Complex Event as an Core Aspect of Enterprise Architecture
EDEMF: Event Driven Enterprise Architecture Modeling Framework

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

Keywords: Enterprise Architecture, Business Modelling, Complex Event Modelling, Event Driven Architecture, Complex Event Processing, Model Driven Architecture, Linked Data, Data Integration.

Abstract: For the last decade, Complex Event Processing (CEP) has been emerged as a powerful instrument supporting a more agile and intelligent enterprise. Although several studies have attempted to extend current business modelling standards with elements of complex event modelling they have not fully integrated into Enterprise Architecture (EA). EA provides holistic approach of enterprise business and IT system modelling. In our view CEP is not just an element associated with a business process, but at the centre of business analysis and design. This paper proposes an EA modelling framework that takes CEP as a core aspect of modelling. The framework supports CEP based business architecture within different levels of abstraction: contextual, conceptual/logical, and physical. Meta-models in three hierarchies and model mapping between the meta-models have been developed and demonstrated on a linked data platform in a case study. The case study used the framework as a tool of analysis and modelling, aiming to integrate scattered information as event data by specifying a complex event layer on top of legacy systems. The designed models were then transformed into executable program codes using the model transformation, for example, RDF script from information model; SPARQL query from complex event model.

1 INTRODUCTION

In recent years, there has been an increasing interest in Complex Event Processing (CEP) that empowers enhanced recognition of dynamic situation changes and better response to the changes in a quick and dynamic manner within Event Driven Architecture (EDA) approach (Luckham, 2008). CEP is implemented based on architectural patterns of subscription, detection and publication of changes as complex events. Complex events are usually aggregated or derived from multiple data sources by monitoring a sequence of single events and identifying their correlation. The captured complex events can provide valuable insights on current and future changes that can lead to checks of constraints on business goals and corresponding strategies, as well as triggers of business actions. Several studies have attempted to integrate complex events with business modelling in order to support complex event based business process design. For instance, Decker and et al. (2007) investigated the different types of event correlations and proposed graphical notations for complex event modelling by extending BPMN. Another example is integration of workflow patterns into Event Driven Process Chain (EPC) model to extend EPC with an empty connector and multiple instantiation concepts (Mendling et al., 2005). Although these studies revealed some of the ways in which business modelling were incorporated with complex events, they did not provide a method to identify and model complex events within whole business architecture in a consistent view, for example, identifying business goal and strategies with associated events for its alignment with operational business intelligence.

Enterprise Architecture (EA) is a tool that has been widely adopted to capture all the business requirements in a consistent and holistic view. Service oriented approach is a popular architectural style being integrated in EA. For instance, OMG published a language specification called SoaML for SOA based business modelling. (Casanave, 2009). A leading EA industry consortium, the Open Group, has also published their effort on SOA driven enterprise modelling, demonstrating fitness of their TOGAF framework to service oriented modelling in (The Open Group, 2011). SOA is commonly

Kim H., Oussena S. and Komisarczuk P..
Complex Event as an Core Aspect of Enterprise Architecture - EDEMF: Event Driven Enterprise Architecture Modeling Framework.
DOI: 10.5220/0004440802920301
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 292-301
ISBN: 978-989-8565-61-7

approached with a view of not only identifying and modelling service as a business unit but also supporting communications and interactions between services using service orchestration or choreography. Enterprise Service Bus (ESB) is one of the possible solutions for publishing event message to the subscribed services via a message bus using a secure message delivery scheme such as event message queue. Some of CEP engines have integrated with existing ESB solutions, i.e. Esper with WSO2 (Essl, 2011). Although CEP has been popularly implemented and integrated with SOA based enterprise systems, very little work has been done to supply event driven design in EA.

People tend to be event-driven rather than process-driven. Therefore, event driven thinking can lighten work of business analysis and design, as well as support direct translation of event driven action to business architecture and to IT implementation. In this work, we aim to take complex event as core aspect of modelling from very top-level business analysis to low-level implementation. We propose an EA framework, where modelling is divided into three different hierarchies and five dimensions in order to separate the modelling concerns. Modelling hierarchies and model transformation between different model layers are key techniques approached in Model Driven Architecture (MDA). It has been applied to EA for better business and IT alignment by generating skeleton of program codes from higher-level business models (De Castro and et al., 2009).

As Web becomes a fundamental communication channel, enterprise systems and information gets more decentralized and open. In particular, federated enterprise architecture takes external changes into more important aspects to acknowledge changes (event detection). Besides, understanding domain context when significant event occurred is also essential. A few approaches have proposed semantic CEP that provides semantic reasoning on top of event processing. Semantic web platform is an example implementation of such an approach and has been demonstrated in (Anicic and et al., 2011). In semantic web platform, linked data play a key role for enterprise data integration as it enables enterprise to catch external events by linking external data sources with internal data in Resource Description Format (RDF). Complex events correlation is expressed in a specialized query language for RDF data, SPARQL. We adopted this approach for physical modelling to ground EA models onto semantic CEP platform in the case study. This examines the feasibility of the proposed framework in a use case that investigates solving data integration problem by using CEP and linked data.

The remainder of the paper is structured as follows: Section 2 summarizes related works. Section 3 introduces the proposed modelling framework with its layered architecture and defined meta-models. Model transformation is also described. Section 4 presents the case study. Section 5 concludes and gives an outlook to future work.

2 RELATED WORKS

Works related to ours mainly fit into three areas: event-driven business modelling, model transformation, and linked data integration.

Event-driven Business Modelling. EA has been popularly referenced and used as a tool to supply business modelling with better business and IT alignment check of compliance with external regulations, analysis of business changes. However, current EA frameworks do not provide event-driven business modelling in an integrated manner. Event notation is used in EA modelling framework such as Zachman (Zachman, 1999) or TOGAF (Josey, 2009) but they only support high-level abstraction of event in which an event is represented with its name and relationships with other model notations.

One of classic and popular approaches for event driven behaviour modelling is EPC (Event driven Process Chain) that allows the identification of connections between events and between activities by using AND, OR, and XOR operators. For instance, business cases such as one process get triggered from an concurrence of two events, or one process triggering two different types of events can be identified using EPC. It is well fitted with conceptual modelling but not precise or sophisticated enough for logical level modelling of complex types events. For example, time constraints such as sequence and while, and contextual constraints between events data are hard to be captured and expressed in an EPC model.

There are a few other approaches that attempted to merge workflow pattern to EPC to enhance its expression power with operators of sequence, split and etc. (Scheer et al., 2005) (Thomas and Fellmann, 2007). Based on our knowledge, however, they failed to link contextual constraints with event data at model level, which is necessary to be aware of complex enterprise situations. There are also some works that attempted to extend existing business process modelling notation with more complex types' event operations such as event aggregation rules and its constraints (Decker et al., 2007). Some of researches have proposed a specific way of event representation with model driven development approach in a way that an event type is identified as an object class and an event instance is treated as object instance using Object Oriented Programming metaphor (Rozsnyai et al., 2007). Another example is an event view model in which developer specify event types of SOA artifacts to monitor events using Domain Specific Language (DSL) (Mulo et al., 2010). However, they have not yet been integrated into EA frameworks. Identifying complex events in EA framework requires identifying events in multiple hierarchy and dimensions with more consideration to business analysis, modelling, and its compliance with IT. In that context, we have built a modelling framework at the top of TOGAF to deliver three dimensions of modelling layers with different viewpoints. We have extended its meta-model language, ArchiMate with complex events modelling support.

Model Transformation. MDA approach has been used to assist better business and IT alignment with automated transformation from business models into executable program codes (Soley, 2000). Irrespective of its popularity and productivity, current EA modelling frameworks do not support it fully. Although some meta-models such as Business Process Modelling Notation (BPMN) for business process modelling can be converted to Business Process Execution Language (BPEL) which is executable, we believe that process based modelling does not fully conform to Event Driven Architecture (EDA) in a sense that a process model follows predefined flow rather than responding dynamic event rules. Moreover event enhanced BPMN (Decker et al., 2007), for example, has not been grounded to implementation thus no model transformation is supported.

In our approach, we try to generate lower level models from higher one, for instance, ArchiMate logical information model into physical RDF graph model, and complex event model in extended ArchiMate into executable SPARQL queries.

Linked Data Integration. Integration of scattered information is crucial to support enterprise situation awareness and intelligence in real-time and dynamic way. One emerging approach is the use of linked data (Heath et al., 2012), semantic web. Linked data technology facilitates integration of public open linked data with internal one (Coletta et al., 2012).

There are some works that merge CEP with linked data to provide real-time data processing on the web environment. Wagner argues that there is a shift of ideas regarding web communication style changed from request-response to subscribe-publish. Representative examples of this movement can be seen in some works such as HTTP5 and pubsubhubsub (Wagner and Anicic, 2010). In this context, Anicic (2011)'s work fits well into our approach both for real-time event processing and background knowledge inference by extending SPARQL with temporal RDF operation and grounding it in Logic Programme.

3 EDEMF: EVENT DRIVEN ENTERPRISE ARCHITECTURE MODELLING FRAMEWORK

In our approach, events are at the centre to both the analysis and modelling. The Event Driven Enterprise Architecture Modelling Framework (EDEMF) has been proposed to support the analysis and modelling by adopting complex event as another dimension of EA and incorporating event-action modelling into EA.

3.1 Overall Architecture

The EDEMF was developed based on TOGAF and focused on business and information architecture; i.e. enterprise business motivation, operation and information. The figure below shows identified modelling layers and key perspectives of the framework.

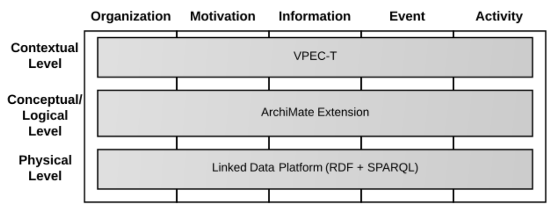

Figure 1: EDEMF Overall Architecture.

In this framework the information is captured in three abstraction layers; contextual, logical, and physical. Within each hierarchy layer, we split modelling concerns into five core viewpoints; organization model, motivation model, information model, event model and activity model. (1) Organization model presents the organization including human resources, departments and their relationships. It identifies stakeholders and their relevant business values and policies. (2) Business motivation model describes business motives with elements of business value and policy. Value includes business goal and strategy. The business motivation is implemented by business process with relevant event, information and service. (3) Information model identifies contents that stakeholders are interested in or produce. In this approach, we assume that all the operations on contents such as creation, modification, and deletion are trigged by event. (4) Event model illustrates all the events that stakeholders are interested in or supposed to trigger. Event correlation types, rules and related information are also captured and verified in this model. (5) Lastly, activity model depicts event driven business activities. The activity model captures not only actions triggered by events but related stakeholders and information, and relevant business values and policies.

In our approach, the five viewpoints are initially established in contextual level for primary business analysis. The contextual analysis is carried out by using the VPEC-T framework (Green and Bate, 2007), in which business requirements are entailed using elements of Value, Policy, Event, Contents and Trust. The contextual model is then translated into the ArchiMate extension meta-model (Kim and Oussena, 2012). Although the ArchiMate allows high-level business modelling with event concept, this is not enough for modelling complex events. Therefore, in our work, the event element that captures event correlation rules and processing are added to both conceptual and logical models. The logical models are then grounded onto the physical layer. For this, semantic web is targeted as an example implementation platform. In this platform all data is shared in linked data format, Resource Description Framework (RDF) and complex event is queried using SPARQL.

3.2 Meta-models

One of our objectives in this work is to provide a consistent modelling view for different EA modelling dimensions. We believe that a unified modelling language makes it possible to enhance communication between different stakeholders and better quality of model artefacts. Thus, we pursue extending ArchiMate with the additional model elements rather than referring different meta-models such as EPC and BPMN for complex event modelling.

We reused some concepts and relationships of EPC in the ArchiMate extension but separated different concerns into different modelling by dividing EPC scope into event modelling and activity modelling. This allows complex event composition model to be built independently using a precise and comprehensive meta-model and to be referred from an activity model later as if it was a single event. An activity is usually triggered by an event and produces an event. The event itself can be related to an organization unit, information object, and document.

We have proposed the ArchiMate extension in (Kim and Oussena, 2012), mainly introducing elements for logical complex event modelling into ArchiMate. We have then improved our approach by connecting the logical modelling to physical modelling. The proposed three modelling layers are illustrated as following.

- Contextual modelling – Contextual meta-model was built based on identifying five core elements of VPEC-T. These elements are mapped into ArchiMate in further modelling phases. So far, we have not mapped the "Policy" element yet. This is the element that realizes business values and constraints on business activity.
- Conceptual/Logical modelling – As shown in Figure 2, Event concept was extended with capability of event correlation design and associated with the other model elements such as value, activity. Some of relationships were also added. An example of this is a relationship between information object and event object defined to enable context reasoning from event data. Another significant change we made was that a rule based activity design approach was used, instead of using process model in which a predefined sequence of activities is identified. As a single activity is identified with associated in and out events, various sequences or combinations of activities can be handled depending on the patterns of event occurrence.

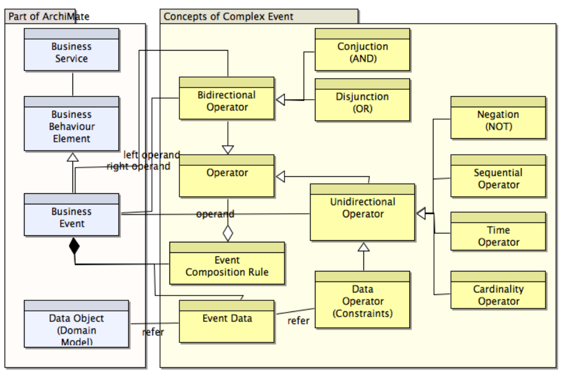

Figure 2: The Extended ArchiMate Meta-model.

- Physical modelling – In this work, linked data platform has been used as an example for the implementation. To support physical modelling of it, we identified key modelling elements that capture the semantics and syntaxes of RDF and a specialized SPARQL; EP-SPARQ, as shown in Figure 3 and 4. The figures illustrate physical information and complex event meta-model respectively. In the physical meta-model, the information model is described using a RDFTriple element which consists of Subject, Predicate and Object. Subject and Object can be a sub-element of either ClassResource or LiteralResource. The information model is also able to present URI and Repository elements for the use of data link.

For the complex event model, ComplexEvent element has been introduced in our extension. It is composed of EventData and EventRule. EventData is used to indicate data retrieved at the time of the event occurrence. The data can be extracted directly from member event or aggregated in some level, or loaded from historical data repository. In our model, it can be either literal or class type for further linking and interpretation. EventRule represents correlation rules of member events. The operators include not only logical types such as JOIN or UNION but also the element for time context enriched SPARQL expressions such as SEQ and EQUALS. Functions to constraint event data and rule can also be identified using Function and Filter element. For example, getDuration() and getStartTime() identify interval time of two different events and first occurrence time of an event respectively.

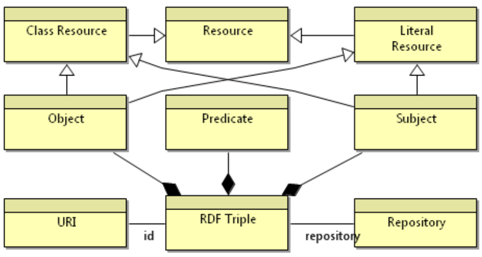

Figure 3: Physical Information Meta-Model.

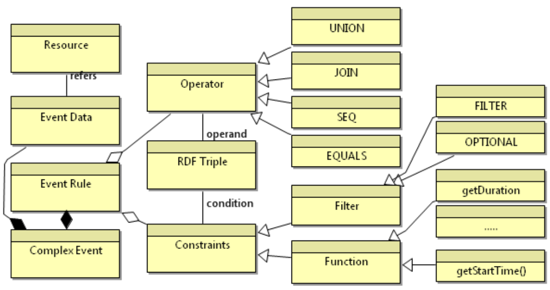

Figure 4: Physical Complex Event Meta-Model.

3.3 Model Transformation

In our approach physical model is built from logical ArchiMate model using model transformation, which generates skeleton of executable codes. The Table 1 shows the mapping rules of corresponding elements and relationships. For example, an

Information object in ArchiMate is converted into a Subject element of RDF Triple that is actually mapped to a RDFS Class. The Association relationship between two Information objects becomes Predicate element, which generates a RDFS Property element having both source and target objects references as Domain and Range elements respectively. Inheritance relationship is directly mapped into Subclass, and Aggregation into Property in similar way of Association.

Table 1: The Model Transformation Rule.

Logical: ArchiMate Model		**Physical:** Linked Data CEP Model	
Information Model	Information Object	RDFS Model	RDF Triple: Subject
	Association		RDF Triple: Predicate
	Inheritance		Subclass
	Aggregation		RDF Triple: Predicate
Complex Event Model	Atomic Event	EP-SPARQL	RDF Triple(s)
	Event Data		SELECT
	Event Rule		WHERE
	AND		SEQ or JOIN
	OR		UNION
	NOT		NOT EXIST or OPTIONALSEQ
	Sequential Operator		SEQ
	Time Operator		FILTER
	Cardinality Operator		SEQ or EQUALS
	Data Operator (Constraint)		FUNCTION

Using this model mapping, ArchiMate logical information model is transformed into physical RDF graph model, and complex event model in extended ArchiMate into executable SPARQL queries. These transformations have been pseudo coded and experimented in the case study.

4 INTEGRATION OF ENTERPRISE DATA RESOURCES USING CEP

Integration of real-time event data streaming using CEP can provide proactive and adaptive situation responses (Engel et al., 2012) (Xu et al., 2012). CEP has shown its advantages in area of business activity monitoring but it can also be utilized in data integration use case. In our work we integrated enterprise data resources as linked event data in RDF format and build a system architecture in which changes of enterprise can be captured and interpreted using CEP layer.

The proposed approach has been applied to a JISC funded project at the University of West London. The project aims to design intelligent student internship management system by looking at ordinary event data from different data sources and applications. The ordinary events are then specialized as notable events and notified to event processing agents using CEP system.

Business architecture is designed using the EDEMF framework. EA models including business motivation, information, and activity models are modelled based on event-centred analysis using questions such as: Which event indicates change of business objectives?; Which event manipulates information?; Which event people are interested and how they respond to (activity is followed by an event) ? The analysis then led to build TO-BE model by carrying activities such as analysing business scenarios; identifying information models, activity models. It was then followed a gap analysis activity such as converting relational or flat data sources into linked data.

This case study demonstrates how business requirements can be captured in event-driven way and how the requirements can be implemented on event-data integration layer using the EDEMF.

4.1 Scenario

The University of West London engages in and actively promotes Student Internship Programme (SIP) by guaranteeing students placements during the period of their course. Within the programme, different stakeholders have different goals; Career Office wants to devote more to provide students with useful help of finding the opportunities. School and academic staffs try to support the students with personal advice. They monitor activities of students and provide practical help in preparing CV and interviews in a classroom. Students expect to get more help of searching the right placements through abundant information on potential employers and opportunities from reliable sources. The scenario below depicts 5 different phases of the SIP management system which represents the system lifecycle from students' point of view. 9 complex events are also described as examples.

- Phase 1: Registration

Students in school of computing usually take PIT5 (Professional IT level 5) module in their second year and register with Career Service once they finished the module. However, some students do not register with the Career Service and some even fail to finish the PIT5 module by second year. The SIP management system needs to monitor and react to these exceptional situations. For example, ***<CE.1>*** *If a student finished PIT5 module but has not registered to Career Service within following 3 weeks' time*, the system alarms both the student and the student's personal tutor.

- Phase 2: Finding an Opportunity

Students who have subscribed to the Career Service get a notification whenever new available opportunity is registered. The system is required to monitor students' activity and interactions with the Career Service and therefore to react actively. For example, ***<CE.2>*** *If no notification is published to a subscriber for a week*, the system recommends the subscriber to check and mend subscription details such as keywords, category, and period. Another example is that ***<CE.3>*** *If no application is submitted until 3 days before the deadline of an application*, the system informs relevant tutors of the opportunity details.

- Phase 3: Appling an Opportunity

When students find an opportunity they submit their application for approval from school and then can apply for the placement after the approval. The system also needs to monitor and handle complex situations in this phase – for example, ***<CE.4>*** *When authorization is emergent including in the case of that the approval request is arrived just less than 3 days before application deadline*; ***<CE.5>*** *No application has been selected more than 3 times from same opportunity provider* (this can indicate a possibility of the unreliable or hazard employer); ***<CE.6>*** *A student has been authorized more than 3 times but has not made real application*; ***<CE.7>*** *Cover letter has not been changed while applying for two different opportunities*, etc.

- Phase 4: Taking a Placement and Making Feedback

If a student manages to get a placement

successfully, the university may visit his or her working place to take an investigation and then provide him or her with proper advice. After the internship, the student must submit a reflection report. While students taking the placements, diverse and complicated situations can be happen. For example, a student may be involved in burglary or accidents during their stay at the company; an employer suddenly changes their mind and cancels the internship opportunity. The system needs to detect these various situations and to respond to them by reasoning unidentified events and placing them in most similar, pre-identified situations. It is like identifying an unexpected pattern of events and then publishing the event data to the event subscriber of the most similar event. For example, in the system, ***<CE.8>*** *If a student finishes an internship but does not submit a reflection report until a week after deadline*, it is informed to a personal tutor. However, ***<CE.9>*** *If a student not having submitted reflection report submits mitigation report*, the system doesn't alarm the personal tutor as it places a mitigation report submission event as a sub event of reflection report event, but informs different correspondent of the event data according to the different situation of the mitigation case.

4.2 Model Artefacts

The analyzed requirements have been written in the proposed ArchiMate extension. The model artefacts produced includes contextual models, logical EA models and transformed physical EA models. The five core viewpoints were designed. For reason of limited space, we only present a part of the model we have designed in Figure 5. The model covers Phase 1 of the scenario.

4.2.1 Transformed Physical Models: Scripts

The initial physical models have been transformed from logical ones and completed with additional details. The procedures for RDF script generation are (1) From the logical information model, identify information objects belonged to different data sources; (2) Then produce physical information models for each data source by using model mapping rules in Table 1; (3) The URIs containing base uri and meaningful name, and data repository information are added. The models are represented in the format of RDFS and the part of generated scripts is here. This script implements Student subject with moduleEnrolment predicate which has Module object.

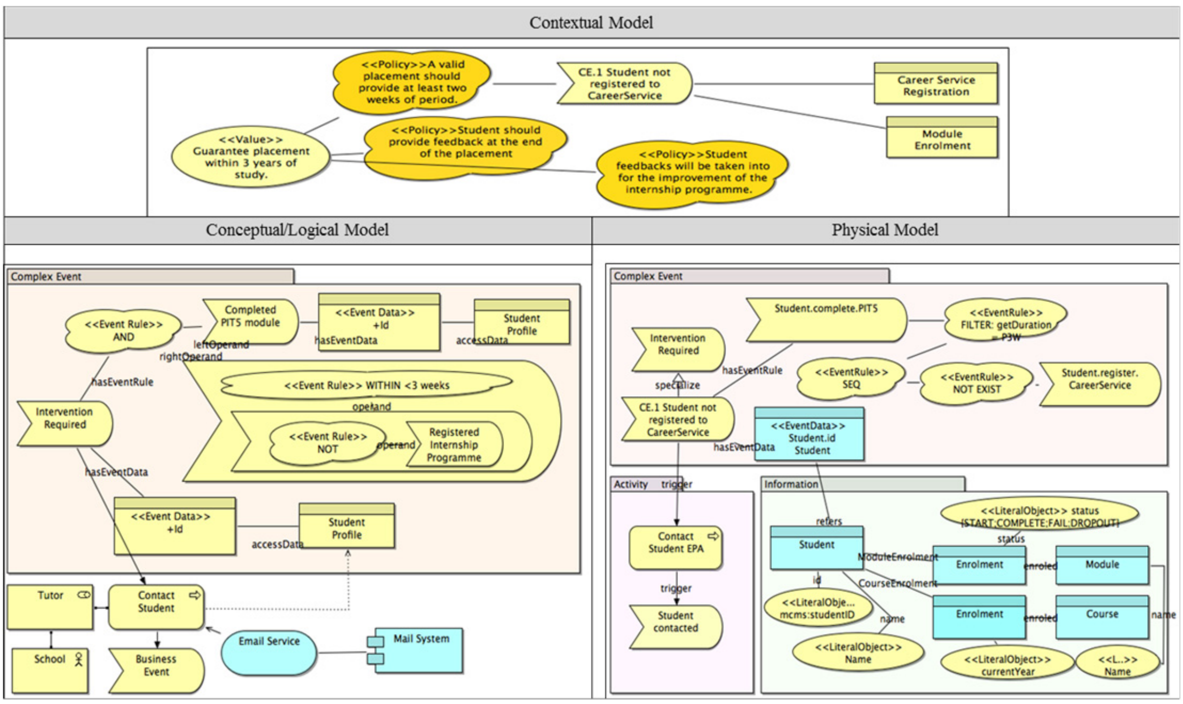

Figure 5: EA Models of Registration Phase.

4.2.2 Transformed Physical Models: Scripts

The initial physical models have been transformed from logical ones and completed with additional details. The procedures for RDF script generation are (1) From the logical information model, identify information objects belonged to different data sources; (2) Then produce physical information models for each data source by using model mapping rules in Table 1; (3) The URIs containing base uri and meaningful name, and data repository information are added. The models are represented in the format of RDFS and the part of generated scripts is here. This script implements Student subject with moduleEnrolment predicate which has Module object.

```
<rdf:RDF xml:base="http://luci.uwl.
ac.uk/unite/" xmlns:rdf="http://
www.w3.org/1999/ 02/ 22-rdf-syntax-
ns#" xmlns:rdfs=
"http://www.w3.org/2000/01/rdf-
schema#">
<rdfs:Class rdf:ID= "Student" />
<rdfs:Class rdf:ID= "Module" />
<rdfs:Property rdf:ID=
"moduleEnrolment">
<rdfs:range rdf:resource=
 "Module"/>
<rdfs:domain rdf:resource=
"Student"/>
<rdfs:isDefinedBy rdf:resource="http
://luci.uwl.ac.uk/unite/ue.rdfs"/>
```

The physical complex event model has also been transformed from the logical one with similar procedures. The transformation is not fully automated as logical model does not always have sufficient information to decide which physical model element is best for mapping. Example of parts requiring your attention during transformation is such as AND operator, and atomic event elements. AND can be converted to either SEQ or JOIN depends on scenario. Converting atomic event name to a RDF triple also requires human interpretation. The below scripts shows example complex event queries converted into EP-SPARQL.

```
PREFIX  cs: <http://luci.uwl.ac.uk/
linkeddata/careerservices/>
PREFIX ue: <http://luci.uwl.ac.uk/
linkeddata/unite/>
PREFIX luci:
<http://luci.uwl.ac.uk/linkeddata/>

SELECT    ?std   ?stdID
WHERE {
?std     a           ue:Student;
         ue:moduleEnrolment ?enrol;
         ue:id           ?stdID.
?enrol   ue:enroled      ?mod.
?mod ue:name     "PIT5".
?enrol   ue:status       "COMPLETE".
SEQ   {
NOT EXIST {
?std_cs  a           cs:Student;
        cs:id           ?stdID.
}}
FILTER ( epsparql:getDuration() =
"P3W"^^xsd:duration )
```

This query implements <CE1>. It checks if a student enrolment status of "PIT5" module changes to "COMPLETE". Once it happens it monitors if the student is added into CS for a 3 weeks time window. If the student ID appears within the time window, the complex event does not triggered.

```
SELECT ?std
WHERE {
?std a          cs:Student;
     cs:application ?app.
?plc cs:place       ?app;
     cs:status      "COMPLETE".
OPTIONALSEQ {
?plc cs:gather      ?feedback.
?feedback   cs:type
"REFLECTION_REPORT".
}}
FILTER ( !BOUND(?feedback)
   &&  (  epsparql:getDuration()  =
   "P2W"^^xsd:duration ) )
```

This script shows <CE9>, the case of monitoring a student after his or her placement finished. If no report is submitted within two weeks time, this event is triggered with student information.

4.3 Example System Architecture

Integration of data is always a challenge for an enterprise. Heterogeneous data formats, various data sources, different technologies used in these systems make the integration process difficult. We anticipate that event driven EA can be used as a tool to easy the integration. Using this approach data integration can be designed and implemented in a separated layer. Hence, de-coupled from business process and data sources.

In this experiment, there were 4 different types of data sources identified; (1) relational data in Oracle server containing staff, student, and module information, (2) flat data in a spreadsheet holding students employment statistics, (3) xml data in a RSS feed delivering students employment statistics,

(4) rdf triple data including course specification. All data recourses were converted into RDF via RDFAdapters partly provided by existing tools such as D2R server but mostly developed.

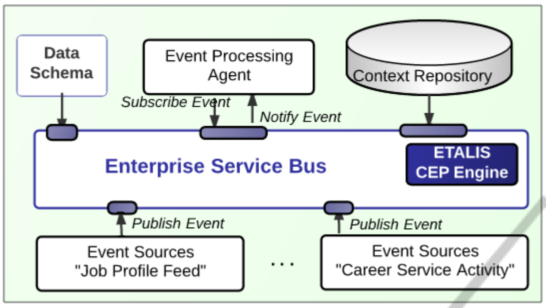

Figure 6: System Architecture.

Historical data converted into linked data was stored in SeSame server. The real-time data published through a service bus is monitored to detect complex event pattern by an open source CEP engine, ETALIS.

5 CONCLUSIONS

This paper proposed an EA modelling framework that provides event centred business analysis and modelling. Meta-models for three different hierarchical modelling and model transformation between meta-models have been developed and demonstrated in a case study. The case study experiment showed that data integration using linked data with CEP could be successfully derived using the proposed approach.

Our future works include developing model editor to support ArchiMate extension modelling and physical model mapping with tool assistance. From complex event layer located between operational layer and process layer, designing interaction model is also another part we aim to develop. For this, we are considering adding semantic descriptions and annotation to ArchiMate, which may make it possible to link event with business value and activity in more proactive and autonomic way.

REFERENCES

Anicic, D., Fodor, P., Rudolph, S., & Stojanovic, N., 2011. Ep-sparql: a unified language for event processing and stream reasoning. In *Proceedings of the 20th international conference on World Wide Web*, pp. 635-644.

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

Coletta, R., et al., 2012. Public Data Integration with WebSmatch. *arXiv preprint arXiv:1205.2555*.

De Castro, V., Marcos, E. and Wieringa, R., 2009. Towards a service-oriented MDA-based approach to the alignment of business processes with IT systems: From the business model to a web service composition model. *International Journal of Cooperative Information Systems*, 18 (2). pp. 225-260.

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.*

Engel, Y., Etzion, O. & Feldman, Z., 2012. A basic model for proactive event-driven computing. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, pp. 107-118.

Essl, S., 2011. Extending an open source enterprise service bus for multi-tenancy support. *Master Thesis, University of Stuttgart*, http://elib.uni-stuttgart.de/opus/volltexte/2011/6986

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

Heath, T., Bizer, C., & White, B., 2012. Linked data: evoluving the web into a global data space. *Journal of Web Engineering*, *11*(2), 177-178.

Josey, A., 2009. *TOGAF Version 9: A Pocket Guide*. Van Haren Pub.

Kim, H., Oussena, S., 2012. A Case Study on Modeling Complex Event Processing in Enterprise Architecture. Proceedings of the 14th International Conference on Enterprise Information Systems (ICEIS 2012), June, Wroclaw, Poland

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.

Mendling, J., Gustaf N., and Markus N., 2005 "Yet another event-driven process chain." *Business Process Management,* pp.428-433.

Mulo, E., Zdun, U., & Dustdar, S. (2010). An event view model and DSL for engineering an event-based SOA monitoring infrastructure. Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems, DEBS'10.

Rozsnyai, S., Schiefer, J., & Schatten, A. (2007). Concepts and models for typing events for event-based systems. Proceedings of the 2007 inaugural international conference on Distributed event-based systems - DEBS '07.

Scheer, A. W., Thomas, O., & Adam, O. (2005). Process modeling using event-driven process chains. *Process-Aware Information Systems*, 119-146.

Soley, R., 2000. Model driven architecture. *OMG white paper*, *308*, 308.

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

Thomas, O., & Fellmann, M., 2007. Semantic epc: Enhancing process modeling using ontology languages. In *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management,* pp. 64-75.

Wagner, A. & Anicic, D., 2010. Linked data and complex event processing for the smart energy grid. *Proc. of Linked Data in the Future Internet at the Future Internet Assembly*, pp.1613–0073.

Xu, Y., Stojanovic, N., Stojanovic, L., Cabrera, A., & Schuchert, T., 2012. An approach for using complex event processing for adaptive augmented reality in cultural heritage domain: experience report. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems,* pp. 139-148.

Zachman, J. (1999). A Framework for Information Systems Architecture. *IBM Systems Journal, 38*(2-3), 454-70.