A Case Study on Modeling of Complex Event Processing in Enterprise Architecture

Hyeonsook Kim and Samia Oussena
School of Computing and Technology, University of West London, St Mary's Road, Ealing, U.K.

Keywords: Enterprise Modeling, Enterprise Architecture, Complex Event Processing, Event Driven Architecture, Model Driven Architecture, Business Modeling.

Abstract: Over the decades, Enterprise Architecture (EA) has been researched to supply all the necessary components for enterprise system modelling including taxonomies, meta-models, architecture development methods, and modelling tools. The main benefits of EA are the knowledge infrastructure for analysis and reporting by all stakeholders and the possibility of designing new conditions in an organized manner. However, EA now faces a big challenge with the growing dynamic of market demands and the rapid changes in business environments, which requires agile system response and self-evolutionary behaviour to support quick decision-making. In technology side, there are already matured, promising paradigm to tackle this challenges, which is Complex Event Processing, however it has not been fully dealt with EA. No standardized method for applying event driven approach in business and IT systems modelling has been developed yet. The paper investigates a possibility of integration of EDA by adding event process layer between business operating and business process layers in the EA stack. Complex event patterns are identified and an event meta-model extending ArchiMate is also proposed to integrate complex event modelling to the business modelling. Using a case study, we modeled a business scenario with event driven approach.

1 INTRODUCTION

Over the years, model-based development has gained rapidly increasing popularity across various engineering disciplines. Numerous efforts have resulted in the invention of concepts, languages, and tools for the definition, analysis, transformation, and extension of domain-specific modeling languages as well as general purpose modeling language standards. For enterprise systems, Enterprise Architecture (EA) has been researched to supply all the necessary components for enterprise system modeling including taxonomies, meta-models, architecture development methods, and modeling tools. Main benefits of EA are the knowledge infrastructure for reporting and analysis by all stakeholders and the possibility of designing new conditions in an organized manner (Lankhorst, 2004). EA is not only an instrument for strategic planning of IS/IT but also other business functions, such as compliance control, continuity planning and risk management.

EA now faces a big challenge with the growing dynamic of market demands and the rapid changes in business environments, which requires agile system response and self-evolutionary behaviour by quick decision making (Kim et al., 2006). EA should provide the requirements in architectural level with support of precise modelling method, language, patterns or frameworks.

Several different styles of architecture are possible. A Service Oriented Architecture (SOA) involves the publication of logically coherent groups of business functionality as interfaces that can be used by components using synchronous or asynchronous messaging. An alternative style, argued as reducing coupling between components and thereby increasing the scope for component reuse, is Event Driven Architecture (EDA) that promotes the production, detection, and consumption of events (Engels, 2008).

An important difference between SOA and EDA is that the latter generally provides scope for Complex Event Processing (CEP) where the business processes within a component are triggered

Kim H. and Oussena S..
A Case Study on Modeling of Complex Event Processing in Enterprise Architecture.
DOI: 10.5220/0003968701730180
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 173-180
ISBN: 978-989-8565-12-9

by multiple, possibly temporally related, events. In SOA there is no notion of relating the invocation of a single business process to a condition holding between the data passed to a collection of calls on one of the component's interfaces. Whilst a complex event based approach to architectural design must take efficiency concerns into account, the primary concern is how to capture, represent and analyze architectural information as an enterprise design. This is especially useful in cases where it is had or sometimes impossible to design business behaviour in predefined sequence of activities, as most environments are event driven and non-deterministic. In multinational organisation, a single dimension top town analysis of business activities is unlikely to work due to the complexity of it environment. It is also even harder for a start-up company to identify and analyse services or business activities that are necessary to implement it business.

Furthermore, services used from an application or orchestration manager expose their interface explicitly and will require changes to any services that bind to them. Consequently, the services are tightly linked, providing less agility and dynamic. Business units such as component or service consuming or producing event are by nature more decoupled, providing the flexibility necessary to adapt to changing circumstances.

Several efforts have been made to integrate SOA to EA. For example, OMG published a language specification called SoaML for SOA based business modeling. It supports service modeling at business level, integrating the modelled services with business processes at IS/IT level by service orchestration or choreography (Casanave, 2009). A leading EA industry consortium, the Open Group, has also published their effort on SOA driven enterprise modeling, demonstrating fitness of their TOGAF framework to service oriented modeling in (The Open Group, 2011).

Very little work has been done to fully apply or consider EDA in EA. In our view, the importance of events has been overlooked and there is no appropriate or standardized way to model business and IT systems with EDA approach in a consistent way. Our contribution in this paper is the proposal for integrating Complex event modelling to EA modelling.

In addition to the architectural support, event driven thinking shows a more straightforward solution to business modelling. For example, events can be easily identified with specific business goals, policies and constraints that they related to. For that reason, the business process can be defined with the focus on 'what' need to be done with the relevant events rather than imposing the details of 'how'. Our approach has been inspired by VPEC-T, an approach that applies event-driven thinking to enterprise modeling by analyzing business with five core concepts including event (Green and Bate, 2007). In this paper, we illustrate our approach by a case study, where complex event modelling is integrated to business operating model.

The remainder of the paper is structured as follows: Section 2 introduces our approach on event driven enterprise architecture modeling. Section 3 presents the case study. Section 4 concludes and gives an outlook to future work.

2 OUR APPROACH: EVENT-DRIVEN EA MODELING

Enterprise Architecture encompasses modelling both business architecture and IT architecture, bridging the gap between them. In this work we argue that complex event modelling needs to be integrated not only to IT systems but also to business modelling. This will provide the consistency between EA models.

It is essential for business modelling to capture business needs rather than putting constraints due to the technology. By adopting event driven modelling in EA, we can identify and model events at conceptual level without concern of the technical architecture – how to recognize the events. The model is then gradually detailed and comprehensively refined in the logical and physical level with added constraints for implementation.

Events in this approach are what trigger a chain of actions in business, for instance, serving its customers, collecting its income, managing its staff and generally meeting its obligations. In addition to triggering actions, it also includes notification of changes that leads to check of constraints of business goal. In (Clark et al., 2011), authors have recognized the importance of events at analyzing business goals. The component of UML has been extended with event concept and event patterns (templates). However, its application only targets simulation of compliance and consistency checking.

A number of specialised modelling notations have been proposed for EA modelling. In most cases these notations provide a number of views or layers that capture the enterprise from different perspectives. These layers provide a good conceptual fit to the problem of representing EA domain elements and their relationship. Although variation of the EA stack have been proposed, fundamentally common pattern and targets remain the same (Banger et al., 2008). Figure 1 illustrates the change that we are proposing to the EA stack, i.e. adding a new layer to deal with CEP.

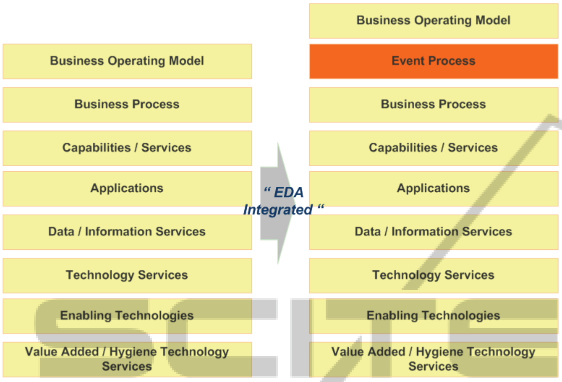

Figure 1: EA Stack before and after EDA integrated.

The Business Operating Model (BOM) layer highlights both how the organization is structured, and how it interacts to achieve its mission, goals and objectives. The Business Process (BP) layer captures and represents the various processes, workflows, and collaborations, originally, both formal and informal, which support the BOM layer. By positioning new Event Process layer between the BOM and BP, we believe that BOM can be enhanced with event information while BP layer can be more dynamic and agile. More specifically, detected events can be fed back into business strategy models within this EA layered implementation, which will permit the enterprise to be implemented in a top-down approach and evolved from bottom-up. Data generated from enterprise system, network and services will be stored in event clouds, in order to filter and fire patterns of events. Event will not only trigger business actions but also gives feedbacks on current business assessment for better compliance between business policies and event processing rules.

A representative example of EA modelling language is ArchiMate which is a standard managed by the open group (http://www.opengroup.org /archimate). To illustrate further our approach, we have extended ArchiMate, by stereotyping its graphic notations.

ArchiMate is a visual language that represents end-to-end enterprise architecture in layers of business processes, applications and technology. In each layer, three aspects are considered: active elements that exhibit behaviour (e.g. Process and Function), an internal structure and elements that define use or communicate information (Lankhorst et al., 2010):

ArchiMate provides modelling concepts in each of the three layers:

- **Business:** actor; role; collaboration; interface; object; process; function; interaction; event; service; representation; meaning; value; product; contract.
- **Application:** component; collaboration; interface; object; function; interaction; service.
- **Technology:** node; device; network; communication path; interface; software; service; artifact.

In this paper, we only discuss the extensions that we propose to model the Business layer. Before discussing the extensions, in next section, complex event patterns are introduced, identifying possible event correlation types that can handle various enterprise business cases. A meta-model for complex event representation is also proposed to express the identified event correlation in business architecture.

2.1 Complex Events in Business Architecture

Complex Event Processing (CEP) is one of well-known EDA implementation that supports the architectural pattern in technology level (Luckham, 2008). CEP permits to recognize complex events which need to track all the context of co-related single events such as time, location, interval, repeat, etc., enhancing situation awareness of enterprise. It has been utilized for fraud and hacking detection at the beginning and now expands the application to operational intelligence which focuses on providing real-time monitoring of business processes and activities, and assisting in the optimization of these activities and processes by identifying and detecting situations that correspond to interruptions and bottlenecks.

Example patterns of event correlation are depicted in Table 1. These correlation types have been classified based on two survey papers; a survey on complex events of possible business scenarios (Barros, 2007) and a survey on common features of EPL languages (Bui, 2009), in order to identify the requirements from scenario analysis in business level and then complement it with execution power in technology level. The correlations defined here are categorized into three different types; co-occurrence, time-window and data dependency, and

Table 1: Correlation types of complex events.

Category	ID	Name	Examples
Co-occurrence	C1	Event Conjunction	AND *(an order placed AND payment for the order done)*
	C2	Event Disjunction	OR *(an order cancelled OR new stock arrived)*
	C3	Event Cardinality	Fixed number, range, variable *(a user logged in more than 10 times)*
	C4	Excluding Event	NOT *(an order placed AND NOT paid for the order)*
	C5	Event Sequence	FOLLOWS BY; -> *(an order was shipped -> the order modified)*
Time-Window	T1	Event Time Relation	WITHIN, WHILE, BEFORE, AFTER *(no payment after order placement WITHIN 24hours)*
	T2	Instance Time Relation	FIRST, LAST *(LAST order of user A made more than 3 month before)*
	T3	Absolute Time Relation	BEFORE, AFTER, AT *(AT every Sat 0:0 am)*
Data-Dependency	D1	Event Data Dependency	KEY, PROPERTY *(fraudDetected.user.id = order.user.id)*
	D2	Process Instance Data Dependency	ARG & PARAMETERS *(same order id)*
	D3	Environment Data Dependency	ENVIRONMENT CONTEXT *(User.failedLoginCount >= securityPolicy.maximumLoginFails)*

total 11 patterns are identified for complex event aggregation. In most cases, the assembled patterns are observed in real business cases rather than a single correlation.

As well identifying the correlation type in the table, we also need to identify other types of constraints such as “how many times does an activity need to be notified of an event occurrence”. The other type is related to constraints of context changes.

2.2 Complex Event Modeling

A number of researchers have contributed in event processing area with their outcomes such as models, languages, and frameworks to express complex events. Event driven Process Chain (EPC) is one of the well-known methods that can be utilized for event representation for business modeling (Aalst, 1999). It supports complex event modeling in high level, putting an emphasis on being easy to use and providing a standardized set of visualization elements, whereas not defining exact execution semantics. In EPC, events are treated as first class citizens, i.e. the occurrences of events are fundamental elements of the business action. Each action is always triggered by one or more events; finishing an action again creates events to trigger further actions (Dumas et al., 2005). EPC has several connector types which link event and activity including AND and OR logical operator but no natural way to express time related and data dependant correlations.

Another example is the BPEN (Decker, 2007) that provides graphic notation of complex events as an extended BPMN. The BPEN enables modeling business process based on complex events by representing event co-relation types such as time-window and AND–OR, however, the approach doesn’t provide full scope of event driven EA modeling, concentrating on extended business process modeling using complex event notations.

As previously mentioned ArchiMate also has event concept but it does not have concepts for correlation of complex event that express composition relationship between events, or constraints such as data dependency and time window mentioned in previous section. These concepts and their relationship need to be supported in order to provide a good coverage for the event modeling. Therefore, in order to support all the required features for event modelling, we have developed a meta-model of complex events that extends the ArchiMate meta-model. Figure 2 illustrates the extended ArchiMate meta-model with concepts of CEP.

The meta-model provides essential concepts for complex event representation including event rules and event information elements. Some business cases require expressing the rules of event aggregation explicitly from top level, so these extended elements will reside in both business layer and information system layer of ArchiMate.

3 CASE STUDY

The case study looks at implementation of a student internship programme in the University of West London. The objective of the programme is to guarantee internship placement within three years of study for undergraduate students in the school of computing and technologies.

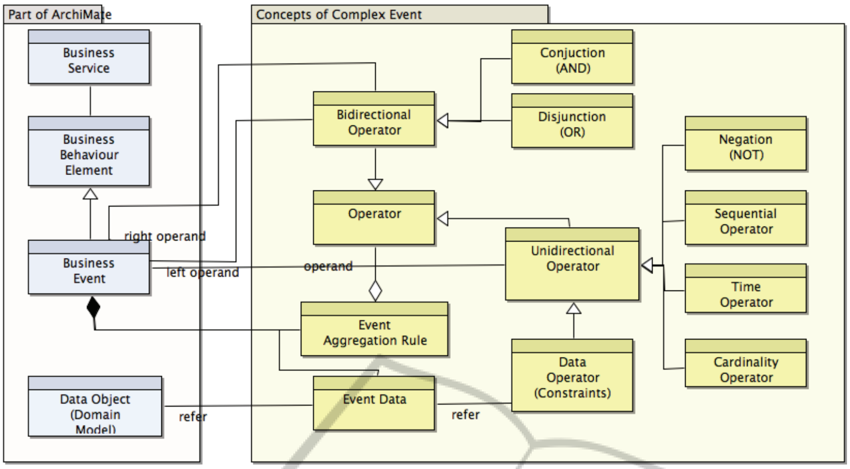

Figure 2: The extended archimate meta-model.

3.1 Business Analysis

To understand the business model of the programme, different values, policies, related events, and correspondent contents have been identified from each stakeholder including school, career office, marketing, and student, through interviews. For the limited space, the paper covers a small part of the models. Figure 3 shows example of the classified elements that captures requirements of TO-BE model. It is explained briefly with a set of policies, events and contents, related to a sub goal 'Maximize the opportunity that student find their own placement'. The students are expected to be active in finding places once they complete PIT5 modules. While school provides a feedback support on student CV and application, there is a requirement for an automated matching service that notifies students once matching opportunity is found.

The identified events here are contain 'Student get ready for internship programme', 'Student registered to internship programme', 'CV uploaded', 'Tutor's feedback uploaded' and etc. Some of these events reside in very abstract layer – for example, 'Student get ready for internship programme', while others are more concrete – for example, 'CV uploaded'. Depending on the scope and purpose of the business case and scenario, the desired granularity of monitoring events can be varied but it is necessary to identify the composition rules of events for some kinds of business activities.

In this example, we have identified an event that triggers a support service for student. When a student has finished PIT5 module but has not registered to the internship programme, we require an automated intervention to provide required support. One of the support rules is that if student finish PIT5 module but not registered the programme within following 3 weeks time, notify personal tutor and email the student with the registration URL and guideline. This example is modelled with event driven approach using complex event meta-model.

3.2 Business Model without EDA

As noted previously, business analysts tend to focus on capture of standard process for achievement of business goals rather than real world reactions of possible variety of events. The figure 4 illustrates the standard process of internship programme management without complex events.

The standard process describes the set of sequential activities as follows. Students register to the internship programme when they completed PIT5 module and then search available opportunity. Once suitable opportunity is found, they request the authorization of the school. Tutor reviews the application with CV and authorize it, then students can apply for the opportunity as next step.

Employers review the application and notify the result to applicant. If the student gets the placement, the internship manager would gather feedbacks from students and employer at the end of the placement. The events and processes need access to the data objects such as 'Student Profile' and 'Application form'. The processes are realized by services from CRM (Customer Relationship Management system), 'Future Skills Application' and Mail System.

Except for the first process, all the other processes are triggered by another process and a change of situation. Therefore, the events not belonging to the standard process are handled

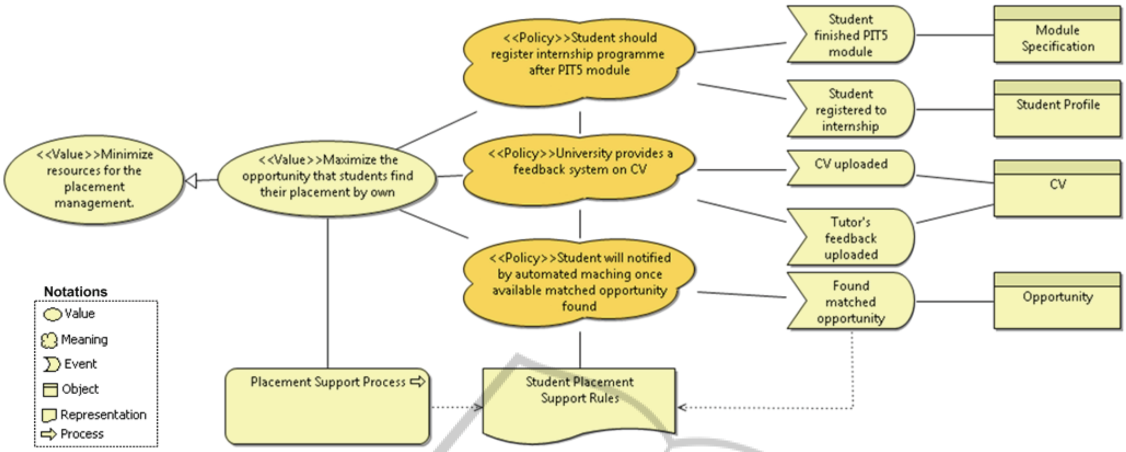

Figure 3: Example of analyzed business model elements.

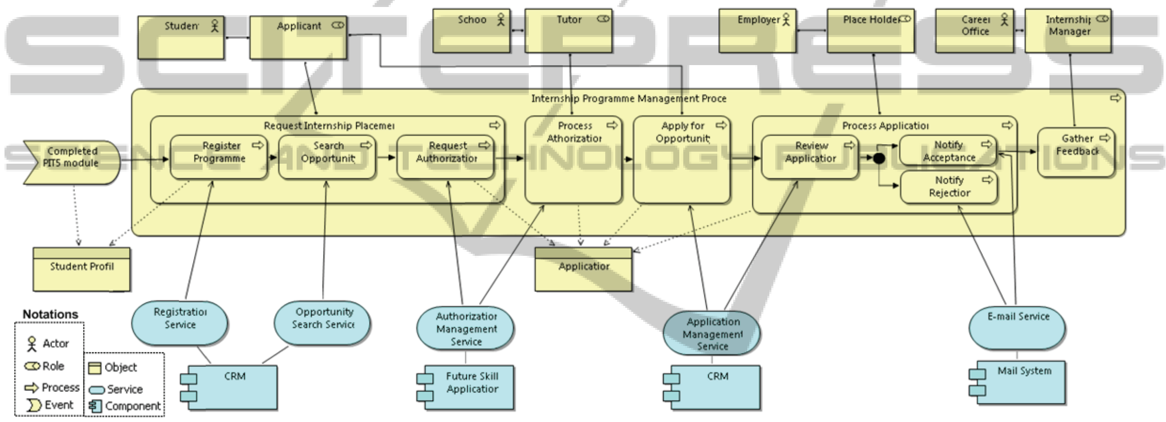

Figure 4: Example process model.

as exceptions. This makes difficult for enterprise to adapt to a changing situation as the business processes are quite static but need to be modified once new situation is unveiled due to complex events.

3.3 Business Model with EDA (ArchiMate Extension)

In this section, we demonstrate how the previous example process model can be redesigned with event driven approach. For this, we apply both event driven process model and complex event model using ArchiMate.

Figure 5 shows the first processe of the example model describe in Figure 4. The business process has input event triggering the process and out event describing the change of state caused from executing the process. We can also indicate which organization unit is assigned to the process. Input and output data objects are also can be placed together. The service and application for realization can be described at the same time. The difference is that the unit of modelling. It is a process consuming an event and producing another event. Although the same sequence of Figure 4 can be discovered when the processes are linked to the same input and output events, the model itself provides more flexibility to possible changes. As the processes are not directly interacted to each other, adding and removing the processes does not require unnecessary modification of the all the related processes. New flow of activities can be easily extended by adding new events and handling processes, without modification of existing ones if it is not essential.

The effects of event driven approach are more powerfully seen with consideration of event hierarchy and event aggregation called complex events. The Figure 6 explains the complex situation that requires tutor's intervention when student does not actively seek internship opportunity. The complex event reflecting the situation is represented using ArchiMate extension meta-model described in Figure 2.

In this example, the complex event named 'intervention required' is detected when the event of 'student completed PIT5 module' is occurred but the event of 'student registered to the programme' is not happened within following three weeks time. Event rules and event data are expressed as stereotypes using 'meaning' element of ArchiMate. As explained in section 2.3, the extended event element is composed of event aggregation rule and event data. This example requires 'AND', 'NOT', and 'WITHIN' operator to express the composition rule. While 'AND' defined co-occurrence of two events, 'NOT' and 'WITHIN' filters events with referencing id and timestamp of event data.

The aggregated complex event is placed in higher level than the single element event of it on the abstraction hierarchy. When the events are aggregated and represented higher level, one the event data belong to the each event are also aggregated and accessed by the composited event – the event data are implemented as parameters of events.

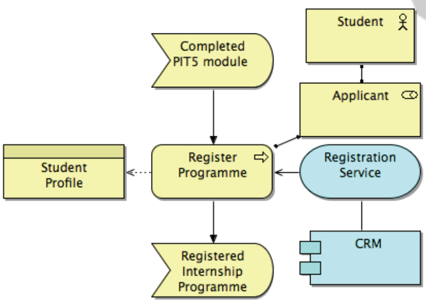

Figure 5: Event driven process models.

In this modeling, events play a major role in business triggers for a chain of actions resulting in the actualization of one or more business objectives. Apparently, the business processes can be designed simpler without concerns of how to detect the changes – event and with more focus on 'what' to do. The business function is linked by events. It is supported by event monitoring and fire mechanism where a monitoring agent or a middleware detects all the changes and examine if the event rules are satisfied and notify when all the conditions are satisfied, hence a service or component needs a notification with event data can register itself with rules how to detect the event.

The event representation model we pursue needs to cover abstraction hierarchy and multiple dimensions in EA. Especially identifying complex events is like defining new abstract event for semantic shift from fine-grained simple events to business concept. Figure 6 assumes that the aggregation rule and data model of the complex event are required to be expressed in the process model to share the information at the business modeling stage. The complex event model can be more simplified once proper graphic notation is developed and applied.

4 CONCLUSIONS

The paper introduces an approach for EDA integration into EA. By using event driven thinking framework, we analyzed a business model, focusing on business events, which can be chained with business policies and constraints and more importantly to aligned to business goals. Identifying events before business services and organization's behaviour are valuable in the phase of business model development. As illustrated in Figure 3 and 5, the extracted business events can be represented as event correlation rules and as business processes running based on service invocation by event notification. By adding a layer for event handling between business process and service, the business process can be modelled in a more light and dynamic way as stated in section 3.3.

In our approach, we have integrated CEP into EA, by proposing modifications to existing models with extended event concepts. We have not looked at having a new EA modelling activity, such as event-oriented activity. However, at least it is clear that existing modeling language are lacking in the support of complex event modelling and a good precise model is necessary for event-driven EA modeling.

As the next step in our research, we are planning to improve the meta-model and develop graphic notation for complex event. Our aim is to provide event modelling notations that helps identifying situation related to Complex Events. By doing so, changes in an enterprise environment and its required response could be easily modelled, providing agility at a strategic and business level. We are also considering adding semantic descriptions and annotation to event models, which may improve situation awareness for more dynamic decision support and autonomic behaviour systems.

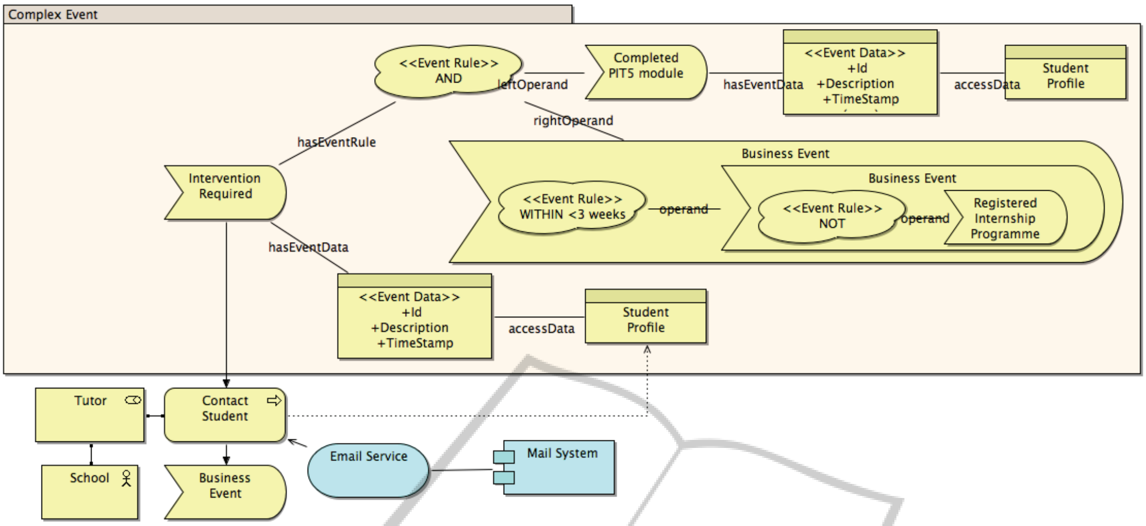

Figure 6: Complex event model.

REFERENCES

Aalst, W., 1999. Formalization and Verification of Event-driven Process Chains. In *Information & Software Technology* 41(10), pp. 639-650

Banger, D. R., Knight, W., & Uk, M. (n.d.). 2008. Maturity Assessment for the Enterprise Architecture (EA) *Function*. Management.

Barros, A., Decker, G. & Grosskopf, A., 2007. Complex Events in Business Processes. In W. Abramowicz, ed. *Business Information Systems*. Springer, pp. 29-40.

Bui, H.-lam, 2009. Survey and Comparison of Event Query Languages Using Practical Examples. *Arbeit*,

Casanave, C., 2009. Service Oriented Architecture Using the OMG SoaML Standard. A Model Driven Solutions, Inc. White Paper http://www.modeldriven.

Clark, T., Barn, B.S. & Oussena, S., 2011. LEAP: a precise lightweight framework for enterprise architecture. In *Proceedings of the 4th India Software Engineering Conference*. ACM, pp. 85–94.

Decker, G., Grosskopf, A. & Barros, A., 2007. A Graphical Notation for Modeling Complex Events in Business Processes. *In Proceedings of Enterprise Distributed Object Computing Conference. IEEE, pp 27.*

Dumas, M., Aalst, W., Hofstede, A., 2005. Process-Aware Information Systems: Bridging People and Software Through Process Technology. Wiley, ISBN: 978-0-471-66306-5.

Engels, G., 2008. Service-oriented Enterprise Architectures: Evolution of Concepts and Methods. *Enterprise Distributed Object*.

Green, N., Bate, C., 2007. Lost in Translation. Evolved Technologist, ISBN: 978-0-978-92184-2.

Kim, T.-Y., Lee, S., Kim, K. & Kim, C.-H., 2006. A modeling framework for agile and interoperable virtual enterprises. *Comput. Ind.*, 57 (3), p.204-217.

Lankhorst, M., 2004. Enterprise architecture modelling—the issue of integration. *Advanced Engineering Informatics*, 18(4), pp.205-216.

Lankhorst, M. M., Proper, H. A. & Jonkers, H., 2010. The Anatomy of the ArchiMate Language. *International Journal of Information System Modeling and Design*, 1(1), pp.1-32.

Luckham, D., 2008. *The Power of Events*: *An Introduction to Complex Event Processing in Distributed Enterprise Systems.* In (eds.). Rule Representation, Interchange and Reasoning on the Web.

The Open Group, 2011. *Using TOGAF to Define and Govern Service-oriented Architectures*, The open group guide.