IHDCB339: Reading Notes

Kenny Warszawski
kenny.warszawski@student.unamur.be

University of Namur — January 20, 2019

1 Article Reference

Lam, An and Haugen, Øystein "Complex Event Processing in ThingML", pp. 20-35, 2016.

2 Overview

The article is a presentation of a cross platform modelling language called *ThingML* which is dedicated to Internet of Things. More precisely, the article focus on an extension of this language for Complex Event Processing (CEP).

2.1 General problem

Nowadays, Cyber Physical Systems (CPS) are increasingly present and need to have extremely fast responses. CEP is an answer to this constraint. ThingML with CEP extension helps the implementation of such complex system by adding an abstraction on the programming language used for deployment. That way, a model described in ThingML has the same behaviour even if it is deployed in Java or C/C++. This article treats the capabilities of ThingML with CEP extension to integrate a CEP system and deal with CPS constraints.

3 Concepts used in the article

- **Complex Event Processing (CEP):** Complex Event Processing (CEP) is a set of methods and techniques for tracking and analyzing real-time streams of information and detecting patterns or correlations of unrelated data (complex events) that are of interest to a particular business [4]
- **Internet of Things (IoT):** The Internet of Things(IoT) is a novel paradigm that is rapidly gaining ground in the scenario of modern wireless telecommunications. The basic idea of this concept is the pervasive presence around us of a variety of things or objects – such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, etc. – which,through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals [1]
- **Cyber Physical Systems (CPS):** Cyber-Physical Systems (CPS) are integrations of computation and physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa [2]
- **Event Pattern Language (EPL):** Event Pattern Language (EPL) is the language to define atomic or complex event and specify the process of filtering (determine event of interest) and extracting events properties for constructing high-level events [3]
- **RFID:** Radio Frequency Identification (RFID) is a wireless technology capable of automatic and unambiguous identification without line of sight by extracting a unique identifier from microelectronic tags attached to objects. [5]

4 Article Summary

4.1 Thematic

ThingML is a domain specific language and compiler for the internet of things. This DSL is used to describe software components and communication protocols without being coupled to a specific programming language. To this end, this technology provides the possibility for developers to deploy the same implementation onto different platforms like Java, JavaScript, C/C++ and Arduino by only a code generation from a description file.

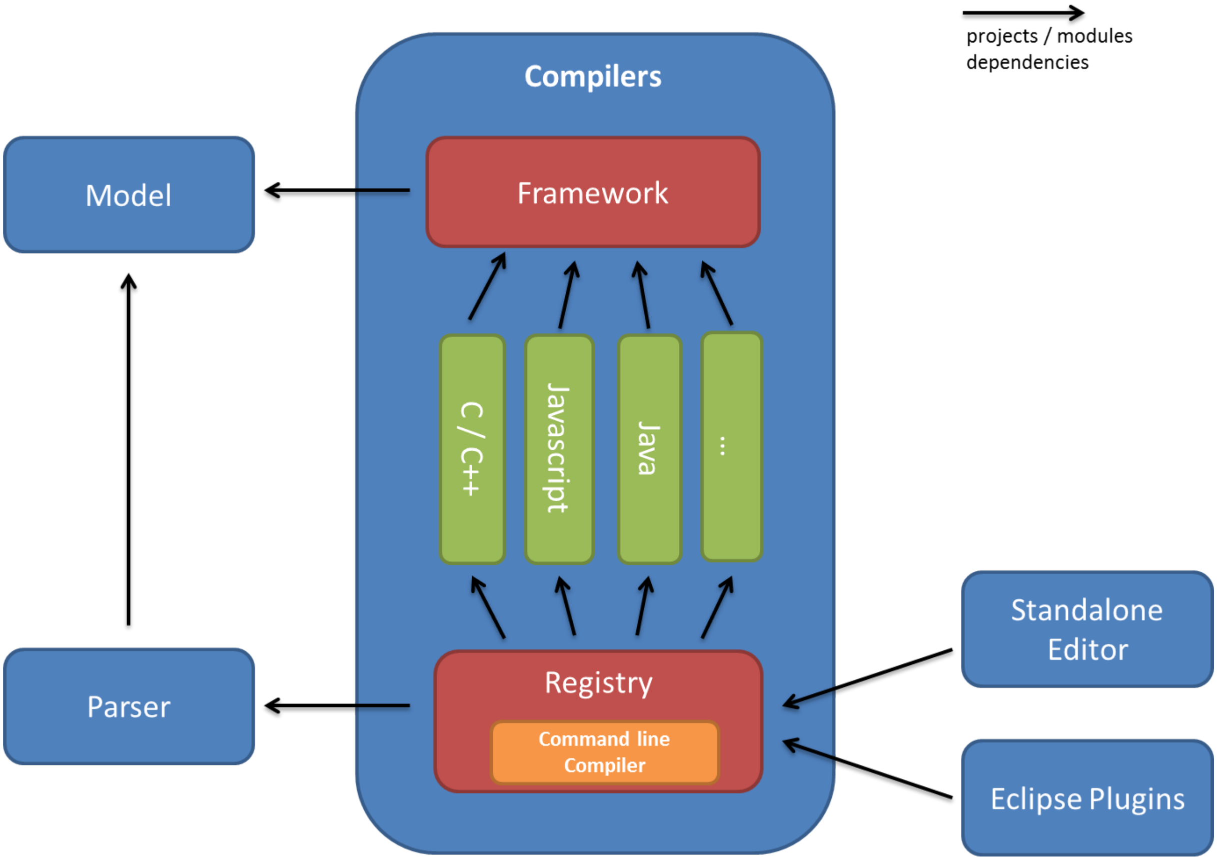

Figure 1: ThingML overview

ThingML is used to describe state charts and components for an Internet of Things environment. The communication between models can be based on various asynchronous messaging protocols. ThingML is based on a modular and open-source platform. Therefore, there is an opportunity to contribute on the development and by the way adding extensions to the language.

4.2 ThingML CEP Extension capabilities for Complex Event Processing

As already mentioned, ThingML can be extended with various plug-in. The one which is focused in the article is the CEP extension. The article wrote by Lam An and Haugen Øystein defines the capability of ThingML to integrate a Complex Event Processing environment by answering to two questions.

4.2.1 Is ThingML CEP Extension an efficient language for developing CEP applications ?

The article answers to this question by evaluating the language expressiveness on two characteristics: the lines of codes and the number of keywords. The authors wrote different CEP operators using ThingML with and without CEP extension. The result is that there is less lines of codes and number of keywords by using the CEP extension but there isn't significant differences. However, this extension can be very useful in a large scale to save time by reducing those characteristics and in some cases a code duplication can be avoided. In a disjunction query for example, the non-usage of the CEP extension leads to check the occurrences of each event. This kind of situation conducts generally to code duplication and errors.

4.2.2 Is ThingML CEP Extension powerful enough for CPS systems ?

The article answers to this question by comparing ThingML CEP Extension with Esper which is an open source CEP Engine. The experiment done by the authors was base on the same workstation for all tests. They compared both technologies on Latency, CPU Utilization and Memory Utilization. The statistics show that ThingML with CEP extension uses less CPU and Memory than Esper. The CPU used by ThingML is around 50-60% and 75-90% for Esper. The Memory Usage is around 35-90MB for ThingML and 50-120MB for Esper. Nevertheless, the Esper latency is around 2 times faster than ThingML with CEP extension.

5 Conclusions

In summary, ThingML CEP extension revealed to ease the creation of CEP applications than using ThingML without the CEP extension. The study also revealed that ThingML can be used with limited resources and shows reasonable latency for processing events. There is a need to evaluate ThingML with more complex systems to have more relevant informations. In fact, the authors tested those Frameworks with basic operations. They also tested it on a workstation with generous resources. It can be relevant to test ThingML on a Arduino or another physical limited workstation.

List of Figures

References

[1] D. Giusto et al. *The Internet of Things*. Springer, 2010.

[2] Edward A. Lee. “Cyber Physical Systems: Design Challenges”. In: (Jan. 2008).

[3] D. Robins. “Second International Workshop on Education Technology and Computer Science”. In: 2010.

[4] I. Schmerken. *Deciphering the myths around complex event processing*. Wall Street & Technology, May 2008.

[5] Wu Y et al. “RFID enabled traceability networks: A survey”. In: (2011).