantified information of games when they have been used, being the attributes: "championName", "games", "wins" and "losses".

Case study A was the one mostly used to validate the XIS-Analytics approach through out its development and implementation, and it is the result of several case studies refined before reaching this final case, and helped to determine the final format of the inputted JSON file. Case study B was been used in the evaluation user session with people not involved in this research work, Section 6.2 provides more information about this session.
Chapter 4

XIS-Analytics Language

This chapter presents the XIS-Analytics language, a DSL that allows the specification of visual data representation in the form of charts. As stated before, the XIS-Analytics language is divided in two main views: (1) Domain Entities View; and (2) Data Analytics View. These two views corresponds respectively to the two main steps of data exploration: data mapping, and specifying how the user wants to explore it. The Domain Entities view is where the user maps the dataset according to its preferences and the Data Analytics view is where all specifications regarding the data visualizations defined. As such, Section 4.1 describes the Domain Entities View and Section 4.2 describes the Data Analytics View. Additionally, the full and extensive specification of the XIS-Analytics language is provided in Appendix A.

4.1 Domain Entities View

The Domain Entities View is the first view to be defined, thus, this view identifies and defines the concepts relevant to how the data is represented and how it will be read. The way this view needs be defined should have the structured data input in consideration, since if it is inconsistent with it, it will result in problems with the generation of the data visualizations. This view, as stated before, is highly based on the star schema and snowflake schema models that are used in most of the Data Warehouse databases, since XIS-Analytics has as a goal to be as extensible and scalable as possible, to improve the capabilities of the XIS-Analytics Language it self.

As it can be seen, in the Figure 4.1 which represents the metamodel of the Domain Entities view, all the stereotypes for this view can be seen. At the first sight, the similarities with the star schema model (Example in Figure 2.9) can be observed. To represent the multiple entities on the dataset, there are the stereotypes of XisEntityFact and the XisEntityDimension. The XisEntityFact and the XisEntityDimension of XIS-Analytics correspond to the Fact Table and the Dimension Table on the star schema model. It was designed to be that way so that a Domain Entities View is always modeled with a premeditated purpose like one would build a OLAP Cube using star schema models. In the case of XIS-Analytics, the “premeditated purpose” is the dataset which is going to be analyzed, being that, since it is already a structured JSON file, it is assumed that while building the file, the data gathered in it is already there because it fulfills the “purpose”.

Concerning the IMDB dataset from case study A, the domain entity model for it would be something like what can be seen in Figure 4.2. As it can be seen, in this example the user attributed the XisEntityFact stereotype to the Movie entity, meaning it is entity to be considered as the equivalent of a fact table on a star schema, as the XisEntityDimensions are the equivalent of the dimension tables. The keys
Figure 4.1: Metamodel of the Domain Entities View

Figure 4.2: Domain Entities View of the IMDB dataset from case study A
are determined by XIS-Analytics by reading the common elements, and as such, the inputted formatted JSON file should guarantee at least some attributes in common. The attributes of the XisEntityFact should be the ones from which the questions should be ask upon, i.e., for this particular example, "what was the rank of each movie?", or, "how many movies did each actor participated in?". But to specify these questions, the next view needs to be modeled.

### 4.2 Data Analytics View

The Data Analytics View is the second view the user should define. This view needs to respect the previous Domain Entities View, otherwise the it will not be approved by the model validator. Thus, different Data Analytics view can be associated with the same Domain Entity view but not the opposite. However it is not required to the user to make different Data Analytics view to generate more XisDataAnalyticsUseCases (the use case stereotype in XIS-Analytics), since more then one can be represented in the same model, generating as many files as XisDataAnalyticsUseCases. That being said, The Data Analytics View is where the user specifies his preferences on how he intents to query and represent the dataset.

The stereotype from this view that requires the most attention is the XISDataAnalyticsUseCase. With the usage of this stereotype class, the user defines everything related to the specifics of the visual representation, from what data attributes should be considered and in what axis should they be considered to, as well as how they should be ordered on those same axis (if applicable), up to what type of chart does the user wants to represent in each specific use case. A link of the XisEntityFact from the Domain Entity view is required so the XisDataAnalyticsUseCase had have access to the attributes of the dataset.

Still following the Case Study A, the data analytics model for it would be something like what is shown in Figure 4.2. As it can be seen, in this example the user specified two different XISDataAnalyticsUseCase, and each regards the example questions presented in the previous section. The first is meant
Figure 4.4: Data Analytics View of the IMDB dataset from case study A (with the tagged values)

Figure 4.5: Data Visualization for the XisDataAnalyticsUseCase “IMDB Movies Analysis” from Figure 4.2 to answer to the question “what was the rank of each movie?” and the second one to the “how many movies did each actor participated in?”. The tagged values for each XISDataAnalyticsUseCase can be seen on the figure as well. As it can be seen, the user specifies what attribute he wants in what part of the data visualization. The generated data visualizations resulted from generating the code for this case study with this specifications on the model are presented in Figure 4.5 and Figure 4.6. Figure 4.7 shows an example of an alternative data visualization by changing the tagged value “Chart” from “BarChart” to “RadarChart” in the XisDataAnalyticsUseCase “IMDB Movies Analysis”, which while not being the best representation for the specification, the chart is still generated to provide some sort of feedback to the user.
Figure 4.6: Data Visualization for the XisDataAnalyticsUseCase “IMDB Movies Analysis 2” from Figure 4.2

Figure 4.7: Alternative Data Visualization for the XisDataAnalyticsUseCase “IMDB Movies Analysis” from Figure 4.2
Chapter 5

XIS-Analytics Framework

This chapter describes the XIS-Analytics framework, meaning, the integrated environment that supports the Model-Driven Development (MDD) of the data analysis process and visual representation of data using the XIS-Analytics language. First, Section 5.1 provides an overview of the XIS-Analytics framework, namely, describing its architecture, used technologies and proposed development process. Section 5.2 explains how EA has been customized and tailored in order to allow design model using the XIS-Analytics language. Section 5.3 focuses on explaining how the Model-to-Text (M2T) transformation stage had been implemented in the XIS-Analytics framework. At last, Section 5.4 presents the plug-in adapted from the JSON discoverer, which was made to become an extension to the XIS-Analytics framework.

5.1 Overview

The XIS-Analytics language is supported by a MDD-based framework that makes this proposed DSL more relevant and allows its use. As shown in Figure 5.1, the suggested development process of analyzing a data set and generating the charts that represent it, using the XIS-Analytics, which consists in three steps: (1) the definition of the chosen entities from the structured JSON, as well as define the charts, both using the Visual Editor; (2) their validation with the Model Validator; and finally (3) the generation of the application’s source code through the Code Generator. If, after step (3) wishes to continue to explore and analyze the dataset, the process should be repeated from step (1) by redesigning the models for a new iteration. However if the intent is only for small changes to the final generated code, the user can change it since it is generated to be readable and comprehensible.

![Figure 5.1: Development process with the XIS-Analytics Framework](image-url)
The XIS-Analytics framework is implemented using the Model Driven Generation (MDG) Technologies provided by Sparx Systems Enterprise Architect (EA) together with the Eclipse Modeling Framework (EMF). The use of these technologies leverages the environment they provide, as well as some compatible plug-ins. As the Figure 5.1 suggests, the development process by using the XIS-Analytics framework is composed by three modules: (1) Visual Editor; (2) Model Validator; and (3) Code Generator.

First, the Visual Editor is implemented on top of EA through the use of an MDG Technology plug-in. This implementation was specifically created for the XIS-Mobile language, and modified to fit to XIS-Analytics language. The choice of using EA’s MDG Technologies was so that XIS-Analytics language can be defined as a UML profile while being fully compliant with the OMG specification for UML2, as well as allowing the creation toolboxes, diagram types, model patterns and model templates customized to the XIS-Analytics language. While developing the language, at the beginning, Papyrus was used in the preliminary versions of XIS-Analytics. Papyrus is a modeling environment provided by EMF, and it was only used in the preliminary versions since while in the development of XIS-Mobile, and even while using it in the development of XIS-Analytics, it has been proven to be poorly suited to model a language of the complexity of XIS-Analytics, due to bugs that hinder the modeling process as well as lack of proper support from some modeling tools.

Second, the Model Validator is implemented as a plug-in using EA’s Model Validation API. It plays a crucial role avoiding errors by the designer, improving the quality of the models and, consequently, enhancing the quality of the generated code and visual representation. OCL[18] is the standard language commonly used for UML model validation, but unfortunately, due to many limitations with stereotypes validation and ongoing developments of the OCL plug-in for EMF, it was ultimately decided that was not the best option as of right now. That said, even though it is not a standard like OCL, this solution allows the definition of constraints and rules, custom error messages associated with those constraints, the assignment of severity levels (error or warning) to them, and the immediate automatic navigation to the element that did broke the rule and cause the message.

Third, the Code Generator, which is based on Acceleo, a template-based code generator framework available as an Eclipse plug-in. Acceleo implements the MOF Model to Text Language (MTL) standard (OMG, 2008) and allows defining code templates for any kind of model compatible with the EMF. The code templates are composed of regular text (static part of the template) and several annotations (dynamic part of the template) that are replaced by values of the model during generation time. In XIS-Analytics, the values on the model correspond to values on a dataset, hence the current need for a structured file. For now, the XIS-Analytics framework support the generation of JavaScript and HTML5, and utilizes the amCharts library to add the interactive charts to the output. An important point is that the code is ready to be edited if the user wants or sees fit to do so.

5.2 Visual Editor

As mentioned before, the Visual Editor of the XIS-Analytics Framework has been developed leveraging the features provided by EA Model Driven Generation (MDG) Technologies. This MDG technology was specifically created for XIS-Analytics, based on the one created for XIS-Mobile, and it was created in order to tailor the EA environment to the XIS-Analytics language. Additionally to the XIS-Analytics profile definition, this MDG technology implementation contains the defined custom diagrams, custom toolboxes, model patterns, a project template and custom quicklinks. This section provides a description of each of these components below.

Regarding the definition of the XIS-Analytics diagrams, EA allows the definition of custom diagram
and their inclusion in a MDG technology. This custom diagram profile provides specific meta-
classes for each type of diagram. It also allows associating custom toolboxes to the those same custom
diagrams, making it a perfect fit for a DSL such as XIS-Analytics. The XIS-Analytics Profile diagrams
are represented in Figure 5.2.

The custom XIS-Analytics diagram toolboxes have also been implemented through the creation of
profile diagrams. Since XIS-Analytics currently has three profile diagrams, one for each view of XIS-
Analytics, and they have been inclined in the implemented MDG technology. EA provided a custom
toolbox profile[2] which contains the metaclass ToolboxPage, that allows defining the elements and as-
sociations of a custom diagram toolbox. This custom toolbox profile also allows to associate icons to
specific toolbox items, through the use of the metaclass ToolboxImageItem, providing a better toolbox
for the user to utilize. Figure 5.3 shows as an example, the profile diagram for the toolbox of the Domain
Entities View of XIS-Analytics.

Even though XIS-Analytics does not make full use of it, EA also allow the specification of model

---

1 http://goo.gl/pVAwTx (Accessed on April 2016)
2 http://goo.gl/Xsntgw (Accessed on April 2016)
patterns and their association to toolboxes, i.e., toolboxes can have model patterns as items. Basically a pattern consists in a fragment of model that is usually used. As said before, in the case of XIS-Analytics the definition of patterns such as these was not done, since at its current state it was not needed, however it might be used in the future, as it is suggested in Section 7.2 Future Work. By using a Design Pattern, it enables you to rapidly create template solutions for code structures that perform the same type of task in other situations, and to use items defined in the Pattern with the UML model, which might be useful to the user. Models of project templates can also be created using EA MDG Technologies. Upon added to the MDG technology, these templates can be chosen from the list of standard model templates that is already provided by EA. For XIS-Analytics, it was only created a model template that creates the package structure with both views and respective diagrams of a XIS-Analytics project. Figure 5.3 demonstrate this, and the project structure is highlighted in green.

At last, there were also defined custom Quick Linker menus for the XIS-Analytics elements. This is specified by adding to the UML profile diagram a QuickLink Document Artifact element that contains the Quick Linker connections for each element and diagram in a CSV (Comma-separated values) format. This was also created in XIS-Analytics with the goal of reducing the errors, since it restricts the type of associations and elements that be created from a given element.

5.3 Model-to-Text Transformations

The XIS-Analytics’s M2T transformations stage is fully implemented using Acceleo, a plugin provided by Eclipse. Acceleo is an open source framework that allows the creation of template-based code generators. Acceleo was created by Obeo, a french company specialized in providing software solutions to create and transform complex systems like industrial systems (e.g. avionics, space or defense), or IT applications. Acceleo was released in 2006 as a plugin Eclipse, and in 2009 included as in Eclipse’s Model to Text (M2T) project. Acceleo implements the MOF Model to Text Language (MTL) standard.

Table 5.1: Comparison between four template-based code generators[1].

Acceleo allows defining code templates (also known as modules and MTL files) for any kind of EMF-based model, including UML models. As mentioned before, the code generation mechanism is based on templates, which is a special file composed of regular text (static part of the template) and several annotations (dynamic part of the template). These annotations are substituted by values of model elements during generation time.

Acceleo templates are composed of two major types of constructs: templates and queries. Templates are sets of Acceleo statements used to generate any kind of text, and can always be complemented further. They are delimited by the [template ][/template] tags. In turn, queries do not generate text. They are used to extract information from the model. They always return values and are specified with [query ][/query] tags that use OCL inside. Additionally, Acceleo allows the use of Java service wrappers when it is not possible to define the intended template using templates or queries constructs. A Java service wrapper is a Java file that allows navigating the model. The Acceleo non-standard library provides a service invoker which allows invoking the Java service as traditional query. Table 5.1 shows an excerpt of an Acceleo template for generating Java classes. In this case, it was assumed that the input UML contains one class named Movie with four attributes (title, description, rank and date).

Furthermore, Acceleo provides a template editor which allows editing templates in an easier and assisted way, a profiler that allows analyzing the quality of the generation, a debugger to test and place breakpoints and to move between instructions of templates during generation time.

The code templates defined for the XIS-Analytics framework have been developed with Acceleo’s best practices in mind, particularly in terms of naming conventions, project and module organization and Java service use. Figure 5.5 shows the organization of the Acceleo project of the XIS-Analytics's
Figure 5.5: Organization of the Acceleo project

code generator. The naming convention follows the Acceleo recommendation: ![project name]![kind of input]![input metamodel name]![gen]![output language name]. The differences between the packages are described below:

- **common**: This package contains the utility modules, which are the modules with the queries commonly used by the templates. For example, this package possesses a file with several queries that check the XIS-Analytics stereotype of a certain UML element.

- **files**: This package contains all the modules that will generate files, i.e., like the ones exemplified in Table 5.1.

- **main**: This package contains the modules with the main templates and their matching launcher class, i.e., the entry points of the generator;

- **services**: This package contains all the modules that make use of Java service wrappers, as well as the corresponding Java files.

For now, the XIS-Analytics framework supports the exploration of a dataset outputting visual representations of that data with the specifications the user might prefer. The XIS-Analytics framework generates a html file for each applied case. The code of the file follows good practices, and it is ready to edited if the user has the knowhow and the intention. Additionally, it is important to emphasize that whenever the user desires to add different heuristics to the analyze of the dataset, he needs to redefine the corresponding code templates.

Acceleo was the choice, since it had been already used for XIS-Mobile, and once again proved to be a wise choice due to its complete and updated support of UML profile-based models, like the ones produced while using XIS-Analytics. The features mentioned before (editor, profiler, debugger, templates, java services), as well as Acceleo’s extensive and good documentation (guides, videos and tutorials) greatly aided the development process. A very useful resource was the Eclipse dedicated
5.4 JSON Discoverer Plugin

As stated previously in Section 2.3.1.1, a plugin to make use of JSON Discoverer capabilities of discovering the implicit schema of a JSON file was developed. The architecture of this plug-in can be seen in Figure 5.6. It starts by using JSON Discoverer on the desired JSON file, returning the respective Ecore model. Then, making use of the ATL language (briefly described in Section 2.1.7), the UML is generated making use of the XIS-Mobile profile stereotypes of the domain view, since the XIS-Analytics profile was not built yet (and still does not possess the standard domain view stereotypes on its current version). This generated UML is already compatible with Eclipse, and it parsed to generate the EA XMI, i.e., the model is now compatible with Enterprise Architect, and it can be imported to the visual editor. The goal for this EA XMI was to be the input for the XIS-Analytics, as it can be seen in Figure 5.7.

With this EA XMI imported to the visual editor, the user can define the specifications together with the XIS-Analytics other views of, built by the user, to generate the code of the data visualizations, making use Acceleo. Since XIS-Analytics was not designed with a domain view yet, the created plug-in was not yet integrated (further elaborated in Section 7.2). It currently only used as a tool to verify the structure of the JSON files at a development stage.

Chapter 6

Evaluation

This chapter presents and discusses the twofold evaluation performed to XIS-Analytics. First, in Section 6.1, the XIS-Analytics M2T capabilities are evaluated through the analysis of the limitations of the code generated for case studies A and B (see Section 3.3) for all of the types of data visualizations currently supported by XIS-Analytics (BarChart, LineChart, RadarChart, PieChart and BubbleChart). Then, Section 6.2 presents the results of an assessment session of XIS-Analytics by third parties. This assessment focused on three aspects of XIS-Analytics: (1) the Language, (2) the Framework and (3) the generated Data Visualizations. Section 6.3 summarizes the chapter describing the main conclusions obtained from this twofold evaluation.

6.1 Case Study Implementation

This section presents and analyzes the results obtained from implementing the case study A, the IMDB dataset (presented in Section 3.3), using XIS-Analytics. The focus of this implementation was to find the limitations of the code generated from the M2T approach as well refine the case studies themselves for the user session evaluation. The limitations that were looked for were incompatibility within some of the charts and specific combinations of data types (presented in Figure 2.7), since these could cripple the data processing capabilities for XIS-Analytics. These data types were divided between String (Nominal, Ordered, Binomial) and Int/Float (Ordinal, Quantitative, Ratio and Continuous) since XIS-Analytics determines in what category they fall after receiving either one of these.

To find the limitations for each of the data representations, for each case study, a Domain Entity View was modeled by mapping the concepts of the dataset, while a Data Analytics View as modeled as it can be seen in Figure 6.1. As it can be seen in the Figure 6.1 making use of the capability of XIS-Analytics to generate multiple files for a simple Data Analytics View, five case studies where created, one for each supported type of data visualization.

Table 6.1 show the results from testing the different combinations. The Int/Float type is called Value on the table for simplification. The Z field is only applicable to the Bubble chart, since it is the only chart capable of representing three dimensions through his attributes. Taking the nature of XIS-Analytics, and its concern of representing something even if the user does not make a good choice of chart type, a radar chart is able to represent a combination of String x String, even if while interpretation does not provide most suitable option.

Taking in consideration the constraints found in the generation of each data visualization, it can be seen that there is the need to add more variety of visualizations type to the XIS-Analytics language, especially visualizations that would allow representation of Strings combinations. However, it's worth
Figure 6.1: Answers to the questions for the XIS-Analytics Language aspect.

Table 6.1: Different data types combinations that the charts of XIS-Analytics can process.
pointing out that the set of visualizations that XIS-Analytics currently supports, are able to represent most of the possible combinations.

6.2 User Session Assessment

To better evaluate XIS-Analytics, specially taking in consideration that it generates data visualizations and the interpretation of these can be subjective at times, it was important to receive feedback from people no directly involved in this research work. By doing so, these users could detect potential bugs and limitations. So it was decided to conduct a pilot user session. This session involved a group of 12 participants in total with ages ranging from 21 to 28 and with at least a Bachelor of Science degree. From the 12 participants 9 had previous knowledge and experience with UML, while 3 did not (being from other professional areas). 4 participants had professional experience in Information Technology.

The user session was conducted under the following conditions:

- Tests are conducted in the laboratory (controlled environment);
- The task must be performed without previous use and learning (for the first time);
- The user must have a computer running Windows and previously installed Java Runtime Environment (recommended version 7) and Sparx Enterprise Architect (version 7.5, 10 or above);
- The user must have python installed (recommended 2.7 or above);
- Direct Observation, i.e., while users perform the assigned task, their behavior and performance can be logged;
- Users can think out loud and share ideas if they want;
- The evaluator does not interact with the users until the tests are finished (except in case of blocking errors);
- The session will last 50 minutes (at most);
- The user must fill a survey in the end (regarding its experience with XIS-Analytics);

Based on these conditions, the participants received a 10 minutes presentation of XIS-Analytics fundamental concepts, regarding the language and its framework, followed by a 5 minutes briefing for the participants who had never used EA visual editor and had never worked with UML. This briefing was provided in the form of a demonstration of the development with the dataset of case study A (IMDB) using Xis-Analytics, from model design until code generation. Thereafter, a script was provided to the participants (see Appendix B) describing the case study B (Riot Games League of Legends, described in Section 3.3), and were asked to create the corresponding models in XIS-Analytics, validate them and launch the code generation. In the end, participants were asked to fill in a survey to rate the XIS-Analytics language, its supporting framework, and the generated data visualizations. The analysis of the results gathered from these surveys is described below.

Survey Analysis

The survey used in the user session focused on analyzing the quality of three aspects about XIS-Analytics: (1) the Language, (2) the Framework and (3) the Data Visualizations. The answers were classified in a Likert scale [53] (1 to 5) being the classifications: 1 (Very Low/Very Bad), 2 (Low/Bad), 3 (Medium/No Opinion), 4 (High/Good) and 5 (Very High/Very Good).

The XIS-Analytics Language aspect included five questions:
Figure 6.2: Answers to the questions for the XIS-Analytics Language aspect.

- **QL.1.** How suitable is the size (number of stereotypes) of the Language?
- **QL.2.** How easy is to use the notation (defined as a UML Profile)?
- **QL.3.** How easy is to learn the language with the star schema concepts?
- **QL.4.** How easy is to learn the language with the data visualization concepts?
- **QL.5.** How suitable is the language to be applied to a data analytics problem?

Figure 6.2 presents the answers regarding the Language aspect, broken down by question. In general, we can observe that all questions had a positive score (average greater than 2.5) which implies some sort of success. Taking notice of question 1, almost all the participants considered the size of the language suitable (giving a score of 4 and 5), however two participants considered it inadequate, either implying it is too large or perhaps too small. It is the lowest scoring question because of it. Concerning all the other questions, favorable scores were obtained, indicating that XIS-Analytics language notation is easy to use, the domain entities view (star schema) concepts were easy to learn, and it is considered suitable to be applied to data analytics problems. Regarding both question 3 and 5, the participant or participants who answer to those questions with a 3 might be one of the participants who are not in the IT area, and perhaps considered the concepts harder to learn. Still, on question 3, I do not exclude the possibility of refinement to the domain entities view and regarding question 5, there are still cases for which XIS-Analytics is not suited yet. Summarizing, we assume that the XIS-Analytics language contains several concepts that initially can be a bit difficult to learn and understand all of them, especially if the user is not in a data exploration mindset. Despite that, we believe from these results that its size, notation and concepts are adequate for data exploration and analysis.

The XIS-Analytics Framework aspect included six questions:

- **QL.1.** How do you rate the usability of the XIS-Analytics plugin in EA?
- **QL.2.** How do you rate the usability of the Visual Editor?
- **QL.3.** How do you rate the usefulness of the Model Validator?
- **QL.4.** How do you rate the simplicity of the Model-to-Text transformation process?
- **QL.5.** How do you rate the productivity of XIS-Analytics to deduce the value of a dataset?
- **QL.6.** How likely would you use a tool such as XIS-Analytics for a preliminary dataset evaluation?

Figure 6.3 presents the answers regarding the Language aspect, broken down by question. Similarly to the Language aspect, we can observe that all answers had a positive score. Question 4 particularly
obtain a very high score, and comparing this score to the obtained in question 2 and 3, we can conclude that the participants found the process of triggering code generation simpler than using the model validation and the visual editor. The answers in question 2 and 3 might be related to the inexperience of some of the participants with EA and even UML, however, regarding the model validation, during the session there was one participant who, despite his models being correct, the model validator detected a false error, which was troubleshooting to be in the access of the XisAnalyticsUseCase stereotype to the domain entity view through the XisDA-DEAssociation, and it was corrected right after. Regarding question 5 and 6, since the dataset used might not have been of the interest of some of the participants, it can have influenced their opinion, however I do not exclude the possibility of some work might need to be done to the data analytics view, allowing an user to be even more specific on how he wants to generate the data visualizations.

Finally, the XIS-Analytics Generated Data Visualization aspect included two questions:

- **QL.1.** How would you say the generated data visualization look?
- **QL.2.** Using XIS-Analytics multiple charts, how do you think are your odds of extracting conclusions out of the dataset?

Figure 6.4 presents the answers concerning the XIS-Analytics Generated Data Visualization aspect, broken down by questions. We can observe a good reaction to the design of the data visualizations as well as a good reaction to the information displayed, since the score obtained in both questions is highly positive. That said, the participants considered the generated data visualizations better looking then useful for data interpretations, however, regarding this conclusion, the dataset used for the session might be in fault, since some of the participants might not care about the data contained in it.

Summarizing, as it can be seen in Table 6.2, the results were encouraging of the XIS-Analytics.
Table 6.2: Survey's average score (scale of 1-5) for each XIS-Analytics Framework aspect.

<table>
<thead>
<tr>
<th></th>
<th>Language</th>
<th>Framework</th>
<th>Generated Data Visualization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>4.382</td>
<td>4.252</td>
<td>4.375</td>
</tr>
</tbody>
</table>

approach with overall positive scores in all three analyzed aspects. Nevertheless, it was observed that the XIs-Analytics Framework namely in the visual editor and model validator had the lowest score and possibly need to be refined in order to improve their simplicity and performance respectively. It was also observed that the learnability of the XIS-Analytics language concepts, namely in the domain entities view, should be improved.

On a side note, it can be stated that the number of participants of the session is relatively small. We believe that number is sufficient to take meaningful conclusions, because usability experts have noted that a group of 5 testers is enough to uncover over 80 percent of the usability problems\cite{54}. Also, since the survey focuses on the usability of the language, framework and approaches, we believe 12 participants is a reasonable number for an exploratory assessment, at least in order to identify problems on the usability of these aspect. Since the group of participants was composed by IT professionals and other different area professionals, the simplicity of learning the language challenged and tested as well. The complete results of the surveys are provided in Appendix C.

### 6.3 Summary

This chapter presented the results of the twofold evaluation performed to XIS-Analytics. First, the limits of each data visualization were tested making use of the case study A, the IMDB dataset (presented in Section 3.3). This evaluation was done throughout the implementation of XIS-Analytics, but the one documented in Section 6.1 was the final one before the user session. The results obtained were quite satisfactory, concluding that there is a large combination of data types covered by the current XIS-Analytics data visualizations. However, since there were identified some combinations which XIS-Analytics can not yet cover, namely some tri-dimensional combinations, there is the need to extend the code templates further to solve this problem.

The second part of the evaluation focused on the assessment of XIS-Analytics by third parties not involved in this research work. This assessment was done by conducting a user session where the participants had to generate the data visualizations using the XIS-Analytics framework, from the model until code. In the end of this session, the participants filled in a survey focused on three aspect of XIS-Analytics: (1) the Language, (2) the Framework and (3) the data visualizations. The results obtained from the survey were mostly positive and encouraging with positive scores on all three analyzed aspects. It was observed however that some participants found that the model validator, and possibly the concepts of the domain entity view, might need some revision and refinement, since these were the two points where it was observed from the participants some learnability issues.

These evaluation results support the thesis of this dissertation, which advocates that a MDD-based approach can be used to analyze and explore a dataset, mitigating the complexity of the task. Moreover, these results showed preliminary evidences that demonstrate XIS-Analytics’ usefulness and feasibility.
Chapter 7

Conclusion and Future Work

This chapter provides some conclusions regarding the work of this dissertation as well as some future work ideas and suggestions. In Section 7.1 the conclusion is presented, while in Section 7.2 a description of some suggestions for future research directions is provided.

7.1 Conclusion

We live in a data-driven world, where every business and individual is feeling the unprecedented impact of rapid demographic changes, economic shifts, increasing resource scarcity, urbanization, and technology breakthroughs. In this world, data is generated at unprecedented rates. The importance of data processing and analysis was never as important as it is right now. Big companies (e.g., Google, Twitter and Facebook), who possess the resources and knowledge to process and analyze the dataset they possess get great value in return, helping them make better decisions and create new products and services. Not all data has value, but value can be found in all kinds of dataset. However, data analysis usually represents big investments, since the knowhow and the adequate tools to do so are not available for every one, and risk, since some times the data analysis might return no value at all. Thus, given these problems, the ideal scenario would be to have a tool, that could be utilized by most people, which would help determine if a dataset contains value or not.

Fortunately, some work has been conducted over the last years that tackle this problem presented previously. Many tools and frameworks have been developed for open-source or commercial use that allow a user to, as long as they supply the data, to process and analyze it in some fashion, e.g., Google Analytics and Piwik. Most of these recur to data visualizations, which can be used to allow any kind of user to interpret data analysis. Still, most of these solutions don’t allow the user to be simultaneously both the designer and the reader, like the one presented in this dissertation. The presented solution allows this by using a Model-Driven Development (MDD) approach. MDD approaches seek to mitigate these problems by considering models as the center of the development process. Other deliverables, such as source code or documentation, can be generated automatically from those models through model transformations.

In this dissertation, we have proposed XIS-Analytics as a MDD approach to perform data analysis while mitigating the complexity of the task. The XIS-Analytics language is a DSL (defined as a UML profile) focused on data analysis which, given a dataset, represents it with data visualizations from where the user can draw its own conclusions and repeat the process to further continue the exploration. The XIS-Analytics language has a two-view organization that adheres to the "separation of concerns" principle. This fact represents an advantage since, if the user wants to change something specific in the
models, it only needs to iterate the specific one. The two vies of XIS-Analytics are Domain Entities View and Data Analytics View, which are focused in two steps of data analytics: the mapping of the dataset and the specification of the analysis, respectively.

Along with the XIS-Analytics language, a supporting framework is also proposed. This supporting framework is based on Sparx Systems Enterprise Architect MDG Technologies and EMF, which intends to generate the source code for the data visualizations from the model specifications, through the use of model-to-text transformations. Composed by four major modules, this framework suggests developing data visualizations in three steps: defining the required views using the Visual Editor, validating them using the Model Validator, and finally generating the visualizations source code through the Code Generator. This way, the user takes advantage of the MDD benefits, drawing conclusions from the data visualizations whenever possible and easily redoing the models and generating again if not pleased with the previous results.

XIS-Analytics has been developed over the last year following the Action Research Methodology and has been evaluated in order to assess its usefulness and adequacy for the purpose of data analysis. This evaluation process has been done in a gradual and interactive way, through all iteration of the action research, the constant testing with different dataset case studies and by conducting a user session. The implementation of case studies allowed to assess the limitations of common data visualizations. The user session focused on the assessment of XIS-Analytics by third parties, i.e., users not directly involved in the development. During the user session the participants had to analyzed and explore a case study dataset and at the end, fill a survey. The survey focused on the three main aspect of XIS-Analytics: the Language, the Framework and the Generated Data Visualizations. The collected results from the survey were positive and showed preliminary evidence that demonstrate XIS-Analytics’s usefulness, feasibility and support the thesis behind this dissertation. Additionally, it is important to note that all the goals established in Section 1.3 have been accomplished during the course of this research work.

### 7.2 Future Work

This section presents some other issues and possible tasks that can be followed in a future research related to this work. During the course of this work several ideas have emerged, but either due to time restrictions, for research strategy or due to their complexity, they were considered too ambitions and so were not implemented. It is important to emphasize that none of these research directions undermine the achievement of the goals of this work. Thus, some of these open issues are summarized and explained subsequently:

**Extension:** XIS-Analytics could make use of a Domain View, and the attempts to provide one are on the development of adaptation of the JSON Discoverer plugin, which would provide a generated domain view automatically from reading the dataset. After, an ETL process view which would represent the transformation from the Domain View to the Domain Entities View would need to be implemented. It would also be interesting to implement the InteractionSpace/NavigationSpace views, seen on XIS and XIS-Mobile, which would allow the user to customize and depict the UI layout of produced data visualizations, and specify aggregations to create dashboards. Finally, the inclusion of Model-to-Model transformation to accelerate the design process, by providing suggestions to the user, or include the heuristics on the model validator, would also be an interesting extension.

**New types of data visualizations:** The code templates for the five different data visualizations that XIS-Analytics offer are complete. However, given the way XIS-Analytics code generator is implemented, the addition of more types or variations on the already existing ones can always be added.
Support other data formats: Currently XIS-Analytics can only read from a structured JSON file, but even if the limitation of the structure is kept, XIS-Analytics can be adapted to read from other similar formats, e.g., CSV, XML or any other standard formats used in this area.

Generate code for other platforms: Besides generating the HTML5, CSS and JavaScript code for the data visualization, XIS-Analytics could provide the part of the equivalent query in data analytics languages such as R, SQL and Python.

Support other language workbenches: For now, the XIS-Analytics language and framework can only be used in EA, but allowing its use for other commonly used workbenches or editors would be beneficial to increase its acceptance and popularity.

Support incremental model and code generations: XIS-Analytics supports the generation of code, but does not support incremental generations. This means if the user changes a piece of code or models and then needs to perform a new generation, the changes he made will be lost after the generation. Even though the generated data visualizations provide with interaction, it does not suffice for most cases.

Integration with the other XIS languages: Given the fact that XIS-Analytics language does not contain any of the views specified in the other XIS languages, namely XIS[50] and XIS-Mobile[49], and most of its stereotypes are unique, it can be included and integrated with some functionalities of them, namely, the other most recent addition to the XIS family, the XIS-Web, which makes use of Web SQL, which can export JSON.
Bibliography


XIS-Analytics Profile Specification

XIS-Analytics UML Profile comprises two views: Domain Entities View and Data Analytics View. Each one of them is detailed in the next sections.
1. Domain Entities View

<table>
<thead>
<tr>
<th>Stereotype</th>
<th>Metaclass</th>
</tr>
</thead>
<tbody>
<tr>
<td>XisFact</td>
<td>Class</td>
</tr>
<tr>
<td>XisDimension</td>
<td>Class</td>
</tr>
<tr>
<td>XisAttribute</td>
<td>Property</td>
</tr>
<tr>
<td>XisEnumeration</td>
<td>Enumeration</td>
</tr>
<tr>
<td>XisEnumerationValue</td>
<td>EnumerationLiteral</td>
</tr>
<tr>
<td>XisFactAssociation</td>
<td>Association</td>
</tr>
<tr>
<td>XisDimensionAssociation</td>
<td>Association</td>
</tr>
</tbody>
</table>

**XisFact**

Represents the fact dimension for the this specific view of the domain

<table>
<thead>
<tr>
<th>Taged Value</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Specifies the name of the domain entity</td>
</tr>
</tbody>
</table>

**XisDimension**

Represents the dimensions associated to the fact dimension for this specific view of the domain

<table>
<thead>
<tr>
<th>Taged Value</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Specifies the name of the domain entity</td>
</tr>
</tbody>
</table>
### XisAttribute
Represents an attribute or property of a fact or a dimension

<table>
<thead>
<tr>
<th>Tagged Value</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Specifies the identifier of the enumeration</td>
</tr>
<tr>
<td>Value</td>
<td>String</td>
<td>Specifies if the entity should be persisted or not</td>
</tr>
<tr>
<td>type</td>
<td>PrimitiveType</td>
<td>Specifies the type of the domain entity</td>
</tr>
<tr>
<td>nullable</td>
<td>Boolean</td>
<td>Specifies if the attribute can be null</td>
</tr>
</tbody>
</table>

### XisEnumerationValue
Represents an enumeration that can be used as an attribute of a fact or dimension

<table>
<thead>
<tr>
<th>Tagged Value</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Specifies the identifier of the enumeration</td>
</tr>
</tbody>
</table>

### XisEntityAssociation
Represents a enumeration that can be used as an attribute of a fact or dimension

<table>
<thead>
<tr>
<th>Tagged Value</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>roleName</td>
<td>String</td>
<td>Specifies the identifier of the association</td>
</tr>
</tbody>
</table>

### XisFactAssociation
Represents an association between a fact and its dimensions (a parent and its child)

### XisDimensionAssociation
Represents an association between a dimension and its own dimensions (a parent and its child)
2. Data Analytics View

### Stereotype Metaclass

<table>
<thead>
<tr>
<th>Stereotype</th>
<th>Metaclass</th>
</tr>
</thead>
<tbody>
<tr>
<td>XisActor</td>
<td>Actor</td>
</tr>
<tr>
<td>XisActor-DAAssociation</td>
<td>Association</td>
</tr>
<tr>
<td>XisEntityDataAnalyticsUseCase</td>
<td>XisUseCase</td>
</tr>
<tr>
<td>XisEntityDA-DEAssociation</td>
<td>Association</td>
</tr>
</tbody>
</table>

**XisActor**

Represents an Actor

<table>
<thead>
<tr>
<th>Taged Value</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Specifies the identifier of the actor</td>
</tr>
</tbody>
</table>

**XisActor-DAAssociation**

Associates an actor (XisActor) to the use case (XisDataAnalyticsUseCase)

**XisDataAnalyticsUseCase**

Represents a data analytic use case that performs the aggregation of all the attributes over a business entity to be able to produce the result

<table>
<thead>
<tr>
<th>Taged Value</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XIS-Analytics Profile Specification</td>
<td>Afonso Silva</td>
<td>4</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>Specifies the identifier of the result produced</td>
</tr>
<tr>
<td>Chart</td>
<td>ChartType</td>
<td>Specifies the type of graph associated to the use case: Barchart, PieChart, LineChart, BubbleChart and RadarChart</td>
</tr>
<tr>
<td>fileName</td>
<td>String</td>
<td>Specified the name of the file to be considered as input. Needs to come in the structure of &lt;filename&gt;.&lt;filetype&gt;</td>
</tr>
<tr>
<td>Xattribute</td>
<td>String</td>
<td>Specifies the attribute defined as X</td>
</tr>
<tr>
<td>XattributeOrder</td>
<td>OrderType</td>
<td>Specifies the type of order associated to this attribute: AlphabeticalA-Z, AlphabeticalZ-A, Chronological, NumericalAsc, NumericalDec or Custom</td>
</tr>
<tr>
<td>Yattribute</td>
<td>String</td>
<td>Specifies the attribute defined as Y</td>
</tr>
<tr>
<td>Zattribute</td>
<td>String</td>
<td>Specifies the attribute defined as Z (only applicable when ChartType != BubbleChart)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XisEntityDA-DEAssociation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associates a XisDataAnalyticsUseCase to the XisDataEntities.</td>
</tr>
</tbody>
</table>

The Xattribute, Yattribute and Zattribute (only used as an attribute when ChartType == BubbleChart) tagged value should respect this format:
- <DomainEntityName>.<DomainEntityAttributeName>;

The Zattribute can also be used in string format for when ChartType != BubbleChart being that, if it’s a valid condition, Yattribute must respect it. (Example: Yattribute > 6)
Appendix B

Appendix B
User Session Guide


The objective of this pilot-user test session is to perform an evaluation of XIS-Analytics (both language and framework) by users that are not familiarized with it, in order to detect bugs or user limitations. The evaluation method involves a dataset in the form of a JSON file, already structured as the input for XIS-Analytics. This dataset is composed with information from Riot Endpoint (https://developer.riotgames.com). The creation and interpretations of the data visualizations using the XIS-Analytics technology by the users, will be used to assess the language usability and to study further improvements to both the language and the framework. The case study application description is as follows:

**Case Study - Riot Games League of Legends**

Riot Games is a video game publisher and developer, primarily known for the game League of Legends, a real-time strategy video game where players assume the role of a "summoner" and control characters named "champions". This dataset is composed with information from Riot Endpoint (https://developer.riotgames.com) built as JSON file respecting the structure require by XIS-Analytics. The dataset is composed by three main entities: Summoner, SummonerStatus and TopUsedChampions. It has been built with the intention of using the Summoner entity as the fact entity, thus all entities have the summonerID acting as a fact key. The Summoner entity possess the main attributes regarding the player: "summonerName", "summonerID" and "rank" (ranging from 0-15, being 15 the highest). SummonerStatus contains all the information regarding the games of each summoner as a player, having quantified information such as the number of "wins", "kills", "assists" and "minionKills". The TopUsedChampions contains the most used character of the players with the quantified information of games when they have been used, being the attributes: "championName", "games", "wins" and "losses".

**Test Conditions:**

- Tests are conducted in the laboratory (controlled environment);
- The task must be performed without previous use and learning (for the first time);
- **The user must have a computer running Windows and previously installed Java Runtime Environment (recommended version 7) and Sparx Enterprise Architect (version 7.5, 10 or above)\(^1\);**
  - Direct Observation, i.e., while users perform the assigned task, their behavior and performance can be logged;
  - Users can think out loud and share ideas if they want;

---

\(^1\) Users that are from IST can have access to free and full licensed versions of Enterprise Architect, available in https://delta.ist.ist.utl.pt/software/ea.php. For other users, there is a 30 day trial version that can be downloaded from http://www.sparxsystems.com/products/ea/trial.html.
- The evaluator does not interact with the users until the tests are finished (except in case of blocking errors);
- The session will last 50 minutes (at most)
- The user must fill a survey in the end – available at http://goo.gl/forms/woIOZoa9CN

**Instructions:**

1. Download the XIS-Analytics EA Plugin installer from here: https://goo.gl/QNmYzj
2. Install the XIS-Analytics EA Plugin. This plugin provides an extension for EA which contains information about the XIS-Analytics profile (language), its diagrams, toolboxes and project template. Functions provided by the XIS-Analytics framework (like validation and generations) are also provided by the plugin.
3. Open EA and create new Project.
4. Upon the prompting of the Model Wizard window, select the “XIS-Analytics Framework” technology, then check “XIS-Analytics Framework model” option and confirm your choice. On the right side, it should appear a Package Diagram with 2 packages (one for each view of XIS-Analytics).
5. Begin by modeling the Domain Entities View:
   a. Create a XisEntityFact named “Summoner” and give it the following XisEntityAttributes:
      i. “summonerID” with type “int”;
      ii. “summonerName” with type “String”;
      iii. “rank” with type “int”;
   b. Create a XisEntityDimension named “SummonerStatus” and give it the following XisEntityAttributes:
      i. “wins” with type “int”;
      ii. “kills” with type “int”;
      iii. “assists” with type “int”;
      iv. “minionKills” with type “int”;
   c. Create a XisEntityDimension named “TopUsedChampions” and give it the following XisEntityAttributes:
      i. “championName” with type “String”;
      ii. “games” with type “int”;
      iii. “wins” with type “int”;
      iv. “losses” with type “int”;
   d. Connect the “Summoner” entity to the “SummonerStatus” entity using a XisFactAssociation.
   e. Connect the Summoner” entity to the “TopUsedChampions” entity using a XisFactAssociation.
6. Make sure to save the Model.
7. Continue by modeling the Data Analytics View:
   a. Create a XisActor named “User”.
   b. Create a XisDataAnalyticsUseCase named “EvaluationTest”; Double-click on it and select the “XIS-Analytics” menu (to the right) to see its tagged values:
      i. Set the taggedvalue “filename” to “RiotGamesJSON”;
      ii. Set the tagged value “Chart” to “RadarChart”;

73
ii. Set the tagged value “name” to “RiotGames”;
iii. Set the tagged value “Xattribute” to “Summoner.summonerName”;
iv. Set the tagged value “XattributeOrder” to “NoOrder”;
v. Set the tagged value “Yattribute” to “TopUsedChampions.championName”;
vi. Set the tagged value “Zattribute” to “NULL”;

8. Make sure to save the Model.

9. Validate the Model:
   a. Click on the “Extension” menu option, select “XIS-Analytics Plugin” and click on the “Validate Model” option.

10. (if the model validation ended with no errors) Apply the Model-to-Code generation, to automatically generate the code for the data visualization, otherwise fix the errors and try again.
   a. To do so, click on the “Extension” menu option, select “XIS-Analytics Plugin” and click on the “Generate Code” option. Choose the path on which you would like the code to be generated to.
   b. Click the “Generate!” button.

11. Check if the target generation folder contains the source code of your visualization.

12. Copy the provided JSON file to the directory of the generated file.

13. (if you have python installed) Open the command prompt and go to the directory of the generated file. When you are there, write the command “python –m SimpleHTTPServer 8888”. After doing so, a python server is generated. Open Google Chrome and go to “localhost:8888”. Open the generated file and enjoy the data visualization! (Needed because of the Anti-XSS filter).

14. After generating the first use case proceed to:
   a. Create a XisDataAnalyticsUseCase named “EvaluationTest2”; Double-click on it and select the “XIS-Analytics” menu (to the right) to see its tagged values:
      i. Set the tagged value “filename” to “RiotGamesJSON”;
      ii. Set the tagged value “Chart” to “BarChart”;
      iii. Set the tagged value “name” to “RiotGames”;
      iv. Set the tagged value “Xattribute” to “Summoner.summonerName”;
      v. Set the tagged value “XattributeOrder” to “AlphabeticalAZ”;
      vi. Set the tagged value “Yattribute” to “Summoner.rank”;
      vii. Set the tagged value “Zattribute” to “NULL”;
   b. Repeat step 8 to 11.
   c. Since the JSON file should already be in the generated folder, repeat step 13.

15. Remember to please remember to fill the survey at http://goo.gl/forms/woIOZoa9CN.
Appendix C

Appendix C
## Survey 12 responses

### Summary

#### Age

<table>
<thead>
<tr>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>27</td>
</tr>
<tr>
<td>27</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>21</td>
</tr>
</tbody>
</table>

#### Gender

<table>
<thead>
<tr>
<th>Gender</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>11</td>
<td>91.7%</td>
</tr>
<tr>
<td>Female</td>
<td>1</td>
<td>8.3%</td>
</tr>
</tbody>
</table>

#### Degree

- [Diagram showing degree distribution]
Number of Years of Professional Experience

<table>
<thead>
<tr>
<th>Years</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
</tr>
</tbody>
</table>

Knowledge and Previous Experience with UML

<table>
<thead>
<tr>
<th>Experience</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>9</td>
<td>75%</td>
</tr>
<tr>
<td>No</td>
<td>3</td>
<td>25%</td>
</tr>
<tr>
<td>Total</td>
<td>12</td>
<td>100%</td>
</tr>
</tbody>
</table>
How suitable is the size (number of stereotypes) of the Language? [XIS-Analytics Language (UML Profile)]

How easy is to use the notation (defined as a UML Profile)? [XIS-Analytics Language (UML Profile)]

How easy is to learn the language with the star schema concepts? [XIS-Analytics Language (UML Profile)]

How easy is to learn the language with the data visualization concepts? [XIS-Analytics Language (UML Profile)]
How suitable is the language to be applied to a data analytics problem? [XIS-Analytics Language (UML Profile)]

How do you rate the usability of the XIS-Analytics plugin in EA? [XIS-Analytics Framework (EA Development Environment)]

How do you rate the usability of the Visual Editor? [XIS-Analytics Framework (EA Development Environment)]
How do you rate the usefulness of the Model Validator? [XIS-Analytics Framework (EA Development Environment) ]

How do you rate the simplicity of the Model-to-Text transformation process? [XIS-Analytics Framework (EA Development Environment) ]

How do you rate the productivity of XIS-Analytics to deduce the value of a dataset? [XIS-Analytics Framework (EA Development Environment) ]
How likely would you use a tool such as XIS-Analytics for a preleminary dataset evaluation? [XIS-Analytics Framework (EA Development Environment)]

How would you say the generated data visualization look? [XIS-Analytics Data Visualization]

Using XIS-Analytics multiple charts, how do you think are your odds of extracting conclusions out of the dataset? [XIS-Analytics Data Visualization]
<table>
<thead>
<tr>
<th>Rating</th>
<th>Frequency</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Very Bad</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>2 - Bad</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>3 - No opinion</td>
<td>1</td>
<td>8.3%</td>
</tr>
<tr>
<td>4 - Good</td>
<td>8</td>
<td>66.7%</td>
</tr>
<tr>
<td>5 - Very good</td>
<td>3</td>
<td>25%</td>
</tr>
</tbody>
</table>