OntoIFML: Automatic Generation of Annotated Web Pages from IFML and Ontologies using the MDA Approach: A Case Study of an EMR Management Application

Naziha Laaz and Samir Mbarki
MISC Laboratory, Faculty of Science, Ibn Tofail University, Kenitra, Morocco

Keywords: SWAs, MDWE, MDA, ODM, IFML, EMR, e-Health, Ontologies.

Abstract: The Semantic Web is a complex ecosystem that integrates a set of software devices necessary to model, disseminate and exploit the knowledge stored in an ontology. The development of its applications is not an easy task, it is costly and time-consuming. It consists of the different design and modeling phases and sub-phases, each with its own modeling language. Among its most delicate development phases, ontology modeling is mentioned, as well as, the integration of semantic annotations in web pages. In this paper, we present OntoIFML a new MDA-based approach used as an IFML-based tools plugin to facilitate the development of web applications relying on Semantic web paradigm. With this approach, we explore the possibilities for automating the generation of domain ontologies and interfaces annotated by semantic vocabularies from the front-end IFML model. The approach is illustrated by an EMR management application. The result obtained have shown the effectiveness of our approach.

1 INTRODUCTION

Concretized more than 10 years ago by the W3C (World Wide Web Consortium), Semantic Web is, the metadata web (Berners-Lee et al., 2001) continues to rise in terms of established links between information that had previously been treated separately. Therefore, the semantic web applications integrate the web with the possibility to aggregate several data linked together: either "semantically" or by the attributes that distinguish them. Technically speaking, the semantic web relies on various standards and semantic technologies (Klašnja-Milićević et al., 2017). They can be used to standardize the terms and concepts of a domain, facilitating communication and sharing of knowledge. These technologies can also semantically annotate a set of resources to perform more "intelligent" researches and find assets to meet certain needs. However, the development of semantic web applications is not a simple task considering the variety of technologies behind. This affects the amount of requested time to learn and understand these technologies for a web developer.

The implementation of this kind of applications needs to be improved by using MDA-based model-driven development techniques for generating platform-compatible user interfaces.

This paper emphasis on the modeling and generation of ontologies, as well as their use for semantic web applications. Indeed, our process centers around the production of several models, going from abstract to concrete to finally obtain interfaces of Semantic Web Applications (SWAs). We use two independent platform models; PIMs, derived from two standards that have visual design annotation, this reduces the time and resources of development. While Unified Modeling Language (UML) is used to generate a domain ontology file, Interaction Flow Modeling Language (IFML) is used to derive the set of annotated interfaces. Then through well-defined transformation rules, we will get PSMs respecting Ontology Definition Metamodel (ODM) and Hypertext Markup Language (HTML5), and then we mergers both models in the code generation phase to obtain semantically annotated interfaces with the domain ontology derived by the class diagram. To validate our ideas and examine the applicability of our approach, we applied the process to a case study of an Electronic Medical Records (EMR) management application.

Laaz, N. and Mbarki, S.
OntoIFML: Automatic Generation of Annotated Web Pages from IFML and Ontologies using the MDA Approach: A Case Study of an EMR Management Application.
DOI: 10.5220/0007402203530361
In *Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2019)*, pages 353-361
ISBN: 978-989-758-358-2

Table 1: Presenting Existing MDA-based approaches using IFML in the PIM level that support web apps.

Studied papers	IFML Design		Code generation					
	Manual design	By Editor	Components	Interactions	data	Semantic Annotations	Layout	Style
(Acerbis et al., 2015a)	---------	Y	Y	Y	Y	N	Y	Y
(Bernaschina et al., 2017a)	---------	Y	Y	Y	Y	N	Y	Y
(Bernaschina, 2017)	---------	Y	Y	Y	Y	N	N	N
(Rodriguez-Echeverria et al., 2018)	---------	Y	Y	Y	Y	N	N	N
(Laaz and Mbarki, 2016a)	Y	N	Y	Y	N	N	N	N
(Roubi et al,2016)(Roubi et al., 2016)	Y	N	Y	Y	N	N	N	N
(Laaz and Mbarki, 2016b)	Y	N	Y	Y	N	N	N	N
(Yigitbas et al., 2015)	---------	Y	Y	Y	Y	N	Y	Y

Y: Supported **N:** Not Supported

The rest of this paper is organized as follows. Section II analyzes the Model Driven Web Engineering (MDWE) approaches based on model driven architecture (MDA) and IFML. Section III outlines the Semantic Web and the MDA based standards. Section IV presents our approach and gives an overview of model driven development process (MDD) describing its different phases. In section V, we illustrate our proposal with an EMR case study. Finally, we show the results of our process and conclude by determining the outcome of our work and describing future works.

2 RELATED WORK

Focusing on the topic of model-driven user interfaces development integrating ontology and semantic annotations, several aspects must be taken into account. Based on MDA, our approach makes it possible to generate ontologies from UML class diagrams and derive annotated pages from IFML diagrams. Therefore, our work is related to and influenced by a wide range of research fields in the view of overcoming the gap between semantic web and Model-driven methodology. In the following, we cite existing MDA-based approaches taking as input IFML models for generating web applications and approaches for an MDA-based ontology development and set them in relation to our own solution.

Indeed, during these last few years, several research works have proposed model-driven approaches for the user interfaces and interactions development based on the IFML standard. IFML has rapidly advanced in the software industry and academia, under the motivation of many studies (Hamdani et al., 2018; Laaz et al., 2018; Queiroz et al., 2018; Wakil and N.A., 2017) and approaches. They have exploited the IFML in different domains such as: Interactive systems (Gotti and Mbarki, 2016), interactive visualizations(Morgan et al., 2018), Big Data (Bernaschina et al., 2017a), execution models (Gotti and Mbarki, 2016), mobile applications(Acerbis et al., 2015b; Bernaschina et al., 2017b; Brambilla et al., 2014; Huang et al., 2018; Umuhoza et al., 2015). Table 1 highlights the MDA-based approaches using IFML in the PIM level that support web apps code generation. We identified the evaluation criteria based on the development process of web applications and design experience to evaluate each of these approaches, and then, provide a comparative analysis. Therefore, "IFML Design" and "Code generation" have been chosen as major criteria. The IFML design must be developed either by the classical method using a dynamic instance directly from the IFML metamodel syntax, or from a graphical editor such as IFMLEditor (Ed-Douibi and Brunelière, 2015). As for code generation criterion, which is the final step of development process, it encapsulates various information relating to: Components, interactions, data, semantic annotations, layout and style. The selection of these criteria is based on and has been influenced by various sources as mentioned before. This set of criteria will be used to compare and examine these approaches.

After analyzing all these approaches, we notice that the described approaches enable the design and also support the generation of UIs of web applications, but there is no solution which permits to add semantic annotations to web pages. For the purpose of adding this feature to the generated code, it was absolutely necessary to prepare a literature study of different approaches based on MDA for ontology modeling.

The most relevant are: (A\s smann et al., 2006; Belghiat and Bourahla, 2012; Brockmans et al., 2006; Musumbu, 2013; Saripalle et al., 2013; Zedlitz et al., 2012). All these approaches did not propose an MDA-based ontology generation process that exhibits the ODM standard. This work will be the first approach proposed for the generation of domain ontologies from UML class diagrams and ODM models, as well as their use for SWAs development based on IFML.

3 BACKGROUND

3.1 MDA Related Standards

MDA is a branch of software engineering proposed by the OMG (Belaunde et al., 2003), which proposes an architecture that takes into account all phases of a project, ranging from the CIM model of requirements and subsequently the PIM model analysis and design, up to the PSM, the implementation model**.** This ensures a better development organization and allows a separation of the functional specifications of an information system from its implementation.

The MDA approach advocates the use of OMG standards. In this work, we considered the three standards: ODM, IFML, and UML as central elements for design, transformation of models and generation of final interfaces.

The ODM specification was appeared in 2007 for ontology modeling. ODM defines five metamodels; RDFS, OWL, Topic Maps, Common Logic and Description Logic, two UML Profiles (RDFS/OWL Profile, Topic Maps Profile) and a set of QVT mappings from RDFS/OWL to Common Logic, Topic Maps to OWL and UML to OWL. With ODM, it would be possible to profit more quickly from MDA's capabilities to develop ontologies.

On the other hand, IFML is a standard adopted by the OMG in 2013 (Brambilla et al., 2013). It is a platform independent metamodel, which provides a stable set of concepts that can be used to characterize the essential aspects of the user's interaction with a software application interface. Consequently, an IFML model supports a lot of design perspectives as depicted in figure 1. This perspective allows modeler to define elements of user interfaces, their relationships and dependencies between view elements or between view elements and action (Brambilla and Fraternali, 2014).

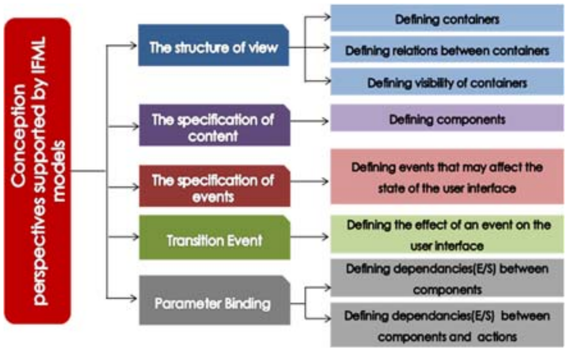

Figure 1: IFML Model Perspectives.

The present specification defines four main technical artifacts. This work uses the first artifact as a domain-specific metamodel to design graphical user interfaces of SWAs.

3.2 Semantic Web Applications (SWA)

Semantic Web is a component of web 3.0 standardized by the World Wide Web Consortium (W3C), bridging a gap between human's real-world domain knowledge and the machines. The main purpose of the Semantic Web is to guide the evolution of the Web to enable users without intermediaries to find, share and combine information more easily.

Compared to other web applications, SWAs use a pyramid of languages and standards for representing knowledge on the web by meeting the criteria of standardization, interoperability and flexibility (Klašnja-Milićević et al., 2017). As see in figure 2, a language of the top layer must be an extension of the language of the layer below. In fact, XML structures the content of documents by providing a basic syntax, but it does not describe the semantics of the document. RDF is a standard of W3C presented as a data model that describes the available resources on the web across RDF triples of "subject", "predicate" and "object". As for OWL, it represents an ontology. As a matter of fact, all these technologies cited above complement each other in order to build linked data database.

Domain ontologies are the most fundamental part of semantic web applications (Djurić et al., 2005; Wakil and Isa, 2015). Indeed, the SWAs are controlled by metadata, described by ontologies that will give understandable meaning by machine to its data. In the semantic web context, this notion can be used to standardize the terms and concepts of each domain, which facilitates communication and knowledge sharing, and this is done by annotating resources of web pages by semantic description using for example the RDFa syntax.

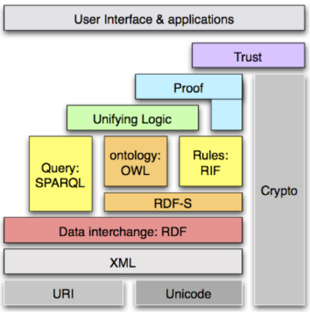

Figure 2: Semantic web stack, as presented by Tim Berners-Lee.

4 OntoIFML

This section gives an overview of our OntoIFML, an MDA-based approach, and how it could be used as a plugin for IFML-based tools, to design and generate automatically graphical user interfaces of SWAs.

5 OntoIFML

This section gives an overview of our OntoIFML, an MDA-based approach, and how it could be used as a plugin for IFML-based tools, to design and generate automatically graphical user interfaces of SWAs.

5.1 The Proposed Approach

The semantic web and the MDA were developed in parallel during this last decade, but by different communities. These two branches can be brought closer together by applying the MDA architecture in the semantic web (Gašević et al., 2007). Many authors have attempted so far to fill the gaps between them and have proposed several solutions, we may mention (Heinz et al., 2017; Partridge et al., 2018). Nevertheless, the only significant result of these efforts is the OMG initiative by defining the ODM standard. Then, the researchers of this group proposed mapping solutions between UML and OWL, but remains the exploitation of this mapping in the MDA process. So, if we want to provide a consistent solution with MDA proposals for the integration of ontologies in web pages, we should also support the automatic generation of a completely operational ontology definitions (for example in OWL) which are driven by models.

This work is built on this integration of OMG's efforts with the semantic web concepts to bring a new solution in order to generate annotated web interfaces by integrating automatically semantic meta data into these pages using the RDFa standard. So, the idea is to represent abstractly the graphical user interfaces accompanied by their data to reuse the common aspects in different applications, as well as the reduction of the repetitive tasks such as the integration of the annotations into web pages.

5.2 Process Overview

We propose an MDD process that allows the development of graphical user interfaces in the semantic web area. The process is based on the OntoIFML

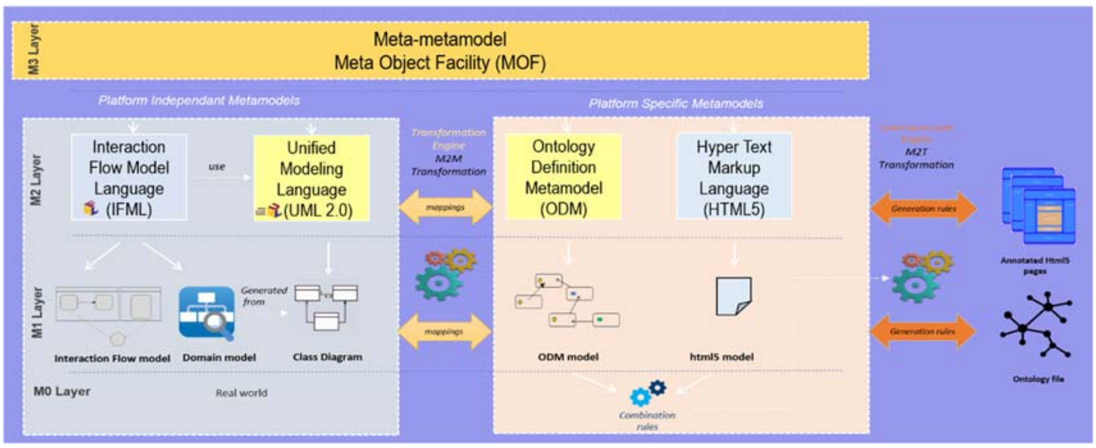

Figure 3: The MDD Process Overview.

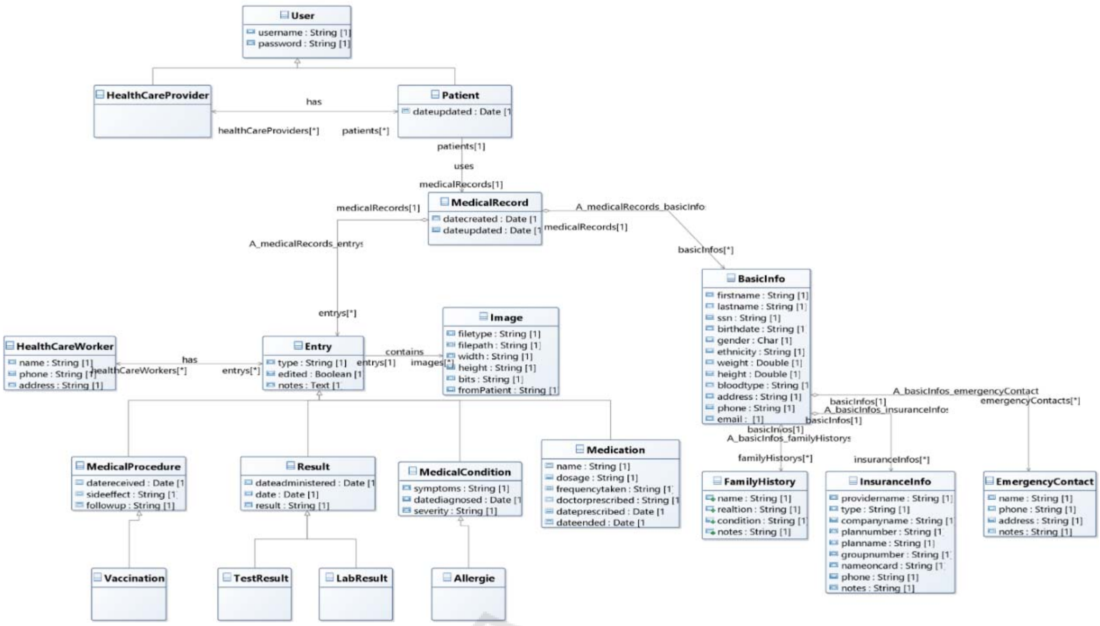

Figure 4: EMR Class Diagram.

approach. We implemented this process using IFML editor and several eclipse modeling tools. Indeed, we designed the IFML Model conformed to the PIM metamodel. This model consists of two sub models; the Interaction Flow Model and Domain Model. We elaborated PSM metamodels, then we established the transformation rules that gives us the models results which describe sufficiently Semantic Web Platform, focusing on domain ontology, the graphic aspect of the application and semantic annotations. Finally, we combined the two PSM models in the code generation phase to obtain UIs of the semantic web application. This behavior allows the automatic incorporation of domain ontology into web pages, and this enriches the interfaces with semantic features. The figure 3 depicts the architecture of the implemented approach. We were based on operational transformation language (QVT) (OMG, 2011) defined by the OMG to establish our model to model transformation.

- IFML2HTML5 Mappings

Table 2: IFML2HTML5 Mappings.

IFML		**HTML5**
Package	Element	
CORE	InteractionFlowModel	WebSite
Extensions	IFMLWindow	HTMLHtmlElement
Extensions	Form	FORM
Extensions	SimpleField	INPUT,TEXTAREA
Extensions	List	UL,OL
Extensions	Details	FIELDSET
Core	VisualisationAttribute	DIV,LABEL..

Table 2 and Table 3 show our main transformation rules.

- UML2ODM mapping

Table 3: Mappings between UM and ODM concepts.

UML	ODM
Model	*OWLGraph*
Package	*OWLOntology*
Class	*OWLClass*
AssociationClass	*OWLClass*
PrimitiveType	*RDFSDataType*
Enumeration	*EnumeratedClass*
Property	*OWLDatatypeProperty*
	OWLObjectProperty
	CardinalityRestriction
	MaxCardinalityRestriction
	MinCardinalityRestriction
	SymmetricProperty
	InverseFunctionalProperty
InstanceSpecification	*Individual*
Slot	*OWLStatement*

We used RDFS/XML as a format for the generated owl file, to do so, we defined the rules that border the construction of Ontology file. On the other hand, the generation of annotated web pages is done using the generation rules injected into the Acceleo template.

6 RESULTS AND EVALUATION

6.1 Case Study

To illustrate our proposal approach, we choose an EMR management application. This case study was not randomly chosen, given the importance of the e-health domain and the need to set up tools for the computerization of medical data processing with the aim of improving healthcare and reducing administrative tasks; specially, the management of EMRs. EMR is a web vocabulary for e-health, published in an ontology form. It is a digital version of the paper charts in a doctor's office (PMR). It consists of collecting information from the physician for their patients' diagnosis and treatment. This ontology is used to annotate web pages that describe a medication or patient. The pages are annotated according to the EMR vocabulary using the RDFa serialization format. On the other hand, the vocabulary is available in RDF / XML format; Turtle, JSON and the newest HTML5 microdata are also other serialization formats.

A data model resulting from a real analysis in the e-health domain for the EMR was presented as a class diagram in UML standard ("PMR - Droid," n.d.). In this context, UML was used to build a domain model that correspond to an abstract representation of data contained in hospitals.

The figure 4 shows the taxonomy of an EMR domain. The healthcare worker(provider) is in charge of inputted the medical data to make an EMR. However, the patient can navigate and view his medical record with an organized and easy way. This EMR domain provides a high-level view of the key types of medical information that includes medications, procedures, vaccinations, conditions, allergies, family history, test and lab results, insurance providers, and emergency contacts.

Figure 5: Scenario prototyping.

Thus, domain modeling naturally interplays with the modeling of the front end of the application. So, we choose the scenario shown in figure 5 (mockup), in which we designed three interfaces related to 'medications' in a high-level of abstraction:

- Interface 1: « add Medication» It represents a **Form** to add medication infos.
- Interface 2: «find Medication » it represents a **List** of medications.
- Interface 3: "display Medication" it represents **Details** where medication information's will be displayed. This page is annotated according to the EMR vocabulary using the RDFa serialization format.

6.2 Results

In order to validate the utility and applicability of our proposal, this section shows an evaluation of its application to the case study presented above. The main goal of this evaluation is to analyze the Model-driven development process that can be functional in real projects. We provide the IFML model, designed by IFML editor, which is basically the only input file to our generator.

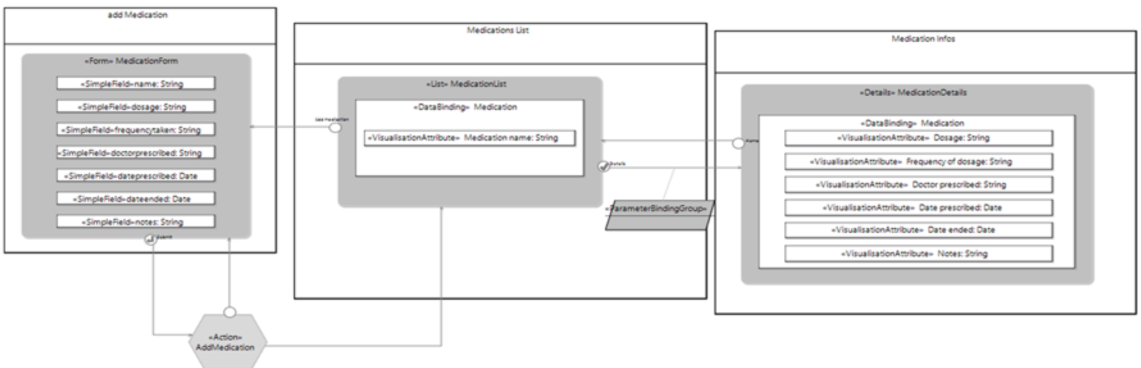

Figure 6: Medication pages designed at the high level of abstraction as an IFML diagram using IFML editor.

After the creation of an UML class diagram of the EMR domain, we proceed to the creation of the Core model which contains both interaction and domain model. From this abstract representation of interfaces associated with a domain model, we defined the three views that we have for our scenario. Figure 6 illustrates the EMR case study designed by the IFML graphic syntax: The ViewContainer initial page contains a ViewComponent displaying a form to add a medication information's by clicking on the **submit** event. The user can select medication and see its details by clicking on a **details** event. The user goes to the Medication Details page, where he can see the details of the selected medication.

```
transformation IFML2Html5(in Source:IFML,in umlModel:uml, out Target:Html5);

main() {

mapping InteractionFlowModel::Model2Web() : WebSite {

mapping IFMLWindow::Window2Page() : HTMLHtmlElement {

mapping Form::ViweElement2BodyElementForm() : FORM  {
mapping SimpleField::viewComponentPart2Inputs() : HTMLBodyElement {
mapping _List::ViweElement2BodyElementList() : UL {
    ID := self.name;
    self.viewComponentParts->forEach(elm){
        log(elm.toString());
        elm.subViewComponentParts->forEach(el){
            log(el.toString());
        }
    };
}
mapping Details::ViweElement2BodyElementDetails() : FIELDSET {
    legend := object LEGEND{
        innerHTML := self.container().oclAsType(IFMLWindow).name;
    };
    inputs += self.viewComponentParts[DataBinding].subViewComponentParts[VisualizationAttribute]
    -> map visualisationAttr2bodyElm();
}
mapping VisualizationAttribute::visualisationAttr2bodyElm() : HTMLBodyElement {
    bodyElements := object DIV{
        ID  := title:= self.name;
        innerHTML := self.name + ": ";
    }
}
```

Figure 7: QVTo Code excerpt of IFML to html transformation.

Once the application has been sufficiently modeled by the IFML, the PIM model is used as an input for the transformation engine developed for the approach. The Implementation of model to model transformation is carried out using QVT language, which is proposed as a standard model transformation language by the OMG. The figure 7 shows an excerpt of the transformation engine.

The resulting files from the step of code generation are three web pages of insertion, finding, and displaying medications, accompanying with owl file representing an EMR domain ontology. The last page displays details, as a result of the click event on the list. Information sent are relative to the medication selected.

Figure 8 shows the code generated of this annotated page. Each information is annotated according to the EMR vocabulary using the RDFa serialization format. For the ontology file, the result is very long either in the ODM model or the ontology generated. Using SPARQL queries, we obtained 270 triples. So it is difficult to display all the results.

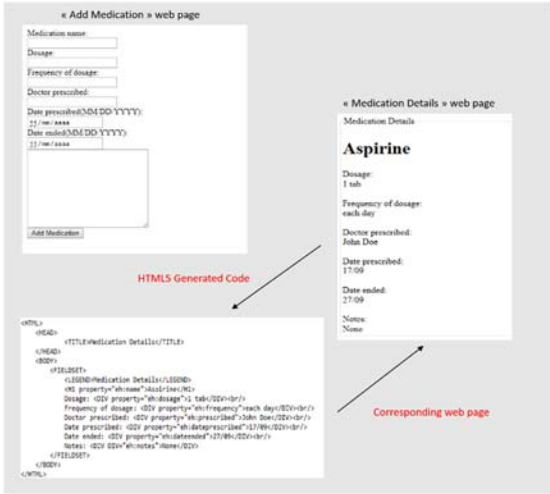

Figure 8: Annotated web pages Resulting from the code generation engine.

7 CONCLUSION

In this paper we presented an approach based on MDA for the user interfaces development of SWAs. Furthermore, we defined a Meta Model for Html5 comprising 177 meta-classes and developed the transformation engine that allows the automatic generation of the output models. These models represent an input to a Model to Text generator that give an almost complete web pages ready to be deployed, focusing on the graphical aspect of the application on one hand, and the annotations and the user event handling on the other. We generated also an owl file for the EMR vocabulary respecting RDF / XML format.

The main contribution in the proposed approach is the generation of ontologies, as well as annotated web pages from one IFML diagram, besides the abstraction of technical details. By using this models driven method, the user interfaces of semantic web applications can be easily generated without having to know all the technical specification of the execution platform. However, this can be only a particular step of MDWE process, because we restricted ourselves to one feature of SWAs which is Semantic annotations. Other, more modern features exist and must be designed using ontology-aware. Future works will cover the implementation of more refined code generator and the application of the proposed method to estimate how this approach scales in large projects. Also, we aim at enrich the proposed process and target other features of SWAs related to user interfaces.

REFERENCES

Acerbis, R., Bongio, A., Brambilla, M., Butti, S., 2015a. Model-Driven Development Based on OMG's IFML with WebRatio Web and Mobile Platform.

Acerbis, R., Bongio, A., Butti, S., Brambilla, M., 2015b. Model-driven development of cross-platform mobile applications with webratio and ifml, in: *Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems*. IEEE Press, pp. 170–171.

A\s smann, U., Zschaler, S., Wagner, G., 2006. Ontologies, meta-models, and the model-driven paradigm, in: *Ontologies for Software Engineering and Software Technology*. Springer, pp. 249–273.

Belaunde, M., Casanave, C., DSouza, D., Duddy, K., El Kaim, W., Kennedy, A., Frank, W., Frankel, D., Hauch, R., Hendryx, S., 2003. MDA Guide Version 1.0. 1.

Belghiat, A., Bourahla, M., 2012. World Acad Sci Eng Technol 6, 380–5.

Bernaschina, C., 2017. ALMOsT. js: an agile model to model and model to text transformation framework, in: *International Conference on Web Engineering*. Springer, pp. 79–97.

Bernaschina, C., Brambilla, M., Mauri, A., Umuhoza, E., 2017a. A Big Data Analysis Framework for Model-Based Web User Behavior Analytics, in: *Web Engineering, Lecture Notes in Computer Science. Presented at the International Conference on Web Engineering, Springer, Cham*, pp. 98–114.

Bernaschina, C., Comai, S., Fraternali, P., 2017b. IFMLEdit. org: model driven rapid prototyping of mobile apps, in: *Proceedings of the 4th International Conference on Mobile Software Engineering and Systems. IEEE Press*, pp. 207–208.

Berners-Lee, T., Hendler, J., Lassila, O., 2001. Scientific american 284, 28–37.

Brambilla, M., Bongio, A., Butti, S., Fraternali, P., Kling, W., Molteni, E., Seidewitz, E., 2013.

Brambilla, M., Fraternali, P., 2014. Interaction flow modeling language: Model-driven UI engineering of web and mobile apps with IFML. *Morgan Kaufmann*.

Brambilla, M., Mauri, A., Umuhoza, E., 2014. Extending the interaction flow modeling language (IFML) for model driven development of mobile applications front end, in: *International Conference on Mobile Web and Information Systems. Springer*, pp. 176–191.

Brockmans, S., Colomb, R.M., Haase, P., Kendall, E.F., Wallace, E.K., Welty, C., Xie, G.T., 2006. A model driven approach for building OWL DL and OWL full ontologies, in: *International Semantic Web Conference*. Springer, pp. 187–200.

Djurić, D., Gašević, D., Devedžić, V., 2005. Journal of Object technology 4, 109–128.

Ed-Douibi, H., Brunelière, H., 2015. Modeling software application front-ends: introducing the open source IFML graphical editor…, in: *EclipseCon North America 2015-Modeling Symposium*.

Gašević, D., Djurić, D., Devedžić, V., 2007. International Journal on Software Tools for Technology Transfer 9, 103.

Gotti, S., Mbarki, S., 2016. Toward IFVM Virtual Machine: A Model Driven IFML Interpretation.

Gotti, Z., Mbarki, S., 2016. Java Swing Modernization Approach-Complete Abstract Representation based on Static and Dynamic Analysis., in: *ICSOFT-EA.* pp. 210–219.

Hamdani, M., Butt, W.H., Anwar, M.W., Azam, F., 2018. A Systematic Literature Review on Interaction Flow Modeling Language (IFML), in: *Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences*. ACM, pp. 134–138.

Heinz, M., Lämmel, R., Varanovich, A., 2017. MODELSWARD 17, 478–486.

Huang, A., Pan, M., Zhang, T., Li, X., 2018. Static Extraction of IFML Models for Android Apps, in: *Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, MODELS '18*. ACM, New York, NY, USA, pp. 53–54.

Klašnja-Milićević, A., Vesin, B., Ivanović, M., Budimac, Z., Jain, L.C., 2017. Semantic Web, in: *E-Learning Systems*. Springer, pp. 115–148.

Laaz, N., Mbarki, S., 2016a. Integrating IFML models and owl ontologies to derive UIs web-Apps, in: *Information Technology for Organizations Development (IT4OD), 2016 International Conference On*. IEEE, pp. 1–6.

Laaz, N., Mbarki, S., 2016b. Combining Ontologies and IFML Models Regarding the GUIs of Rich Internet Applications, in: *International Conference on Artificial Intelligence: Methodology, Systems, and Applications. Springer*, pp. 226–236.

Laaz, N., Wakil, K., Mbarki, S., Jawawi, D.N.A., Laaz, N., Wakil, K., Mbarki, S., Jawawi, D.N.A., 2018. *Journal of Computer Science 14*, 1267–1278.

Morgan, R., Grossmann, G., Schrefl, M., Stumptner, M., Payne, T., 2018. VizDSL: A Visual DSL for Interactive Information Visualization, in: *International Conference on Advanced Information Systems Engineering*. Springer, pp. 440–455.

Musumbu, K., 2013. Towards a Model Driven Semantics Web using the Ontology, in: *The 2013 International Conference on Advanced ICT for Business and Management (ICAICTBM2013)*. p. 700.

OMG, Q., 2011. Meta Object Facility 2.0, Query/View/Transformation Specification.

Partridge, C., de Cesare, S., Mitchell, A., León, A., Gailly, F., Khan, M., 2018. ONTOLOGY THEN AGENTOLOGY: A finer grained framework for enterprise modelling, in: *6th International Conference on Model-Driven Engineering and Software Development*. SCITEPRESS-Science and Technology Publications.

PMR - Droid [WWW Document], n.d. URL http://www.cse.msu.edu/~cse435/Projects/F09/PMR-droid/web/docs.html (accessed 10.22.18).

Queiroz, R., Marques, A.B., Lopes, A., Oliveira, E., Conte, T., 2018. Evaluating Usability of IFML Models: How Usability is Perceived and Propagated, in: *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems, IHC 2018*. ACM, New York, NY, USA, pp. 21:1–21:10.

Rodriguez-Echeverria, R., Preciado, J.C., Sierra, J., Conejero, J.M., Sanchez-Figueroa, F., 2018. *Science of Computer Programming*.

Roubi, S., Erramdani, M., Mbarki, S., 2016. Extending graphical part of the Interaction Flow Modeling Language to Generate Rich Internet Graphical User Interfaces, in: *2016 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD). Presented at the 2016 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, pp. 161–167.

Saripalle, R.K., Demurjian, S.A., De la Rosa Algarín, A., Blechner, M., 2013. *International Journal on Semantic Web and Information Systems (IJSWIS) 9*, 62–97.

Umuhoza, E., Brambilla, M., Cabot, J., Bongio, A., 2015. Automatic code generation for cross-platform, multi-device mobile apps: Some reflections from an industrial experience, in: *Proceedings of the 3rd International Workshop on Mobile Development Lifecycle. ACM*, pp. 37–44.

Wakil, D.K., Isa, M.A., 2015. Analyzing Modern Web Applications to Recognize Features-based Web Engineering Methods, in: *KSII The 7th International Conference on Internet (ICONI)*.

Wakil, K., N.A., D., 2017. International Journal of Advanced Computer Science and Applications 8.

Yigitbas, E., Mohrmann, B., Sauer, S., 2015. LMIS@ EICS 2015, 42–46.

Zedlitz, J., Jörke, J., Luttenberger, N., 2012. From UML to OWL 2, in: *Knowledge Technology*. Springer, pp. 154–163.