XIS - Analytics: a Domain Specific Language for Data Analysis

(Extended abstract of the MSc dissertation)

Afonso Pereira da Silva
INESC-ID Lisbon
Instituto Superior Técnico
Lisbon, Portugal
afonso.silva@ist.utl.pt

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;

I. INTRODUCTION

We live in a time where the amount of data that is been generated every day for the past few years generated massive and complex datasets, which are continually growing larger and larger. 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[1][22]. This edge comes in the form of better and faster decision making as well as developing new products and services. However, the analysis and exploration of any dataset is expensive both in time and resources, thus being a risk if no value is found.

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. 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)[2] like the one suggested in this dissertation. MDE, or its development process, Model-Driven Development (MDD)[3] 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.

Having said that, XIS-Analytics intends to work as a solution to address the problem of the complexity of data analysis and exploration. As a proposed solution, XIS-Analytics is composed by a Domain Specific Language (DSL), the XIS-Analytics language, and its supporting framework, the XIS-Analytics framework, which intent to work as an innovative solution to the problems mentioned earlier.

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 XIS-Analytics framework, responsible for the usability of the XIS-Analytics 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) and the Eclipse Modeling Framework (EMF). Currently the XIS-Analytics framework requests as an input structured JSON file corresponding to the dataset.

Both of these components of XIS-Analytics will be further elaborated in this document.

This research work has been conducted in an iterative and gradual way following the Action Research methodology[4] throughout the past year, resulting in a publication to an international conference[5] as well as the development and preliminary evaluation of the XIS-Analytics approach.

II. BACKGROUND

XIS-Analytics is based on the XIS language [8][9] and a more recent work, the XIS-Mobile [7][10]. As such, it utilized some of the same concepts. This section introduces to some those concepts that underline this research work such as MDE, DSL.

The usage of MDE is to both try to platform fragmentation problems and mitigate software complexity. In the subject of XIS-Analytics we are more interest in the later, but, in essence, MDE is a software development methodology that recur to the use of domain models to move the source code development process to a more abstract level of specification. The resulting models consist in abstract representations of concepts specific of a certain domain problem [6]. The main advantages on using an MDE approach is having the model guides all the development activities resulting in quality improvements, increased productivity and shorter time to market. But as stated before, the at most reason why MDE is utilized in XIS-Analytics is the ability to specify the structure of a dataset, and the definition of the data visualizations the user intends to generate.

A DSL, is a language targeted to a specific domain which aims to describe a system using high level specifications. A DSL is expressed making use of domain concepts, resulting in a language which is easier to read, understand, validate and communicate with[7]. By definition, DSLs generate applications from a specification making use of a compiler which supports them. DSLs are mention that can improve productivity, reliability, maintainability and portability, being that DSL and be either textual or visual.

III. XIS-ANALYTICS LANGUAGE

As stated before, XIS-Analytics is a DSL which has as an ultimate goal to evaluate a provided dataset and return to the user a set of data visualizations. There data visualizations are generated according to specifications provided by the user. XIS-Analytics language is divided in two main views: (A) Domain Entities View; and (B) Data Analytics View. These two view correspond respectively to the two main steps of data exploration[14]: data mapping, and specifying how the user wants to explore it. These two views are explain in this section.

For better understanding and simplicity of the explanation, a small case study dataset is provided (Fig. 1) using a sub-set of IMDB (Internet Movie Database) data, available at ftp.fu-berlin.de. This sub-set is already structured in the format that XIS-Analytics currently expects as input, being the only requirement to make use of the language and framework.

Case Study – IMDB data

IMDB is an online database for films, television, programs and videos, including cast and all kinds of information. For this case study, the created JSON file possess the following structure and information:

```
Data{   Movie { Movie1{attr1; attr2; … }; Movie2{attr1; attr2; …}; … }
        Actor { Actor1{attr1; attr2; … }; Actor2{attr1; attr2; …}; … }
        Location { Location1{attr1; attr2; … }; Location2{attr1; attr2; …}; … }
        Ranking { Ranking1{attr1; attr2; … }; Ranking2{attr1; attr2; …}; … }
    }
```

Fig. 1. IMDB Case Study

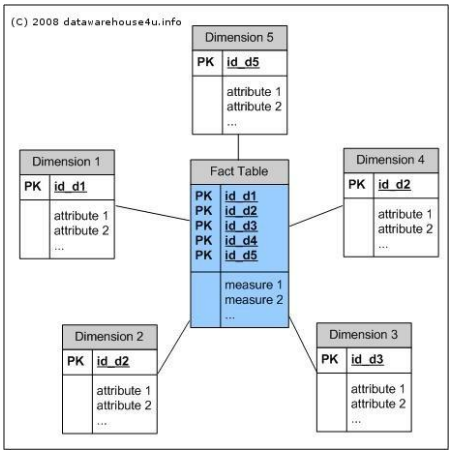

Fig. 2. Star Schema Model

A. Domain Entities View

Before going more in depth on the first view there is the need to briefly introduce the Data Warehousing concept and the Star Schema Model, since the Domain Entities view as design based on it.

A Data Warehouse (DW) is a centralized repository which stores a collection of data from multiple information sources, transforming it after into a common, multidimensional data model for efficient querying and analysis. The most used Schemas used on a DW architecture are the Star schema, the Snowflake schema and the Fact Constellation schema, being that the decision of what model should be used depends upon the analysis of the project requirements and preferences [15][16]. Being the most general of this schema models, the star schema was the one chosen as the best fit to fulfil the goals that had been defined for XIS-Analytics.

The Star Schema (Fig. 2) is the simplest DW schema [17], and it's called as such because the diagram resembles a star. The center of the star consists of the fact table and the extremities are the dimension tables. The fact tables are typically composed by two types of columns: foreign keys to dimension tables and measures those contain numeric facts. The dimension tables process primary keys which are part of the composite primary key of the fact table, and dimensional attributes to help describe the dimensional value. Some of the benefits of using a star schema model is the simple structure, resulting in a easiness to understand the schema, and since it's the most commonly used in data warehouse implementations, there is a large number of business intelligence tools which support it.

The Domain Entities View (Fig. 3) is based on the star schema model for the reasons stated above. As stated previously, XIS-Architecture currently receives a structured JSON file, for which the user should build a Domain Entity View on how does he want to interact with the data, making some of the data the main focus (the XISEntityFact), while making notice of what dimensions to the focus (the XISEntityDimension. There can be multiple of these view for

the same JSON file, depending if the user wants to have multiple takes on the same data, but it is not advised since better results on the generated charts can be obtain if specific JSONs are created for a specific view, similar to the multidimensional cube on data warehouses (one specific cube for a specific set of data).

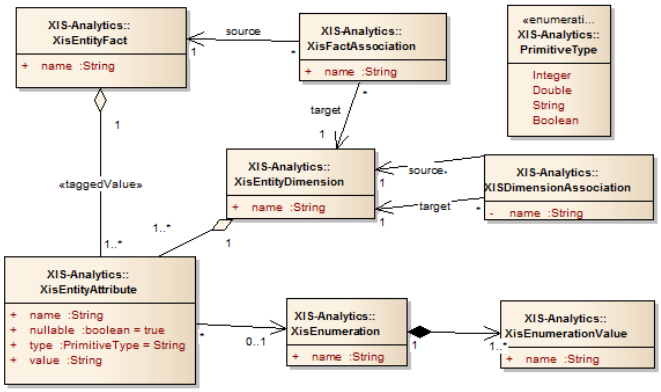

Fig. 3. Metamodel of the Domain Entities View

Following our example, if user intents to query the data from the Movies stand point of view, such a domain entity view as the one in Fig. 4 would have to be specified by the user. As it can be seen, in this example the user selected the "Movies" entity as the fact entity, meaning he intends to query by it. In this specific case, the user decided only to include the dimensions "Actors" and "Character", saying he will only make use of the data contain in these three entities Distintions from the standart star schema model that is shown on Fig. 2 is the lack of formal key, since it was a designer choice to include the possibility for the user to, with the same cube, query diferently with a variety of the combinations of the attributes, as it will be further explained while addressing the next view.

B. Data Analytics View

The Data Analytics View (Fig. 5) is where the user will specifify his preferences on how he wishs to query the data taking one Domain Entity View in consideration.

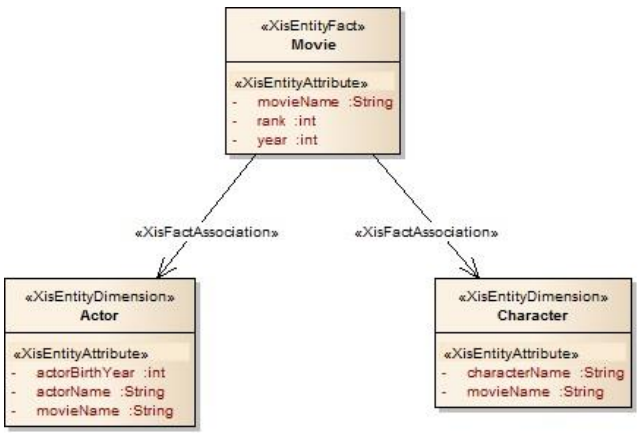

Fig. 4. Domain Entity View for the IMDB case study dataset

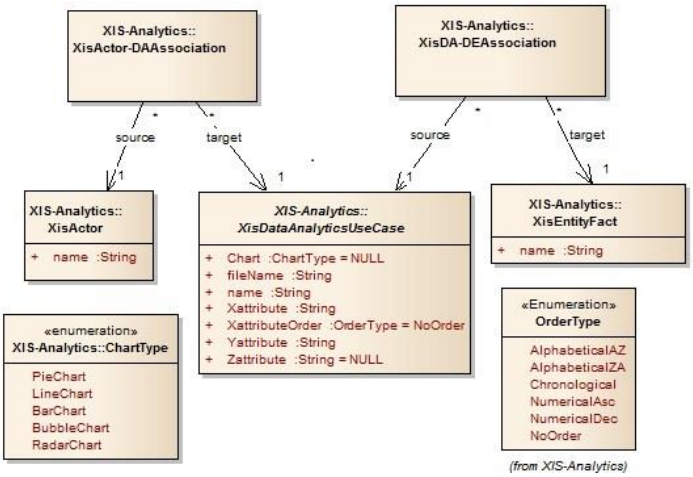

Fig. 5. Metamodel of the Data Analytics View

The class from this view that requires the most attention is the XISDataAnalyticsUseCase. In this class, the user defines every thing related to the specifics of the generated dashboards from what file the user wants to take the information from, to what data attribute should be considered and in what axis should they be considered to, as well as how should they be ordered on those same axis (if appliable), up to what type of chart does the user wants to represent on each specific XISDataAnalyticsUseCase. Every speficication will generate diferent code and all are generated at the same time.

Following our example, and making use of the previoulsy specified Domain Entity View for it (Fig. 4), the Data Analytics View on Fig. 6 would be a valide one to continue the process of the data analysis. As it can be seen, most of the work of the specification is centralized on the XISDataAnalyticsUseCase class with providing the paths to the attributes that are going to be utilized for each of the components of the visualizationt. The terms of Xattribute and Yattribute are still used for the visualization where those name are not valid, such as Radar Chart and Pie Chart, but as specified in the longer version of the profile, lossely explaining, the Xattribute will allways be the attribute where Yattribute is inqueried upon. (The example of the query writen in informal spoken language would be: "All the rankings from all this movies")

The visualizations XIS-Analytics currently offer are chosen from what, according to our research and evaluation, some of the most used, most generic and able to the most different types and kinds of data. Thus, XIS-Analytics offers: Pie Chart, Line Chart, Bar Chart, Bubble Chart and Radar Chart.

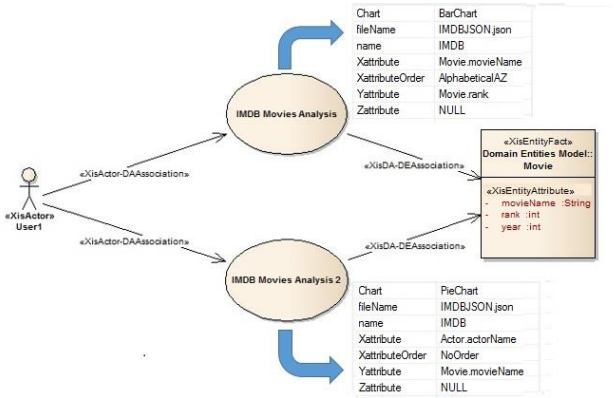

Fig. 6. Data Entity View for the case study

With this set of chart available, a large amount of visualizations can be generated, since most of the relations of different types of data can be represented between them. With the exception of Bubble Chart, all of this chart types are to represent two dimensions by themselves or at maximum three if there is an implicit correlation between the data and the analysis specifications.

Following the data analytics view specified in Fig. 6 the generated chart are shown in Fig.7 and Fig.8. The generated visualizations are interactive, responsive and scalable. For an example of intecation, in Fig. 7, after the chart is generated, a bar at the top of the chart let the user decide what part of the data he wants to explore. In the same Fig. on top we have an example of a sub set, and after the user making use of the bar and expanding the amount of the dataset utilized, on the bottom we can see the full data set represented. Providing this option to the user is an important capability for data exploration applications, since it allows the user to better understand all parts of it.

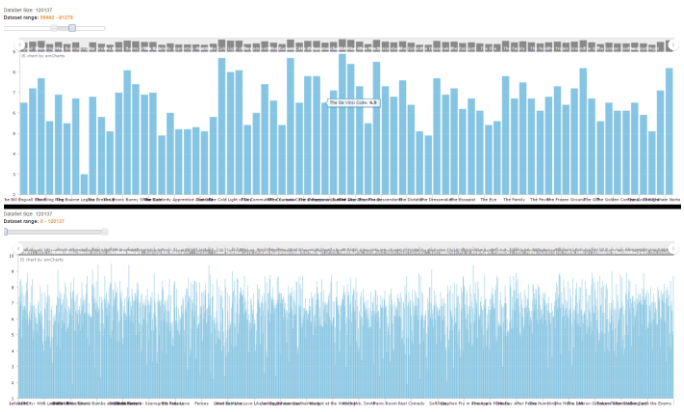

Fig. 7. The generated bar chart for the case study (on the top, a sub set, and on the bottom the full set)

Fig. 8. The generted pie chart for the case study

IV. XIS-ANALYTICS FRAMEWORK

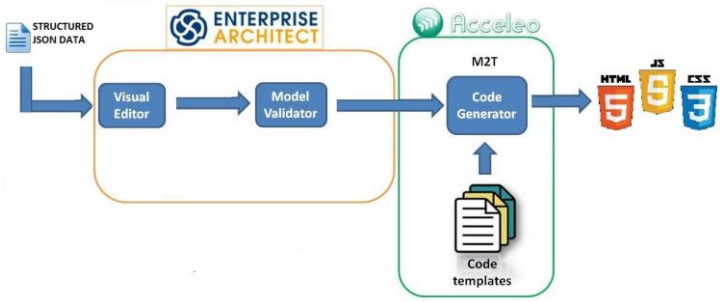

Fig. 9. Simplified development approach with the XIS-Analytics framework.

The current version of XIS-Analytics framework receives as input a JSON file, which needs to be structured as shown in Fig. 1, which can be done by using any JSON Editor programs such as Pentaho's spoon[11], JSONedit or Altova, just to name a few. The decision to make the input as a JSON is given to the fact that this format is lightweight and mostly used for data interchanging, which is generally either a collection of name/values or an ordered list of values, making it easy to understand for the user.

The XIS-Analytics becomes a more relevant language together with a MDD-based framework, and the development process using XIS-Analytics is comprised by three steps: (1) the definition of the model using Visual Editor, (2) its validation through the Model Validator, and finally, (3) the generation of the source code corresponding to the visualizations. The Spark Systems Enterprise Architect (EA), along with the Eclipse Modeling Framework (EMF), together with some compatible plug-ins, were the technologies chosen to help develop XIS-Analytics.

First the Visual Editor is implemented on top of EA, fully compliant with the OMG specification for UML2, therefore offering a very good support for UML profiles, allowing the user for good and thorough specifications of both of the needed views in XIS-Analytics

Second, the model validation, being an issued that must be taken into account, since, in order to avoid errors by the user and consequently, enhance the quality of the generated code and dashboards, restrictions must be defined. When thinking about the UML models validation, OCL is the standard

language commonly used, but unfortunately, due to several limitations with stereotypes validation and onion developments of the OCL plug-in for EMF, the use of OCL wouldn´t be the best option. Therefore, to define the constraints for the validation, it was chosen to use the Model Validation API, which is provided by EA. Even not being a standard like OCL, it's a solution which allows the assignment of constrains, definition of custom error and warning messages and the navigation to the element that caused it.

Third, the XIS-Analytics generator, in charge of performing model-to-text transformations with the guidance of code templates. The generator is based on an Eclipse plug-in, Acceleo which is a template-based code generator framework that impends the Managed Object Format (MOF) Model to Text Language (MTL) standard while supporting any kind of model compatible with EMF. The code templated are comprised by two parts: (1) regular text, the static part of the template and (2) the annotations, the dynamic part of the template. For now, XIS-Analytics only support the generation of javascript and HTML5, making use of libraries such as amChart, D3, JQuery and bootstrap. After the code and dashboards are generated, some interactions are provided at run time, but if the user wishes to edit anything, he will have to specify it on the respective code himself, being that all the generated files are built with good programing practices and are as good they were men made.

On a side note, during the early stages of this research project, a plugin which makes use of JSON Discoverer. The JSON Discoverer is a tool that allows the user to discover the implicit schema on JSON[18]. 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. The plugin produced intended to make use of these capabilities to provide a domain view to XIS-Analytics. The architecture of this plug-in can be seen in Fig. 10. It starts by using JSON Discoverer on the desired JSON file, returning the respective Ecore model. Then, making use of the ATL language[19], the UML is generated making use of the XIS-Mobile profile stereotypes of the domain view, since the XIS-Analytics profile was not build 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 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. It is currently used only as a tool to verify the structure of the JSON files at a development stage.

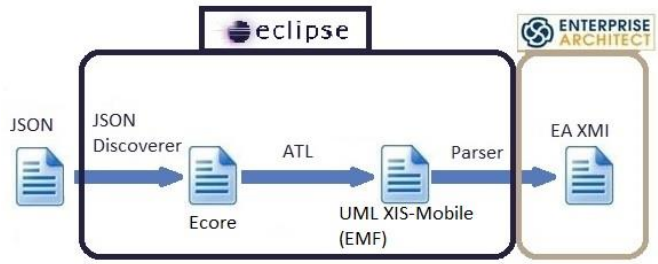

Fig. 10. JSON Discoverer Plug-in adaptation arquitecture

V. RELATED WORK

Pentaho's spoon[11] 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[11]. 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[12] is a web analytics solution offered by Google, and it is currently the most widely used web analytics service on the internet[13], 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.

VI. EVALUATION

The evaluation performed to XIS-Analytics was a twofolded one. First the M2T capabilities of XIS-Analytics were evaluated through the analysis of the limitations of the code generated using case studies, making use of the types of data visualizations currently supported by XIS-Analytics. The second part of the evaluation was an assessment session of XIS-Analytics by third parties, focusing on three aspects of XIS-Analytics: (1) the Language, (2) the Framework and (3) the generated Data Visualizations.

Regarding the first part of the evaluation, 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 Fig.11 was used to “mass test” all of the data visualizations, by making use of the capability of XIS-Analytics to generate multiple files for a simple Data Analytics View, five case studies where created.

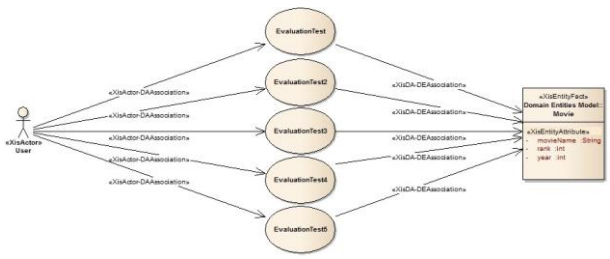

Fig. 11. Example of the mass testing Data Analytics View.

The data types can be divided in: Nominal, Ordered, Binomial, Ordinal, Quantitative, Ratio and Continuous[20]. 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.

Fig.12 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.

Combination / Chart type	X field	String	String	Value	Value	String	String	Value	Value
	Y field	String	Value	String	Value	String	Value	String	Value
	Z field (only considered for Bubble chart	String	String	String	String	Value	Value	Value	Value
Bar		✓	✓	✓	✓	N/A	N/A	N/A	N/A
Line		X	✓	X	✓	N/A	N/A	N/A	N/A
Pie		✓	✓	✓	✓	N/A	N/A	N/A	N/A
Bubble		X	✓	X	✓	X	✓	X	✓
Radar		✓	✓	✓	✓	N/A	N/A	N/A	N/A

Fig. 12. Example of the mass testing Data Analytics View.

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 pointing out that the set of visualizations that XIS-Analytics currently supports, are able to represent most of the possible combinations.

The second part of the evaluation of XIS-Analytics, a user session assessment was planned, since t 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 from different ages and backgrounds. 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. Thereafter, a script was provided to the participants describing a case study, and with it, the participants had to perform the set of tasks described. 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 average of the analysis of the results gathered for each of these aspect is shown on Fig. 13.

	XIS-Analytics		
	Language	Framework	Generated Data Visualization
Average	4.382	4.252	4.375

Fig. 13. Example of the mass testing Data Analytics View.

The results of the session were encouraging of the XIS-Analytics approach with overall positive scores in all three analyzed aspects. Nevertheless, is 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[21].

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.

VII. 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[9] and XIS-Mobile[7], 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.

VIII. 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 everyone, 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, during all iteration of the action research, through 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.

REFERENCES

[1] Liebowitz, Jay, ed. *Big data and business analytics*. CRC Press, 2013.

[2] Mellor, Stephen J., Tony Clark, and Takao Futagami. "Model-driven development: guest editors' introduction." *IEEE software* 20.5 (2003): 14-18.

[3] Beydeda, Sami, and Matthias Book. *Model-driven software development*. Vol. 15. Heidelberg: Springer, 2005.

[4] Baskerville, Richard L. "Investigating information systems with action research." *Communications of the AIS* 2.3es (1999): 4.

[5] Ribeiro, André, Afonso Silva, and Alberto Rodrigues da Silva. "Data Modeling and Data Analytics: A Survey from a Big Data Perspective." *Journal of Software Engineering and Applications* 8.12 (2015): 617.

[6] Silva, A.R. et al.: Integration of RE and MDE paradigms: The ProjectIT Approach and Tools, IET Software Journal (2007)

[7] A. Ribeiro and A.R. da Silva. Xis-mobile: A dsl for mobile applications. In Symposium On Applied Computing (SAC), 2014, ACM

[8] A.R. Silva, G. Lemos, T. Matias, and M. Costa. The xis generative programming techniques. In Computer Software and Applications Conference, 2003. COMPSAC 2003. Proceedings. 27th Annual International, pages 236{241, Nov 2003.K. Elissa, “Title of paper if known,” unpublished

[9] A.R. da Silva. The xis approach and principles. In Euromicro Conference, 2003. Proceedings. 29th, pages 33{40, Sept 2003.

[10] A.R. da Silva, J. Saraiva, R. Silva, and C. Martins. Xis-uml pro_le for extreme modeling interactive systems. In Model-Based Methodologies for Pervasive and Embedded Software, 2007. MOMPES '07. Fourth International Workshop on, pages 55{66, March 2007.

[11] Casters, Matt, Roland Bouman, and Jos Van Dongen. *Pentaho Kettle solutions: building open source ETL solutions with Pentaho Data Integration*. John Wiley & Sons, 2010.

[12] Clifton, Brian. *Advanced web metrics with Google Analytics*. John Wiley & Sons, 2012.

[13] Randall, John Herman. "The making of the modern mind: a survey of the intellectual background of the present age." (1976).

[14] Friedman, Jerome H., and John W. Tukey. "A projection pursuit algorithm for exploratory data analysis." (1974): 881-889.

[15] Devlin, Barry, and Lynne Doran Cote. *Data warehouse: from architecture to implementation*. Addison-Wesley Longman Publishing Co., Inc., 1996.

[16] Chaudhuri, Surajit, and Umeshwar Dayal. "An overview of data warehousing and OLAP technology." *ACM Sigmod record* 26.1 (1997): 65-74.

[17] Adamson, Christopher. *Mastering data warehouse aggregates: solutions for star schema performance*. John Wiley & Sons, 2012.

[18] Izquierdo, Javier Luis Cánovas, and Jordi Cabot. "Composing JSON-based Web APIs." *Web Engineering*. Springer International Publishing, 2014. 390-399.

[19] Jouault, Frédéric, et al. "ATL: A model transformation tool." *Science of computer programming* 72.1 (2008): 31-39.

[20] Agresti, Alan. *Analysis of ordinal categorical data*. Vol. 656. John Wiley & Sons, 2010.

[21] Nielsen, Jakob, and Thomas K. Landauer. "A mathematical model of the finding of usability problems." *Proceedings of the INTERACT'93 and*

CHI'93 conference on Human factors in computing systems. ACM, 1993.

[22] Mayer-Schönberger, Viktor, and Kenneth Cukier. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt,2013.