Curriculum as an event hierarchy model

J. Lang, V. Krajčovič
Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology
Ilkovičova 3, 84216 Bratislava 4, Slovakia
lang@fiit.stuba.sk, krajov@gmail.com

***Abstract*—Activity plays an important role in human life. Psychology as scientific discipline knows several categories of activities, motivations to activity, their characteristics etc. A significant group of these activities expects some cognitions, knowledge and skills. We identify particular relations, arrangement respectively event hierarchy, which is represented by expression of requested knowledge. Existing knowledge is not sufficient for execution certain activities. Therefore it is necessary to interact with the new content - holder of such knowledge. This paper provides an alternative way of identifying of content curriculum by analogy to the technology of Complex Event Processing and Event Driven Architecture. Identified content is a key in relation to national-education program creating as well as school-education program. Proposed method is implemented by modelling tool EDUMO.**

I. Introduction

There are already many different systems to support learning in the form of Learning Management Systems LMS, Content Management Systems and the like. Each has proposed ways to address curriculum content. However, none of them solve the problems of redundancy, reusability and maintainability of content curriculum. One of the objectives of our research was to verify the customary procedure (software modeling) of similar processes in other application domains and thus to work towards an outcome that could be described as the minimum required content of the curriculum model for achieving a certain level of education. Analogy of concept of Complex Event Processing CEP, Complex Event Hierarchies with events of normal human life (implementation of professional skills, the key competencies etc.) was the motivation to the way how to create models of desirable content curriculum. CEP as a relatively newly established technology for creating and managing information systems, including [4], monitoring business activities, management of business processes, integrate business applications, event-driven architecture, network security and business level, in compliance with real-time control and the policies used techniques for detection of complex patterns events in the event cloud processing streams of events, event correlation and abstraction, event hierarchies, and relations between events.

II. Analogy between CEP and CMS event concept

During the process of learning a variety of events occur that affect its course and level of knowledge of the learner. Discrete events in certain combinations represent a composite - a complex event. For example, information security (treatment against the intrusion of SQL injection), GUI implementation with a box for username and password, editing the general conditions of use web site services, functionality for registration in the case that we haven't login data (we are not registered) may represent a decomposition of complex event. The complex event will be a requirement to implement the authentication form web site. Analogous to the approach applied in the EDA and CEP we can finalize the hierarchical structures in the area of the curriculum. Such processing of content will provide useful solutions - event handlers - useful throughout the life of a man. Learning process assumes a hierarchy in which the content of complex events can be broken down into elementary events. Complex event aggregates elementary events and present higher logical meaning than elementary events. Event is usually represented by an object, containing information on activities carried out in the system [4].

Event may be linked to other events. Event consists of 3 aspects:

- Form - Form event is an object which may contain attributes or data. Form may be something simple like a string or a complex tuple data. This data may include data such as: the implementation period of activity, place of origin, source of an event, a description of importance and dependencies.
- Significance – Event represents an activity. This activity is called the importance of the event. Event usually contains data that describe the importance of activities.
- Relativity - The activities are linked to other activities through time, causality or aggregation.

Example of a class of events may look like this:

```
Class InputEvent
  {
    Name NewOrder;
    Event_Id E_ID;
    Customer Id;
    OrderNo O_Id;
    Order (CD X, Book …);
    Time T;
    Causality (Id1, Id2, …);
  }
```

Relationships which may arise between events are as follows [4]:

- Time relationship - defines the ordering of events in time. For example, event A occurred before event B.
- Causal relationship - to determine that event was triggered by another event. For example, event A triggered event B.
- Aggregation – if event A was triggered by more than one event e.g. B1, B2, Bn .. then A is referred to as the aggregated event and B1, B2, Bn are members of the event A. Event A is also called complex event.

An example of event hierarchy is graphically illustrated in Figure 1. Figure includes four events: Message sent, Message received, Acknowledgment and Message delivered.

The first three are causally dependent on each other. So the event Message sent binds report Message received, which then runs the adoption event Confirmed. These three events together are aggregated by the event Message is delivered. In CEP systems the event Message is received (complex event) is represented as a rule. Rule compliance checks three input conditions.

The analogy itself is a powerful means to solve problems, and based on the mimic. Software engineering was influenced by analogy to the architecture of buildings (see the work of Christopher Alexander). Later Clemens et al. used the software architecture analogy to a living

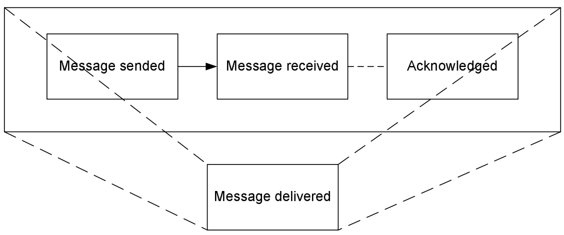

Figure 1. Event hierarchy of part a network communication

organism [1]. And so the analogy has become in our case means for identifying the minimum necessary content.

III. Proof of Concept

For verification objectives we have designed a software tool EDUMO. With this tool, we suggested a few exemplary models. Figure 2 is a part of a model that captures complex event - function definition in certain programming language. The hierarchy consists of four events. There are events: Function without return value, Function with return value and Function with parameters. These are events at the lowest level. Functions with no return value and with return value are independent of each other. Function with parameters is causally dependent on both of them. While the student learns to create functions with appropriate parameters is important to learn how to create a function without parameters. These three functions are aggregated by the event Function definition. So if a student manages handling of the three events from the lower level he actually controls all types of function definitions. Aggregated events in the proposed software tool EDUMO are defined in declarative manner.

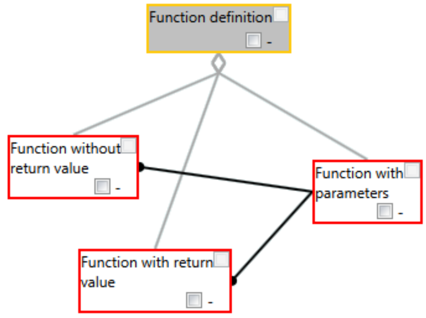

Figure 2. Function definition example

Let's analyze the example of an aggregate event of network communication environment as shown in Figure 1. There are three elementary events: Message sended, Message received and Acknowledged. These three events are aggregated by the event Message delivered, and this event is of higher logical importance.

A. *Complex event example*

A more complex example of events hierarchies is listed for the PHP language course at beginner level. The events in the hierarchy are drawn from the pages of W3C [6]. The whole hierarchy is divided into 3 levels, as shown in Figure 3. Event marked with the capital letter A is different from others because it is a prerequisite for the proposed event-whole hierarchy. It is assumed that a student who wants to undergo the PHP course passed the course of HTML. Complex event marked by the letter B represents the aim of the model. The complex event is a composite of low-level events. This event has already associated the learning content. This fact is indicated by the checkbox marked with the letter D. Point C marks an event that is optional. This means that for successful course is not necessary to control the content of that optional event.

In this event hierarchy can be traced similar features, which are also found in other programming languages. Events marked with a blue circle are events that are general in nature for most programming languages. Processing of such minimum event hierarchies makes "Event API" for different areas of learning. Such "Event API" could create something like a framework for creating curriculum.

B. *Collaboration*

In addition to the use of analogy to create curriculum we use to create the minimum necessary content and collaborative processing of identified events. Collaboration consists of two steps. The first step is manual. Created model contains all the events necessary to achieve the intended purpose. Identifying these events there is a cloud of events. To create a curriculum is needed at least two domain experts. Those following the identification of event Cloud create models of education according to their own ideas. This brings us several subjective models proposals. We assume that will not match. The second collaboration step is automatic. This step is based on an algorithm that performs penetration events and relationships of the available models. The result is the content which is acceptable to all involved.

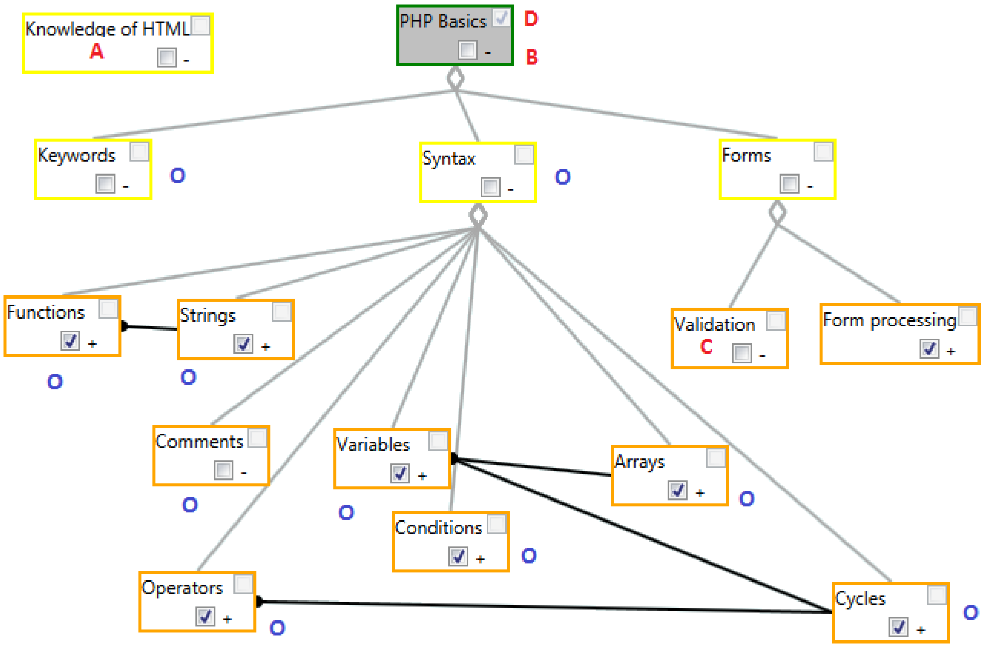

Figure 3. Learning model for PHP Beginer Course

Graphical representation of the collaboration using penetration of individual sets is shown in Figure 4.

As a result we get a model that exists in three different forms - graphical expression of models, XML file, in which the model is saved and the SCORM format

IV. Objectives

To check the proposed method, we set the following objectives:

- Evaluation of the proposed models specification
- Proposed models of the content
- Export to SCORM testing and compatibility with MOODLE LMS

During creating models of training, we tested the mutual compatibility of the proposed methods and modeling tools EDUMO.

A. Proposed models specification evaluation

In order to assess the accuracy of design models of training, we chose we chose to rewrite the content in language Z. The language Z is a mathematical language for formal mathematical description of requirements. In the Z language are written mostly aggregated operations, which are carried out during models design time. In the Z language definition of requirements we caught main operations that are carried out in the draft of models. These are simple operations like add your event, delete event etc. There are also more complex operations as an expression of the resulting events and the relations of the resulting collaborative model. To verify the accuracy of notation Z/EVES tool was used and all tests took place correctly. The result of evaluation is presented in Figure 5. Expression of the resulting events in the language Z is following:

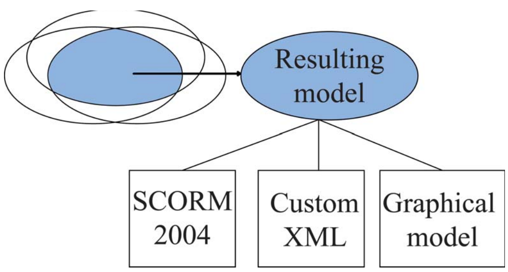

Figure 4. Resulting model of curriculum

```
\begin{schema}{GetIntersectedEvents}
    \Delta ModelEvents \\
    \Delta ResultModelEvents\\
    events1? : \power Events \\
    events2? : \power Events \\
    events3? : \power Events
\where
    resultEvents' = events1? \cap
    events2? \cap events3?
\end{schema}
```

B. Experimental models

During the investigation of the creation of minimum required education models we have developed several experimental models. These models were made collaboratively. When creating experimental models, we confirmed the hypothesis that people from the same area do not create the same model. Therefore, we consider it appropriate to use common features of available models. During designing the models borderline situations may occur. For example, all models will be completely identical. This is the ideal situation, but very unlikely. This case did not occur in our tests. It may also happen that the models are completely disjoint. If this occurs, it will be appropriate to examine the level of understanding of the intended objective. This case in our tests also did not occur.

C. *Export compatibility with MOODLE learning management system*

To integrate with existing systems, we opted to export

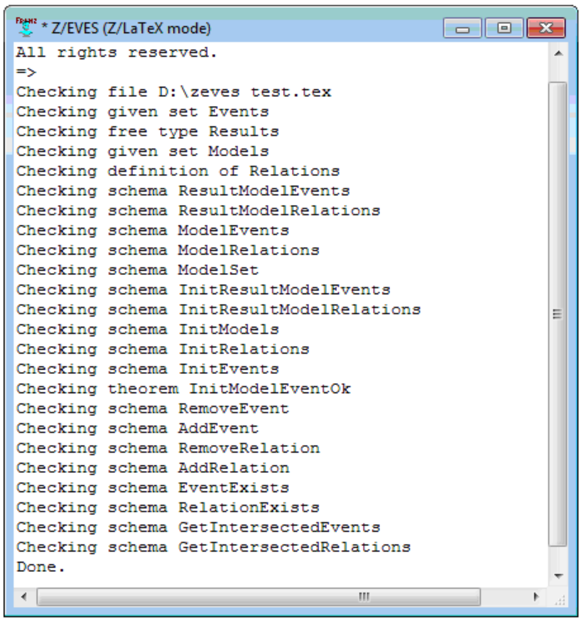

Figure 5. Requirement evaluating with Z/EVES

the model to education world-wide used SCORM format. Export to this format, we tested with the system MOODLE [5]. The course does not support testing and verification of knowledge yet. In the form in which it appears now, it serves just as a resource for learning. EDUMO tool can generate HTML documents containing the content bounded by final model.

V. Discussion

The result of this work is a method and tool for creating models of educational content. This describes the events necessary to achieve a certain degree of academic or professional education. When designing the method we took into consideration the ever-growing number of information which may include individual courses, and therefore we propose a method which could produce the smallest possible model. We have confirmed the hypothesis. People of the same environment created different models for the same goal. In proposed approach of minimizing the content of the curriculum is found a few problems that make them up:

- The problem of ensuring a unique identifier for an event
- The problem with the events generated outside the main hierarchy
- The problem of determining the necessary content
- Problem with model representation

The curricula and syllabuses are only in text form. Created models of education across text form have the following advantages:

- They are transparent – they allow maintenance of large content
- You can perform update without the duplication of parts
- Faster retrieval of information contained in the curriculum
- Easier to transform the model into another format
- Portability between systems
- Easier processes automation

VI. Future Work

For the creation of courses in the LMS-based content would be interesting to finalize the navigation in hierarchy for purposes of knowledge measuring and understanding of the hierarchy. User interface EDUMO is created in such a form that meets the necessary functionality and not always be user-friendly. Some actions should be resolved by mouse movements such as induction of forms and the like. Another interesting possibility could be the extension of LMS system that can process models created in their native form. Thus, a system that would create the management of established models.

VII. Conclusions

The work is based on the analogy between the CEP technology concept, the concept of EDA architecture and the concept of events for identification of curriculum content. Its minimal form, we have proposed to ensure by the collaborative approach. Given the fact that the models were neither identical nor disjoint, the theoretical assumptions of the minimum curriculum content form were confirmed in an embedded modeling tool EDUMO. The proposed models were transformed into formal mathematical description for the purpose of verification. To verify the accuracy of notation Z/EVES tool was used and all tests took place correctly. Finally we would like to make a recommendation for the specifications of the new LMS systems requirements. It would by very interesting to incorporate the support for modeling the curriculum content.

Acknowledgment

The work was supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KEGA 345-032STU-4/2010.

References

[1] Clemens P. a kol.: Documenting Software Architectures. Views and Beyond 2nd Ed. 2010 Addison-Wesley. ISBN-13 978-0201703726

[2] Hrúz, B., Ondráš, J., Flochová, J.: Discrete event systems - an approach to education. Istanbul: The 4th Symposium on Advances in Control Education, July 1997. p. 14-16.

[3] Lang J., Kákody J.: Inovácia obsahu vzdelávania vzdelávacieho programu objektovo-orientovanou didaktickou transformáciou. Trnava: AlumniPress, 2009. SCHOLA 2009. p. 256-261. ISBN 978-80-8096-106-0.

[4] Luckham D.: The power of events. Boston : Addison-Wesley, 2007. ISBN 0201727897.

[5] MOODLE: [Online] [Date December, 5, 2010] http://docs.moodle.org/en/About_Moodle.

[6] W3C. Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C. [Online] [Date November, 26, 2010.] http://www.w3.org/TR/REC-xml/.