The Workshops of the Tenth International AAAI Conference on Web and Social Media
Social Web for Environmental and Ecological Monitoring: Technical Report WS-16-20

Model Driven Development of Social Media Applications for Environmental Monitoring

Marco Brambilla, Andrea Mauri, Eric Umuhoza
Politecnico di Milano. Dipartimento di Elettronica, Informazione e Bioingegneria
Piazza L. Da Vinci 32. I-20133 Milan, Italy
{marco.brambilla, andrea.mauri, eric.umuhoza}@polimi.it

Abstract

The growth of social media and the resulting increase of user generated content attracted people coming from different fields interested in analyzing this kind of data. Recently user generated content has been used for environmental monitoring. In this paper, we propose a model-driven approach for developing social media applications for environmental monitoring that will provide a simplified solution for integrating various data sources, exploiting different platforms and allowing complex data analysis to be performed.

Introduction

The growth of social media (Doherty and Smeaton 2010) and the resulting increase of user generated content attracted people coming from different fields interested in analyzing this kind of data. Recently, user generated content has been used for environmental monitoring (Leeuw 2014; Fedorov, Fraternali, and Tagliasacchi 2014; Fedorov et al. 2013).

When developing applications using these sources, developers face the issues of understanding the specific API of each platform and its data type.

The objective of this work is to define a model-driven approach for the development of social media environment monitoring applications, providing a simplified solution for integrating various data sources, exploiting different platforms and allowing complex data analysis and visualization to be performed. Model-driven development is a development paradigm that uses models as the primary artifact of the development process. This allows abstraction from specific implementation, that improves portability of software to new/changing technologies and the interoperability between different platforms. When applying this paradigm, the application code is automatically generated from those abstract models. The main contribution is an extension of OMG's standard IFML (Interaction Flow Modeling Language) with concepts covering the different data sources (sensors and social network) and data analysis. These concepts are introduced using the IMFL extension mechanism, in order to be fully integrated in the IFML ecosystem.

Interaction Flow Modeling Language (IFML)

The Interaction Flow Modeling Language[1] supports the platform independent description of graphical user interfaces for applications accessed or deployed on such systems as desktop computers, laptop computers, mobile phones, and tablets. An IFML model supports the following design perspectives: (1) The *view structure specification*, which consists of the definition of **view containers**, their nesting relationships, their visibility, and their reachability; (2) The *view content specification*, which consists of the definition of **view components**, i.e., content and data entry elements contained within view containers; (3) The *events specification*, which consists of the definition of **events** that may affect the state of the user interface. Events can be produced by the user's interaction, by the application, or by an external system; (4) The *event transition specification*; (5) The *parameter binding specification*, which consists of the definition of the **input-output dependencies** between view components and between view components and actions; and (6) The reference to **actions** triggered by the user's events. IFML can be easily extended, since it uses the extensibility mechanisms of UML to allow the definition of stereotypes, tagged values and constraints. Finally, an IFML model can be used for the automatic generation of an application (Umuhoza et al. 2015). For instance, the tool Webratio[2] allows the automatic generation of both mobile and web applications from an IFML model.

Monitoring the Environment

In order to define the components to extend IFML we need first to understand the context we are going to address, in term of data sources and types involved. The data sources can be of two types: ad hoc sensors or user devices. The former comprehend sensors that measure environment parameter such as temperature, water level, humidity, atmospheric pressure and so on. The latter comprehend smartphones, that nowadays are equipped with a vast range of sensors, that already have been proved useful for environmental monitoring (Leeuw 2014). Furthermore people usually use smartphones to share their activities on social media, that nowadays became another rich source of data for environmen-

[1] www.ifml.org
[2] http://www.webratio.com/

tal monitoring (Fedorov, Fraternali, and Tagliasacchi 2014; Fedorov et al. 2013).

Social Media Environment Monitoring with IFML

In this section we describe the IFML extension for modeling social media applications for environmental monitoring. In particular, our aim is to cover two types of applications: (i) Sensor applications: i.e. mobile applications that use the user's smartphone to gather information regarding the environment using either the sensors on the device or IoT sensors. This type of application can also have interaction with social networks; and (ii) Analysis applications: i.e. web applications that gather information from different sources and allow the user to perform complex analysis. Hence we propose three classes of new components: social, IoT interaction and complex visualization.

Social Components

This category regroups the components which encapsulate the logic of the interaction with the social platforms. For each social platform we provide a specific action class that extends *Action Class* of IFML standard. These extensions hide the complexity of their APIs from the developer and thus reduce the cost of designing new applications.

IOT Interaction Components

The use of Internet of Things can be worthwhile in environmental monitoring if combined with the actual engagement of users in social networks. Our approach sees mobile devices as IoT enablers for environmental monitoring since they allow the interaction with the surrounding environment (other mobile devices, sensors, etc.) through networks available in those devices. The IFML standard and its mobile extension (Brambilla, Mauri, and Umuhoza 2014) do not cover the aspects related to inter-device communication. We propose extensions that allow the modeling of the communication of mobile device and the environmental sensors. Those extensions include Environmental Sensor Event, an extension of *SystemEvent* of IFML, which allows the modeling of the environmental sensors events; and the extensions of *SimpleContextVariable* class of IFML. For each environmental sensor, we defined a context variable which allows the capturing of the readings of the corresponding sensor.

Complex Visualization Components

The ultimate goal of environmental monitoring systems is to provide value to people. Such value can be perceived through appropriate user interfaces, which visualize information (through reports, or info-graphics), and let users navigate the information. We define the IFML extensions allowing the modeling of such rich visualization elements. This category includes Table, Chart and Map which are defined as extensions of *ViewCompent* class of IFML.

Related Work

Several papers have modeled data analysis architectures, especially taking into the account contribution from people. For instance, BPEL4People (Kloppmann et al. 2005) and WS Human Task (Ings and others 2010) tried to include human tasks in normal business processes. Furthermore, with the similar objective of simplifying the development of applications, (Schall, Satzger, and Psaier 2014) and (Tranquillini et al. 2015) both described a hybrid framework in which data-intensive application can be designed on top of BPMN, where tasks can be either executed by machines or humans. Finally, similar to our work, (Brambilla and Mauri 2012) proposed an extension of WebML in order to incorporate the social features.

Conclusion and Future Work

In this work we proposed an extension of the IFML language for developing an application that needs to exploit social media and sensors for environmental monitoring. Future work will aim to refine these components and implement them in order to be able to perform a suitable validation.

References

Brambilla, M., and Mauri, A. 2012. Model-driven development of social network enabled applications with webml and social primitives. In *MDWE*. Springer. 41–55.

Brambilla, M.; Mauri, A.; and Umuhoza, E. 2014. Extending the interaction flow modeling language (ifml) for model driven development of mobile applications front end. In *MobiWIS*. 176–191.

Doherty, A. R., and Smeaton, A. F. 2010. Automatically augmenting lifelog events using pervasively generated content from millions of people. *Sensors* 10(3):1423.

Fedorov, R.; Martinenghi, D.; Tagliasacchi, M.; and Castelletti, A. 2013. Exploiting user generated content for mountain peak detection. In *SoHuman*, 21–28.

Fedorov, R.; Fraternali, P.; and Tagliasacchi, M. 2014. Snow phenomena modeling through online public media. In *Image Processing (ICIP), 2014*, 2174–2176. IEEE.

Ings, D., et al. 2010. Web services–human task (ws-humantask) specification version 1.1. *OASIS Committee Specification (August 2010)*.

Kloppmann, M.; Koenig, D.; Leymann, F.; Pfau, G.; Rickayzen, A.; von Riegen, C.; Schmidt, P.; and Trickovic, I. 2005. Ws-bpel extension for people–bpel4people. *Joint white paper, IBM and SAP* 183:184.

Leeuw, T. 2014. Crowdsourcing water quality data using the iphone camera. *Electronic Theses and Dissertations*.

Schall, D.; Satzger, B.; and Psaier, H. 2014. Crowdsourcing tasks to social networks in bpel4people. *WWW* 17(1):1–32.

Tranquillini, S.; Daniel, F.; Kucherbaev, P.; and Casati, F. 2015. Modeling, enacting, and integrating custom crowdsourcing processes. *ACM Trans. Web* 9(2):7:1–7:43.

Umuhoza, E.; Ed-douibi, H.; Brambilla, M.; Cabot, J.; and Bongio, A. 2015. Automatic code generation for cross-platform, multi-device mobile apps: Some reflections from an industrial experience. MobileDeLi 2015, 37–44.