Modelling Enterprise Applications using Business Artifacts

Vladimír Kovář[1], Marek Beránek[1] and George Feuerlicht[2]
[1]*Unicorn College, V Kapslovně 2767/2,130 00 Prague 3, Czech Republic*
[2]*Department of Information Technology, University of Economics, Prague, W. Churchill Sq. 4, Prague 3, Czech Republic*

Keywords: Business Artifacts Modelling, Unicorn Universe Process.

Abstract: Most of the existing modeling languages such as UML, BPML, etc. that attempt to capture the semantics of real-*world* objects produce complex technical models that are not suitable for business professionals. Another important limitation of traditional modeling approaches is the lack of a mechanism for modeling the lifecycle of business objects. These limitations have motivated recent interest in alternative approaches such as business artefact modelling that provide a unified representation of data and processes in the form of business artifacts. In this paper we describe the Unicorn Universe Process method that uses business artifacts as a fundamental building block of information systems. We illustrate the application of this method using a University Assignment Submission case study scenario.

1 INTRODUCTION

In order to support the development and evolution of Information Systems (IS), conceptual models must facilitate effective communication among different stakeholders (i.e. management, business analysts and domain experts), and at the same time provide a sufficiently precise blueprint for the implementation of enterprise applications. According to Hull (2008) "The basic challenge in business process modeling is to find mechanisms whereby business executives, analysts, and subject matter experts can specify, in an intuitive yet concise way, the framework and specifics of how the operations of a business are to be conducted." Graphical modelling languages play a particularly important role in promoting common understanding of business processes and information requirements across the entire organization. Most of the existing modeling approaches such as UML (www.uml.org), BPML (www.bpmn.org) or Object-Oriented Modelling (Rumbaugh et al., 1991) that attempt to capture the semantics of *real-world* objects tend to produce complex technical models that are not suitable for business professionals and domain experts. Another important limitation of traditional modeling approaches is the lack of a mechanism for modeling the lifecycle of business objects, resulting in complex and potentially inconsistent models that represent business objects as separate entities as they evolve through their lifecycle. For example, a *student* is represented as a *prospective student* prior to the admission into a course, *enrolled student* following the enrolment, and a *graduate student* after completing the course.

To address the limitations of traditional modeling methods Nigam and Caswell (2003) have proposed the Business Artifacts approach. *Business artifacts* combine data and process aspects into a holistic unit and serve as the basic building blocks of IS (Cohn and Hull, 2009). An artifact lifecycle is an integral part of this modeling approach allowing more intuitive representation of business objects and their evolution (state changes). Furthermore, the artifact-centric approach is regarded by some authors as helping in understanding of the relationship of process and data.

In this paper we describe the Unicorn Universe Process method (Kovář, 2011) that has been used successfully to support the development of information systems by Unicorn Group of companies (www.unicorn.com) both internally and for customer projects over the last decade (Kökörčený and Kovář, 2015). In the following section (section 2) we discuss related research in the area of artifact-centric modelling. Section 3 is a description of the Unicorn Universe Process method and section 4 illustrates the application of this method using a University Assignment Subsystem case study scenario. Section 5 includes our conclusions.

Kovář, V., Beránek, M. and Feuerlicht, G.
Modelling Enterprise Applications using Business Artifacts.
DOI: 10.5220/0006334404190425
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 419-425
ISBN: 978-989-758-248-6

2 RELATED WORK

The artifact-centric approach to work-flow modeling was originally introduces by Nigam and Caswell (2003) and has been used extensively by IBM on both internal and customer projects. The authors developed a technique, called OpS (Operational Specification) that defines notations and methodology for business process design that is intuitive and suitable for business communication, and at the same time is *actionable*, i.e. can be mapped to execution-level models and implemented using a workflow engine (Cohn and Hull, 2009). OpS focuses on the flow of business information artifacts and is a superset of the activity flow diagrams in UML. Unlike the object-oriented concept of Business Objects (Sutherland, 1995) artifacts are instances of *real world* objects that have a unique identity and progress through a lifecycle. OpS includes the concept of Business Tasks that operate on artifacts and repositories where artifacts can be stored to await further processing. The authors argue that the business value of artifact-centered thinking is realized in three main areas: flexibility in representation, ability to analyze change, and ability to manage implementation of systems that support the business. According to Hull (2008) IBM customers have found that the artifact-centric approach has a "great intuitive appeal to business managers, can bring substantial new insights to the managers, and can greatly facilitate communication about business processes between different divisions and regions of an enterprise."

The *Guard-Stage-Milestone* (GSM) meta-model for lifecycles is a semi-declarative model based on Event-Condition-Action (ECA) paradigm developed at IBM Research in the context of the Project ArtiFact (Hull et al., 2010). The GSM model is an evolution of the BEL's (Business Entities with Lifecycles) model that includes an *information model* that captures business-relevant data and a *lifecycle model*, that specifies the progress of the entity through the business by responding to events and invoking services, including human activities. Unlike BEL's meta-model for entity lifecycles that are based on finite state machines, the GSM model uses declarative approach for specifying lifecycles. GSM uses three main constructs: *milestone* – business-relevant operational objective, *stage* – that contains activities that may involve calls to external services, and a *guard* – a condition or a triggering event that allows entry into the stage when the condition is satisfied. Project ArtiFact had three broad objectives: (i) to enable business-level stakeholders to work with GSM specifications, (ii) to produce actionable specifications that can be converted into an implementation (i.e. a running system), and (iii) support for existing BPM (Business Process Management) standards such as BPMN (Business Process Modeling Notation) (Silver and Richard, 2009). The GSM model includes a programming language GSM-L, a prototype engine Barcelona that supports implementation of GSM specifications and a formal specification of the GSM meta-model.

A four-dimensional framework called BALSA (Business, Artifact, Lifecycles, Services, Associations) was used by Kunchala et al. (2014) to review existing approaches to artifact-centric modeling. The authors argue that traditional activity-centric business process modeling that uses task and control-flow constructs only defines how business processes operate without revealing the resulting data. The BALSA framework artifacts have unique identity and content that consists of information and lifecycle model. BALSA artifact *lifecycle* defines key business-relevant stages through which an artifact evolves from its initiation to completion. *Services* are described as business tasks or actions that act on artifacts, and *associations,* and specify relationships between services, artifacts, and constraints.

In parallel with these research efforts business artifacts have been incorporated into software development methodologies such as RUP (Rational Unified Process) (Kruchten, 2004) and used extensively by IT practitioners. RUP was initially developed by Rational Software, and further enhanced by IBM following the acquisition of Rational Software in 2002 (https://www.ibm.com/software/rational). RUP is closely associated with UML (Unified Modeling Language), a visual modeling language developed by Grady Booch, Ivar Jacobson and James Rumbaugh with the intention to provide a standard visual language for software engineering (Rumbaugh et al., 2004).

3 uuProcess METHODOLOGY

The Unicorn Universe Process (uuProcess) is a comprehensive methodology that supports a range of business activities covering the core functional areas of strategy, marketing, sales, production, customer support and management of the relationship with customers and partners. uuProcess includes guidelines for management organizational assets, including people, property and organizational knowledge. In this paper we focus on the modelling component of the methodology and the Unicorn Universe Business Modelling Language (uuBML). uuProcess virtualizes *real-world* objects using artifacts and models organizations from using a Static View that consists of artifacts (e.g. students, lecturers, assignments, etc.) and associations between artifacts, and a Dynamic View that describes the behavior of the system and models business processes. Artifacts represent *real-world* objects that constitute the application system and include a model of their lifecycle. A business process consists of a sequence of activities with clearly defined objectives such as producing a product, or generating added value for a customer. Activities can result in changes in the state of an artifact that progress the artifact through its lifecycle stages.

uuProcess was initially inspired by existing system development and management methodologies including ITIL (ITIL, 2014), Prince2 (Lianying et al., 2012), and RUP (Manalil, 2011, Kruchten, 2004), and later extended and enhanced based on extensive practical experience with management of organizations and the development of software (Kovář, 2011). In the following sections we describe a subset of the uuProcess methodology with focus on artifact modeling.

3.1 uuProcess Artifacts

Artifacts constitute basic IS building blocks and hold information about tangible objects (e.g. cars, products, books, etc.) and intangible objects (e.g. product defects, enquiries, etc.) (Kruchten, 2004). The content of a uuProcess artifact (e.g. a company car) stores descriptive information such as car manufacturer, technical specifications, photographs of the car, etc. Activities related to the company car including car purchase, registration, insurance, maintenance records, and car sale are managed within the artifact lifecycle. Additional documents such as the lease and insurance contracts, registration documents, etc. are stored in related artifacts and accessed via references.

As illustrated in Figure 1, uuProcess artifacts are based on templates called *Meta-artifacts* (e.g. employee contract Meta-artifact) with pre-defined internal structure and a lifecycle that can be further customized for specific use case scenarios. Business process activities are closely related to the artifact lifecycle and their execution is controlled via access privileges assigned to roles, so that for example only subject coordinators can set student assignments. Each artifact belongs to a single *Business Territory* that typically maps to an organization, so that multiple organizations can coexist on a single multi-tenant Unicorn Universe platform.

uuProcess artifacts are typically created as a result of activities and consist of four basic components: *Properties, Sheets, Attachments and Comments*. *Properties* store structured information that can be accessed programmatically and consist of simple elements (i.e. numbers, strings, dates, references, etc.) or complex (structured) elements (e.g. JSON files). An artifact can contain multiple sheets - a semi-structured XML documents that contain different types of objects, including tables, images, paragraphs, chapters and HTML5 widgets. Comments represent notes related to the content of sheets. Other types of objects (e.g. binary files) can be included as attachments and form an integral part of the artifact. The system maintains versions of sheets, attachments and comments as these objects are modified throughout the artifact life cycle.

Artifact life cycle can comprise three generic types of activities: *Tasks*, *Messages* and *Time Reservations*. Tasks are activities that require a response from the user that the activity was assigned to; messages have a similar function to tasks, but do not require a response. Time reservation are typically used to record meetings in a diary.

Access to artifact content is controlled using access privileges that are derived from the organizational structure and managed using *Roles*. Most of the authorization rules are derived from use cases and are *inherited* from corresponding meta-artifacts. uuProcess uses three basic mechanisms for access control: *Security Levels*, *Implicit Access Rights*, and *Explicit Access Rights*. Each artifact is associated with a security level and the system ensures that only users with the appropriate security level (i.e. equal to or higher than the artifact security level) can access the information contained within the artifact.

Implicit rights are derived from the organizational structure and are assigned to roles to enable authorized users to access artifacts in various modes (i.e. read, write, update, etc.). Explicit rights are granted by the artifact owner (creator) and enable individual roles to execute use cases on the artifacts independently of the organizational structure.

3.2 Visual Modelling Language uuBML

The Unicorn Universe Business Modelling Language (uuBML) is a tool for visual modelling and communication developed by Unicorn and used extensively for modeling information systems for clients as well as for communication within the Unicorn organizations. A key uuBML feature is that it is designed for business communications making

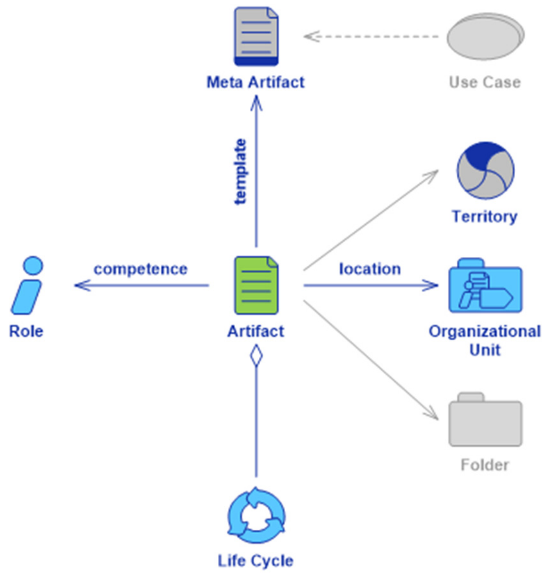

Figure 1: uuProcess artefact.

uuBML diagrams easy to understand for business people without requiring extensive training. At the same time, uuBML is sufficiently rigorous to form the basis for systems implementation (Unicorn, 2016a). We note that all diagrams in this paper are drawn using uuBML.

Diagrams created in uuBML are called *Schemas* and include objects drawn on *Stencils* using icons. Available icons are organized according to themes and business functions (strategy, sales, marketing, management, software development, education, transport, etc.). In addition to icons uuBML schemas include connectors, callouts, blocks, text, titles, legends, and various other symbols. uuBML uses color coding to indicate the relative importance and status of objects. For example, green color indicates that the object state is normal and red color indicates a problem (Unicorn, 2016b).

Schemas can be created using stencils in Microsoft Visio or using uuBML Draw, a HTML5 component fully integrated within the Unicorn Universe platform. Similar to Entity-Relationship models, three basic types of associations (relationships) can be presented in uuBML models: 1:1, 1:N, and N:M (Unicorn, 2016b).

3.3 uuProcess SDLC

The uuProcess SDLC (System Development Lifecycle) has four phases: (1) Analysis and Design, (2) Construction, (3) Pilot and (4) Run Time. The final deliverable is a fully functional uuApp (Unicorn Universe Application) application. Pilot Application is a small-scale application that is used to test the functionality of the application before its full production release. The output of the Analysis and Design phase is a HLC (High Level Concept) specification of the uuApp application. The HLC document has five major sections: Key Concept, Product View, Process View, Business Use Cases, and Organizational View.

The *Key Concept* section describes the main purpose of the application and the key underlying concepts for the implementation of the application. The *Product View* section describes all artifacts, their life cycles and associations. The *Process View* section contains details of business processes and sub-processes (activities), capturing the relevant business roles that interact with the business processes. The functionality of the application is described in the *Business Use Cases* section using diagrams capturing basic process flows for each business use case.

The Business Use Case template defines business roles that can execute or interact with the business use case, pre-conditions that allow its execution (e.g. assignment can only be submitted by active students), the process flow of the use case, and the layout of the use case. The *Organizational View* section defines the location and the structure of the organizational unit within the Business Territory.

4 uuProcess CASE STUDY

In this section we illustrate the application of the uuProcess method using a small-scale scenario depicting the Assignment Submission subsystem at the Unicorn College (UC). The UC information system consists of a set of applications organized according to the main functional areas of the college. A key UC functional area is *Teaching* which is a part of the *Educational Process*, and Assignment Submissions is an important sub-process of the Teaching functional area. As the first step in the SDLC we have created the HLC document defining the Key Concepts and objectives of the application: *"to develop an application for assignment submissions and assessment"*.

Next, we have identified the main artifacts and the objects that constitute these artifacts as illustrated in Figure 2. The assignment submission use case scenario includes two main artifacts: Assignment Handout artifact and Student Assignment Submission artifact. The artifacts Student, Lecturer and Course are external to this subsystem (indicated in gray color) and do not include all the details such as associations with other artifacts. The Student artifact represents a student enrolled in the course in a given term and contains basic information about the student, course attendance, scores, etc. The Lecturer artifact represents a lecturer teaching the course during the term.

The artifact Assignment Handout contains detailed instructions for the students specifying the assignment tasks, the marking scheme and how to submit the assignment. The activity *Do Assignment* is part of the life cycle of the Assignment Handout artifact that notifies the students about the availability of the Assignment Handout. This activity is of type *task* indicating that the Lecturer needs to respond following the completion of this activity. Graphical representation of the activity is an exclamation mark in uuBML. When students submit their assignment solutions, an instance of the artifact Student Assignment Submission is created containing various attachments and an activity *New Assignment Submission to Review* that notifies the lecturer that an assignment is ready for marking. Lecturer assessment of the assignment including the marks and feedback comments is stored in the Assessment object which is a part of the Student Assignment Submission artifact. Each Student Assignment Submission artifact instance has a single Assessment object with an aggregation association indicating that the assessment is a part of the Student Assignment Submission artifact. The *New Student Assignment Submission to Review* activity is of type Message illustrated using the letter "i" (for information) in uuBML, indicating that no response is required.

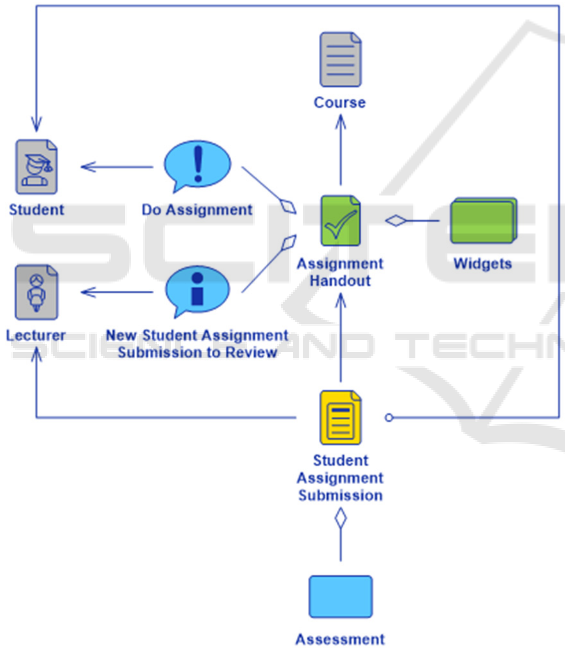
Course
Student
Do Assignment
Assignment Handout
Widgets
Lecturer
New Student Assignment Submission to Review
Student Assignment Submission
Assessment

Figure 2: Assignment Submission sub-system.

Table 1 and Table 2 list the life cycle states of the Assignment Handout artifact and the Student Assignment Submission artifact, respectively.

Table 1: Life cycle states of Assignment Handout artefact.

State	Description
Created	An empty assignment handout is created in the system.
Active	Assignment handout content is completed and the assignment is ready for students.
Assigned	Assignment handout is assigned to students.
Archived	All assignment submissions are marked. The assignment is archived - no further activities can take place.
Attention	There are minor problems with the artifact. (e.g. incomplete handout)
Problem	There are significant problems with the artifact. (i.e. major issue)
Cancelled	The artifact is cancelled.

The cardinality of the association between the Student artifact and the Student Assignment Submission artifact is one-to-one (arrow with a circle on one side), indicating that each Student submits a single Assignment Submission for a given assignment handout, and that each Assignment Submission belongs to one student. The cardinality of the association between the Assignment Handout and the Assignment Submission artifact is one-to-many (indicated by an arrow without a circle), i.e. each Assignment Handout can be associated with multiple Student Assignment Submissions, but each Student Assignment Submission belongs to a single student.

Table 2: Life cycle states of the Student Assignment Submission artifact.

State	Description
Submitted	The student submitted an assignment solution.
Accepted for review	The lecturer accepts the assignment solution for marking. Submitting a new version of the solution is not permitted.
Returned	The lecturer has returned the assignment submission to the student for re-submission.
Assessed	Assignment has been marked by the lecturer. Student can request additional feedback.
Archived	The assignment has been archived.
Cancelled	The artifact is cancelled.

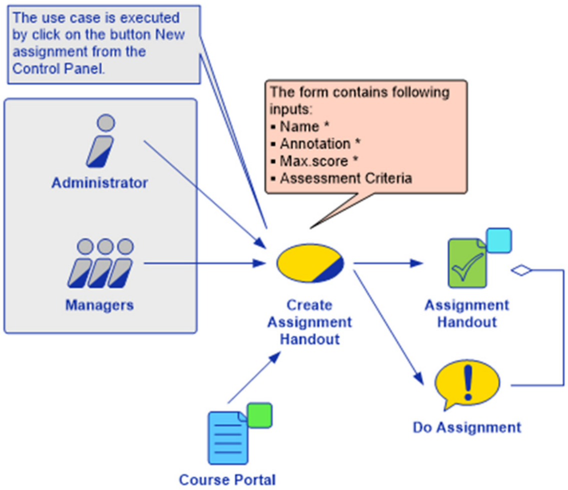
The use case is executed by click on the button New assignment from the Control Panel.
The form contains following inputs:
▪ Name *
▪ Annotation *
▪ Max.score *
▪ Assessment Criteria
Administrator
Managers
Create Assignment Handout
Assignment Handout
Do Assignment
Course Portal

Figure 3. An example of the business use case diagram.

As the users interact with the application via roles, submitted assignments can be easily re-assigned to another lecturer, if the need arises (e.g. if the lecturer is absent on leave). The artifacts contain widgets that provide the required functionality for the users. The widget for submitting assignment solutions and the widget for managing assignments and their solutions are micro applications based on HTML5, and can be implemented using different types of frameworks and technologies e.g. React JS (ReactCommunity, 2016), JQuery (JQuery, 2016), Sinatra (Sinatra, 2016), Ruby on Rails (RubyOnRails, 2016), etc.

Table 3: Mapping of business use cases to roles.

Business Use Case	Roles
Create an empty instance of assignment handout	Administrator/Manager
Modify assignment handout content	Administrator/Manager/ Lecturer
Close assignment	Administrator/Manager
Review and assess assignment submission	Lecturer
Display assignment handout	Administrator/Manager/Auditor/ Student/Lecturer
Display assignment submission	Administrator/Manager/Auditor/ Student/Lecturer

Process decomposition is used to identify business roles and business use cases within the main processes. The following roles were identified: Administrator, Manager, Auditor, Student, and Lecturer. Mapping of business use cases to roles is shown in Table 3. For each business use case, the specification contains a diagram and a detailed description that includes the artifact name, brief description, meta-artifact, basic business process flow, actors, preconditions, and post-conditions. An example of a Business Use Case diagram is shown in Figure 3. The diagram indicates that administrators and managers can execute the use case functionality. The use case form application has five inputs (Name, Annotation, Maximum Score and Assessment Criteria); asterisk indicates that the input is mandatory. The basic flow of the use case is shown on the right-hand side of the diagram. The structure and locations of the artifacts is described in the Organizational View section.

5 CONCLUSIONS

Traditional approaches to modeling of enterprise applications tend result in complex technical models. This makes it difficult for business professionals to understand the models and to communicate about the models with technical experts and various other stakeholders within the organizations. In this paper we have described the artifact-based Unicorn Universe Process method and illustrated the application of this method using a University Assignment application subsystem scenario, emphasizing the importance of modeling artifact life cycles.

A significant benefit of this approach is that the resulting uuProcess models map directly to executable applications supported by the Unicorn Universe Platform - a digital construction kit for the development of applications from reusable components. From the technical point of view, Unicorn Universe Platform is a framework for building enterprise SOA applications that facilitates the composition of SOA applications from individual services. Unicorn Universe Platform supports a range of standard reusable Platform Services, including JSON data storage, binary file storage, Inter-Process Messaging, Application Logging, User Access Management and many others. All services have a standard REST (REpresentational State Transfer) API (Application Programming Interface) and use JSON as the serialization data format.

The Unicorn Universe Process (uuProcess) method has been used successfully on hundreds of customer projects over the last decade as well as for the development of internal systems across the Unicorn organizations. Furthermore, the Unicorn

Universe Business Modeling Language has been used universally as a visual method for communication within the Unicorn organizations providing additional evidence about the advantages of the artifact-based approach.

REFERENCES

Cohn, D. & Hull, R. 2009. Business artifacts: A data-centric approach to modeling business operations and processes. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering,* 32**,** 3-9.

Hull, R. Artifact-centric business process models: Brief survey of research results and challenges. OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", 2008. Springer, 1152-1163.

Hull, R., Damaggio, E., Fournier, F., Gupta, M., Heath III, F. T., Hobson, S., Linehan, M., Maradugu, S., NIGAM, A. & Sukaviriya, P. Introducing the guard-stage-milestone approach for specifying business entity lifecycles. Int. Workshop on Web Services and Formal Methods, 2010. Springer, 1-24.

ITIL. 2014. *What is ITIL* [Online]. ITIL. Available: http://www.itil.org/en/vomkennen/itil/ueberblick/index.php [Accessed 2014].

Jquery. 2016. *JQuery,* [Online]. Available: https://jquery.com/ [Accessed 16 November 2016].

KökörčenÝ, M. & Kovář, V. 2015. Building Enterprise Applications Using Unicorn Universe Services. *In:* TOUMANI, F., ET AL. (ed.) *Service-Oriented Computing - ICSOC 2014 Workshops: WESOA; SeMaPS, RMSOC, KASA, ISC, FOR-MOVES, CCSA and Satellite Events, Paris, France, November 3-6, 2014, Revised Selected Papers.* Cham: Springer International Publishing.

Kovář, V. R. 2011. *Unicorn Method for Enterprise Systems.* PhD, University of Hradec Kralove.

Kruchten, P. 2004. *The rational unified process: an introduction*, Addison-Wesley Professional.

Kunchala, J., Yu, J. & Yongchareon, S. A survey on approaches to modeling artifact-centric business processes. International Conference on Web Information Systems Engineering, 2014. Springer, 117-132.

Lianying, Z., Jing, H. & Xinxing, Z. 2012. The project management maturity model and application based on PRINCE2. *Procedia Engineering,* 29**,** 3691-3697.

Manalil, J. 2011. Rational Unified Process.

Nigam, A. & Caswell, N. S. 2003. Business artifacts: An approach to operational specification. *IBM Systems Journal,* 42**,** 428-445.

Reactcommunity. 2016. *React JS* [Online]. Available: https://github.com/reactjs [Accessed 16 November 2016 2016].

Rubyonrails. 2016. *Ruby on Rails* [Online]. Available: http://rubyonrails.org/ [Accessed 16 November 2016 2016].

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. & Lorensen, W. E. 1991. *Object-oriented modeling and design*, Prentice-hall Englewood Cliffs, NJ.

Rumbaugh, J., Jacobson, I. & Booch, G. 2004. *Unified Modeling Language Reference Manual, The (2nd Edition)*, Pearson Higher Education.

Silver, B. & Richard, B. 2009. *BPMN method and style*, Cody-Cassidy Press Aptos.

Sinatra. 2016. *Sinatra* [Online]. Available: http://www.sinatrarb.com/ [Accessed 16 November 2016 2016].

Sutherland, J. 1995. Business objects in corporate information systems. *ACM Computing Surveys (CSUR),* 27**,** 274-276.

Unicorn. 2016a. Unicorn Universe Business Modelling Language. [Online]. Available: https://unicornuniverse.eu/en/uubml.html (accessed on 19th November 2016)

Unicorn. 2016b. uuBML Draw Documentation, version 1.0.0. [Online]. Available: https://www.plus4u.net, document code: VPH-BT:UUAPPKNH.UUBMLDRAW/DOC_1.0.0 (accessed on 20th November 2016)

Unicorn. 2016c. uuApp Architect – presentation – Main components. [online]. Available: https://www.plus4u.net, document code: UCL-BT:UUAA/PRE_2 (accessed on 20th November 2016)

Unicorn. 2016d. HowTo – High Level Concept [online]. Available: https://www.plus4u.net, document code: VPH-BT:44191583700503283 (accessed on 20th November 2016)

Unicorn College. 2016. Unicorn College Vision 2016. [online]. Available: https://www.plus4u.net, document code: UCL-BT:UCL/VISION2016 (accessed on 23th November 2016)

Unicorn Universe. 2016. Unicorn Universe Process. [Online]. Available: https://unicornuniverse.eu/en/uup.html (accessed on 17th November 2016)