When applications are business processes

An ArchiMate based explanation

By Bogumila Anna Sobolewska (S0170232)

Supervisor: Dr. L.J.P. Groenewegen

March 2011

Universiteit Leiden

Bachelor Advanced Computer Science
Leiden Institute of Advanced Computer Science
University of Leiden, The Netherlands

Abstract

In the ICT-intensive organizations, application as a core business becomes increasingly occurring phenomenon. In e-commerce or e-learning, online software converts into a leading business process. At the moment it's quite challenging to model this new behavior correctly, using existing modeling languages according to their standards. In this paper, we want to propose a new modeling approach based on ArchiMate. Using an example of online community, we managed to represent application in the ArchiMate's Business Layer. The adjustment goes through several stages, starting with the software in Application Layer, and moving it slowly to the upper level by introducing step-by-step improvements. The approach showed in this paper should not be seen as the ultimate way of application modeling, but it could be used in situation when one wants to emphasize the business view of the application, together with engaged stakeholders instead of simply model its technical functionality.

Keywords: ArchiMate, modeling, business process, application.

Table of contents

1 Introduction

This document contains the research result for the bachelor thesis at Leiden University. In the fall of 2010, while studying online communities as part of another research, we came across interesting discussion on locating an application at ArchiMate's business level. Going deeper into this topic within the master research, could create a risk of shifting the main focus from the core subject of the research. However the debate proved to be interesting enough to continue in relative isolation in the current research paper.

1.1 Problem description

Business process is the inherent and defining element of any organization. It's the base for any activities and its correct and optimal implementation has direct influence on the success or failure of the company. Contemporary businesses can't prosper with another key support, which is the IT. Hardware and software provide the automation which is no longer competitive advantage, but substantial for survival commodity. The initial support and cooperation of those two main pillars slowly transformed into mutual dependence and permeation, with applications becoming core business processes and custom software created according to the business process logics. This approach allowed for combining benefits of both different words to create competitive advantage for modern company.

Another important fact is, that to fully understand the organization, the processes running within it, the alignment of software and hardware and people around it, it's necessary to put all those elements on paper. Following the reasoning above, while it became difficult to find the strong border between business process and application, the question of putting them in particular architectural domain became a real problem statement and core for this discussion. We believe that in some cases it's more logical to model application as business process. This paper will try to examine the correctness of this statement.

1.2 Solution suggestion

To study the problem described above we decided to use an example of online application and model it within random enterprise environment.

There are several different modeling languages used to capture the structure of the organization, with the UML being one of the most popular. However, for this paper we decided to concentrate on ArchiMate. Firstly because this document is a result of the problem which arose during another research already based on ArchiMate, secondly we believe that the fact that ArchiMate supports the same logic and similar elements in each architectural domain (layer), it should be possible to model the same behavior regardless the layer.

In our approach we will try to originate with the application in ArchiMate's Application Layer, then step by step move it to the Business Layer, finally validating our hypothesis.

1.3 Research question

The main research question of this paper can be formulated as follows:

Main research question: Is it possible to represent application as business process in ArchiMate modeling language?

To answer the main research question, the following sub questions can be formulated:

Sub-question 1: **What are the characteristics, differences and similarities of business process and application?**

For better understanding of the problem, it's important to get familiar with the theoretic background and definitions of the key concepts. Establishing the characteristics, differences and similarities will form the bases for second sub-question and finally should help us to form the conclusion for the main question of this paper.

Sub-question 2: **To what extend and in which cases it might be advantageous to represent application as business process (and vice versa)?**

Answering this question should add some clear value to this paper. Finding the advantages of representing application as business process might make the paper interesting not only from the academic point of view, but might point us to actual business value supported by relevant business cases visualized in the chosen example.

1.4 Academic relevance

This paper should be interesting for both business and academic purposes. In case of business it should show the new way of application and business process modeling, in our believe more accurate for the rapidly changing ICT-intensive organizations.

In case of academic value, this research should start the discussion about the new ways of appliance for ArchiMate and possibly for many other modeling languages. If our assumption is correct, we would be able to show the thought-provoking flexibility and adaptation property of ArchiMate.

2 Related work and literature review

For better understanding of the discussion presented in this paper, we will start with literature review, defining the main concepts of business process and application process, looking at their characteristics, similarities and differences.

2.1 Business process

Organizations are defined by their business processes. To understand what an organization does, one needs to understand its processes. In a very real sense, improving that organization—making it more responsive to change, more profitable, and more valuable—means improving its business processes.

Business process, its modeling and reengineering is a very popular topic of countless number of publications, starting form XVIII century with description of pin production written by Adam Smith.

Let us first look at the definitions of business process given by Thomas Davenport and Rummler & Brache in the early 90's.

Business process is:

"a structured, measured set of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how work is done within an organization, in contrast to a product focus's emphasis on what. A process is thus a specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs: a structure for action. ... Taking a process approach implies adopting the customer's point of view. Processes are the structure by which an organization does what is necessary to produce value for its customers." Thomas Davenport (1993) [3]

"a series of steps designed to produce a product or service. Most processes (...) are cross-functional, spanning the 'white space' between the boxes on the organization chart. Some processes result in a product or service that is received by an organization's external customer. We call these primary processes. Other processes produce products that are invisible to the external customer but essential to the effective management of the business. We call these support processes." Rummler & Brache (1995) [8]

Looking at the definitions above, we can summarize the following characteristics of the business process:

- Definability: it must be possible to clearly define the system, its boundaries, input and output.
- Direction: it must consist of activities that are ordered according to their position in time and space.
- Customer: there must be a beneficiary of the process' outcome, a customer.
- Value-creation: the goal of the business process is to create the customer value.
- Embeddedness: a business process can't exist in itself; it must be embedded in an organizational structure, or as sub-process in another process.
- Cross-functionality: a business process usually can, but not necessarily must, extent to several functions.

2.2 Application

Application, often referred to as "Software", is a program or group of programs designed for end users. It can be divided into two general classes:

- systems software - consists of low-level programs that interact with the computer at a very basic level. This includes operating systems, compilers, and utilities for managing computer resources.
- application software - includes database programs, word processors, and spreadsheets. It resides above systems software, because it is unable to run without the operating system and system utilities.

There are many types of applications: application suite, enterprise software, enterprise infrastructure software, educational software, simulation software, and mobile applications product engineering software, each having its particular characteristics and appliance. In this paper we are mainly concentrating on enterprise software and so called process-driven application.

Enterprise software, also known as enterprise application software (EAS), is software used in organizations, such as in a business or government, as opposed to software chosen by individuals (for example, retail software). Services provided by enterprise software are typically business-oriented tools such as online shopping and online payment processing, interactive product catalogue, automated billing systems, security, content management, IT service management, customer relationship management, resource planning, business intelligence, HR management, manufacturing, application integration, and forms automation. Enterprise level software aims to improve the enterprise's productivity and efficiency by providing business logic support functionality.

A process-driven application is a software application that is driven by an underlying process engine where the process can be exposed and reused. In effect all applications are process-driven and the logic of any application can be extrapolated into a flowchart to represent the logical process of execution. Process-driven applications are a growing trend in enterprise solutions involving humans, systems or both.

Putting the definitions of enterprise application software, process-driven application against the one of the business process, we can come up with the following characteristics, already well known from the previous section:

- Definability: it must be possible to clearly define the system, its boundaries, input and output. Programming the applications is univocal with clear definition of functionality, limitations, as well as input and output variables.
- Direction: it must consist of activities that are ordered according to their position in time and space. Like a flowchart which can be derived for each application, showing the logical process of execution.
- Customer: there must be a beneficiary of the process' outcome, a customer or in case of application any end-user of the software, employee of the organization or the client.
- Value-creation: the goal of the application is to improve the enterprise's productivity and efficiency and support the key business processes, all that contributes to costs savings and additional value creation.
- Embeddedness: application software can be embedded in an organizational structure, business process, or as sub-application in another system.
- Cross-functionality: applications can extent to several functions, departments, systems. For example, within organization, applications like CRM, SAP can be used by HR, Sales, Marketing, etc.

Looking at the above characteristics, we can initially state that there is a great resemblance between business process and application software. This conclusion is crucial to further discussion in this paper.

2.3 Business process vs. application

In the chapters above we looked at the definitions of business process and application and looked into similarities in their characteristics. In this section we will concentrate on differences, displayed in Table 1.

Business Process	Application
Performance human dependent. People are the core part of business process; their performance can be easily influenced through health, skills, free decision making, and other impacts from environment.	Performance system dependent. Although people operate the applications, the performance of the software is dependent on the stable code and system below, which are more robust, steady and environment independent.
Long installation process. Requires training manuals, long transfer and adaptation period, which can vary per location making global setup extremely challenging.	Relatively short and large-scale installation. New software can be easily set up across the departments or even international offices.
Relatively low flexibility, requires additional training and good communication with people involved.	High flexibility, rather easy to introduce changes on global scale.
High customization and localization. Business process can be adapted to fit best in the given environment (country, resources, culture), allowing for experiments and development of best practices.	If implemented on large scale, customization in separate locations is limited for better performance and maintainability, which makes the application locally less flexible.
Open process, constantly under management and monitoring, which increases the possibility to discover flows and errors, even by non-experts.	Black box, once tested the correctness is assumed. Difficult to discover the hidden errors, debugging and recovery can be done only by experts. See Moody's case [8].

Table 1: Business process vs Application

2.4 Application as business process.

In the age of rapidly growing technology it's hard to imagine any well prospering business without any IT support. Even businesses whose products and services have nothing to do with computers, are nearly always at least supported by any sort of software and hardware. In this sense applications support business processes, or even become their main component.

In the previous chapters we looked at business process and application. After giving the general definitions, we looked at their similarities and differences. Similarities will enable us to combine business process with application; the differences will be used as a competitive advantage.

One of the biggest advantages of application and business process cooperation is IT-enabled business process innovation. Thanks to powerful tools like EPR and CRM, companies are able to quickly perform changes

throughout their organizations. In a 2008 Harvard Business Review article [7], Andrew McAfee and Erik Brynjolfsson describe the main advantages of IT-based business process innovation:

- Because software can be easily installed throughout an organization, an improved business process can be quickly replicated in every store, every factory, or every office.
- A software-based process is self-enforcing. Rather than relying on, say, training manuals and hoping that employees follow a new process, the software itself carries out the process in a consistent way.
- The results are immediate, produced as soon as the new application goes live, shorting the long transfer and adaptation period, training of the employees.
- Because the business process is application based, it can be executed exactly the same way everywhere and every time in the organization.

According to McAfee and Brynjolfsson, the ability to embed a better business process in software, then replicate it reliably across an organization is a primary way to create competitive advantage with information technology. However, achieving this kind of unique advantage from commercial, off-the-shelf software is hard. Almost by definition, packaged software implements well-understood, common processes. And since the same package is available to everybody, differentiating the organization with ready software is hardly possible. Unique processes require custom, homemade applications.

On the other hand, using in-house application as a base for core business process brings with it serious risks, as any error in implementation, which in case of software is much more common comparing to business process, can have huge impact on the whole organization. To illustrate this issue, let's look at the case of Moody's [4]. In May 2008, the bond rating agency Moody's disclosed that it had incorrectly assigned AAA rating to billions of dollars in debt products. The error led some of Moody's customers to invest in products that were significantly riskier than they expected.
It also put Moody's on front-page coverage in the Financial Times, and seriously damaged the company's reputation. The incorrect ratings weren't the result of an analyst making poor judgments on these products, but they were caused by an error in the custom application that Moody's used to model risk. At the point when core business process becomes so highly dependent on software, it's crucial to assure the absolute correctness of the application, and minimalize the consequences of black box issue mentioned in Table 1.

As we talked before about the advantages of the application which can be used to improve the business process, now it's time to look the other way around, at the benefits of treating application as a business process.

One of the biggest advantages is at the same time solution to the application correctness problem described above. We are talking about application lifecycle management (ALM), where software development and maintenance is itself a critical business process.

Application Lifecycle Management (ALM) is a continuous process of managing the life of an application through governance, development and maintenance. It's using business process management techniques in software engineering to facilitate and integrate requirements management, architecture, coding, testing, tracking, and release management [2].

The lifecycle begins with an idea of an application. Once the application is created, the next big step is deployment, when the application goes into production. Finally, when it no longer has business value, the application reaches end of life and is removed from service. Governance, which encompasses all of the decision making and project management of the application, extends over this entire time. This approach has multiple advantages like: increased efficiency, correctness and quality, improved resource utilization and maintenance, however we are not going to delve into this particular subject. The most important is the fact that those advantages are achieved by treating application development and operation as business process.

3 ArchiMate

In the current age dominated by internet and fast developing technology, organizations have to learn to quickly react and adjust to new market requirements and business goals. The changes have often influence on many different levels, from technology, to business process, services, product portfolio, manufacturing, employees etc. To capture and control the impact of those changes, good understanding of the components and dependences between them is necessary. Clear picture can be created using integrated architecture that would describe and visualize the different domains and their relations.

There are several successful modeling languages for designing organizational structure, business process, information systems or infrastructure. They might be very helpful to model particular domain, but they all use their own language, techniques and tools, and can't be combined in one clear picture revealing important links and dependencies between domains [Figure 1].

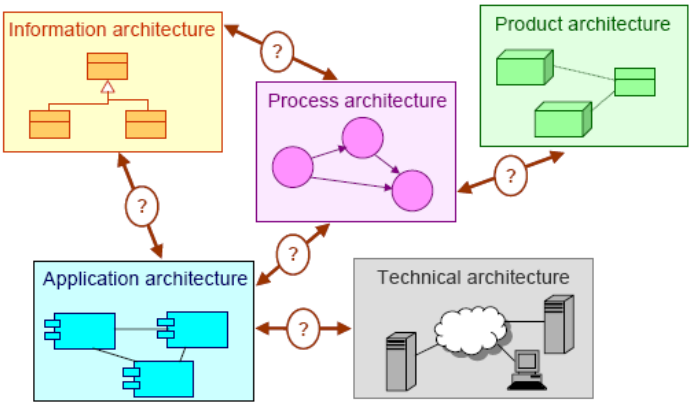

Figure 1: Heterogeneous architectual domains.

The ideal language for modeling enterprise architecture should have the following characteristics.
It should be:

- able to model the global structure within each domain, illustrating all the components and their dependencies
- able to visualize the relations between domains
- easy to understand by experts and non-experts across all domains
- possible to visualize models in a different way (from a different point of view), tailored towards specific stakeholders with specific information requirements.

Keeping those requirements in mind, in 2002 Telematica Instituut started the ArchiMate project. Together with Ordina, the Radboud Universiteit Nijmegen, the Leiden Institute for Advanced Computer Science (LIACS) and the Centre for Mathematics and Computer Science (CWI) they have started intensive research. Through the years the results have been validated and applied in practice with ArchiMate business partners: ABN AMRO, the Dutch Tax and Customs Administration (Belastingdienst), and the ABP pension fund. Finally in February 2009 the ArchiMate Version 1.0 Technical Standard has been formally approved by the Board of the Open Group.

As mentioned above (section 1.2) there were two main reasons why we decided to use ArchiMate for this research:

- this document tries to answer the question which ascended during another research already based on ArchiMate,
- it is guessed that the fact that ArchiMate supports the same logic and similar elements in each layer, it should be possible to model the same behavior regardless the layer.

3.1 Core concepts of ArchiMate

The leading concept of the ArchiMate modeling language is service. The researchers concluded that "service orientation supports current trends such as the service based network economy and ICT integration with the Web services."

The services are defined as follows: "unit of functionality that some entity (e.g., a system, organization or department) makes available to its environment, and which has some value for certain entities in the environment". The service in ArchiMate has a very wide meaning; it doesn't restrict itself to service provided by the companies to their customers, but can also be a service delivered by application to business processes or by technological facilities (e.g., communication networks) to applications.

The ArchiMate has a layered structure, distinguishing different domains within the organization, where the lower layers provide services to higher levels.

There are 3 main layers:

1. Business Layer - consists of business actors (for example employees of the organization) taking part in business processes to provide products and services to external customers.
2. Application Layer - is built of software applications providing supporting application services to the business layer.
3. Technology Layer – provides infrastructural services (processing, storage and communication services) which are needed by the application layer. It consists of devices, hardware and system software.

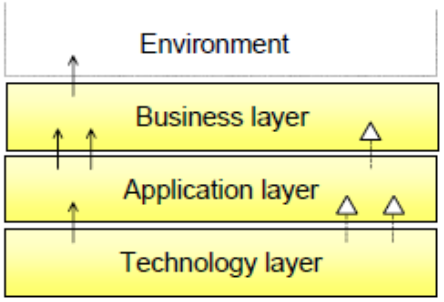

Figure 2: ArchiMate Layers

Mentioned above service orientation of this modeling language is expressed in the use of relations. There are 2 important relations between the layers:

- use relation – visualizes how the higher layers make use of the services of lower layers
- realization relation – shows how elements in lower layers may realize items in higher layer

The structure, types of concept and applicable relations within the layers are pretty generic, which makes this language easier to use and understand by experts in each domain; however their particular nature and details differ.

The figure below shows the modeling elements of ArchiMate and the relations between them. We can clearly see which elements belong to which of the three layers and which relationships are possible between items in one layer and between layers.

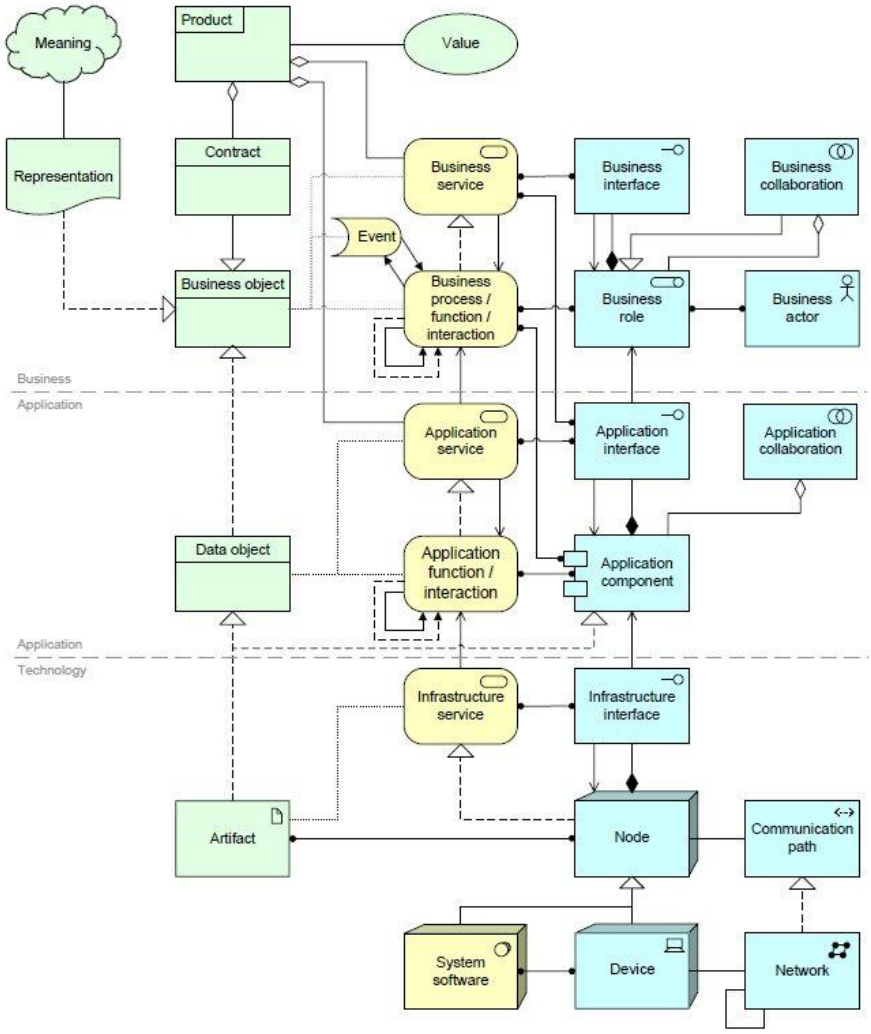

Figure 3: Overview of the main concepts and relationships

3.1.1 Business Layer

Business layer describes the business processes and people around them called business actors. Business actor can be an individual person (staff member, client) or a group of people and resources. Each actor has an assigned business role. It separates actor from the business activity and makes it easier to allocate activity to him. The actors use *organizational services* realized by *business process* (usually divided in several *sub-processes*).

Services are grouped to form (financial or information) products. The product or service has also associated *value*, often expressed in terms of money, but non-monetary value is also essential to business, for example, practical or functional value and the value of information or knowledge.

3.1.2 Application Layer

The structural entity in the application layer is modeled using the *application component*. It can be a software element that is a part of one or more applications, but also complete software application, sub-application or information system (CRM, CMS, website, financial application).

Similarly to business layer, the service orientation is continued by *application services* provided to business layer above. An *application function* describes the internal behavior of a component needed to realize one or more application services.

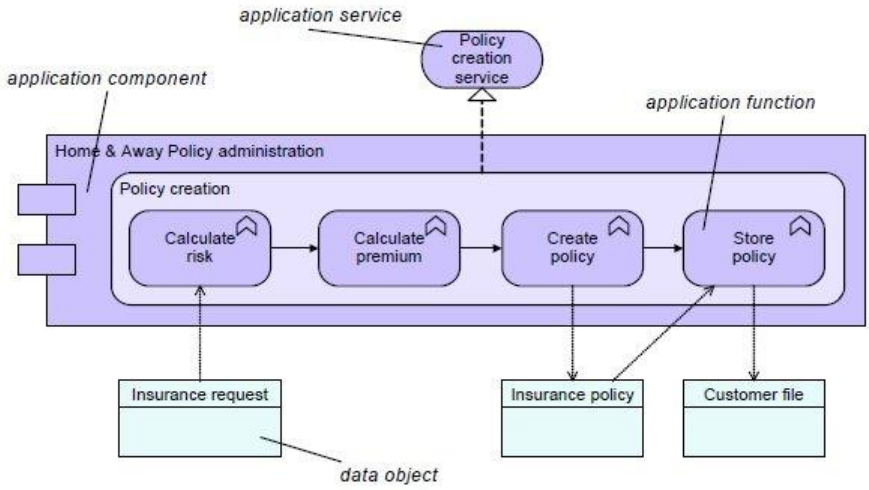

Figure 4: Example of application layer model.

The figure above shows an example of application layer model. The application component is composed with the application functions activating one each other (which is modeled with triggering relation), forming Policy creation process. Application functions create data objects which can relate to business objects in the layer above. The most important role of the application however is the application service provided to business level.

3.1.3 Technology Layer

Node is the main structural concept for the technology layer. There are two types of node: *device* (mainframe, desktop) and *system software* (mail server, database).

The relation between nodes is modeled by communication path, through which nodes can exchange information. The physical realization of a communication path is a modeled with a *network*.

Correspondingly to business and application, also the technology layer provides external service called *infrastructure service*. In this layer it is however chosen not to model the internal behavior of infrastructure

components such as routers or database servers; that would add a level of detail that is not useful at the enterprise level of abstraction.

3.1.4 Relations

One of the main goals of ArchiMate language is to model the relations between and within different domains. In each of the layers described above, different relations between elements can be applied:

- The *access relation* models the access of business or data objects, by processes, functions or interactions.
- The *use relation* models the use of services by processes, functions or interactions, or the use of interfaces by roles, components or collaborations.
- The *composition relation* indicates that an object consists of a number of other objects.
- The *aggregation relation* indicates that an object groups a number of other.
- The *assignment relation* links units of behavior with active elements (e.g. roles with actors that fulfill them)
- The *association relation* models a relation between objects that is not covered by another, more specific relation.
- The *realization relation* links a logical entity with a more concrete entity that realizes it.
- The *specialization relation* indicates that an object is a specialization of another object.
- The *triggering relation* describes the temporal or causal relations between processes, function, interactions and events.

4 Modeling application in business layer

To analyze the correctness of my assumption we decided to use ArchiMate to model an online community, even though the same reasoning can be applied to many other systems. The choice comes from the fact that online community is a quite complicated entity in itself. It's people who interact, ask, read and answer the questions, forming sort of support group. It's also business service provided by the organization which supports it financially and sustains its existence. In the end it's also web application, platform which technically enables all of the above activities. That's why modeling online community only as a business process or only as an application, doesn't give its true picture and surely doesn't disclose all its details. The multiple layers of ArchiMate can give a fuller view of the whole system and people around it.

Also, as mentioned above, online community can be defined as group of people, business process, or web application. Keeping that in mind we could assume it should be possible to place online community in Application as well as in Business Layer of ArchiMate.

In the sections below we will start from placing the online community in application layer. By modifying and improving the model according to ArchiMate standards we will progressively move the online community part

from application to business layer, at the same time changing the role of the online community from software to business process.

4.1 Online community

ArchiMate gives the possibility to describe the same system from different points of view, concentrating on aspects which are particularly interesting and need to be described in more detail.

The models below describe the online community, but concentrate on content:

- content creation, which might include submitting and answering the questions, forum discussions, commenting and rating of the website's content
- content consumption, which comprise reading of forum, comments or Q&A section, without contributing.

Describing this aspect of community gives us the opportunity to look at the different groups of stakeholders and their roles within the online community, as well as an important subject of collaboration, enabled by technology but extremely important for business process of the organization.

Model 1: Online community in Application Layer.

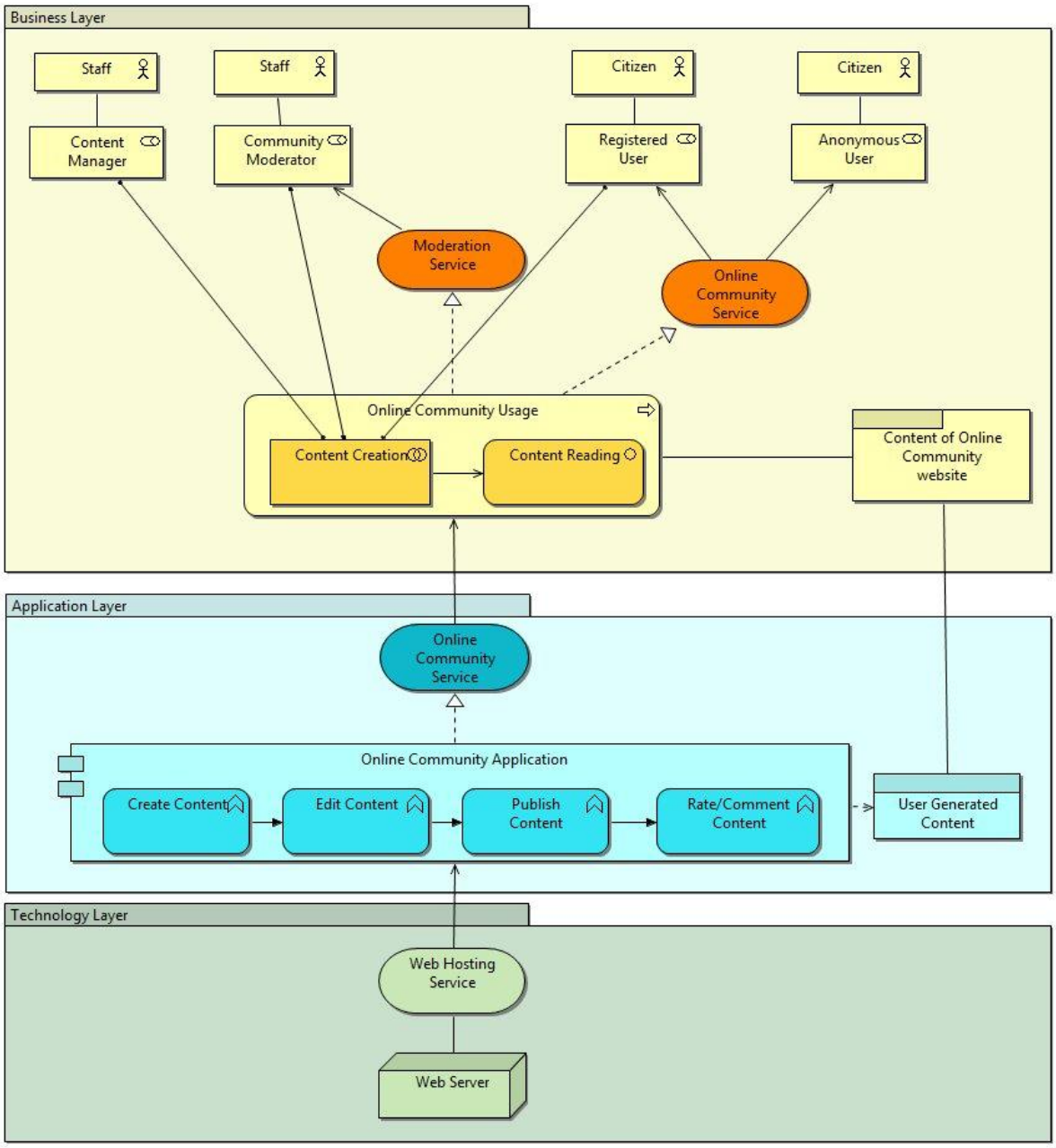

Figure 5: Online community in Application Layer.

Description

Starting from the bottom, the Technology Layer consists of the Web Server modeled in ArchiMate as node, more accurately as device. Web Server hosts the online community website and gives the Web Hosting Service to the Technology layer above. Because this layer is not particularly important for our discussion, we chose to keep its modeling representation simple. It minimize the level of detail that is not useful at the enterprise level of abstraction [6].

In this model online community is placed in Application Layer and is represented as web application. Its construction is based on an example from Figure 4.
It has been chosen to highlight the important functions of online community, which are crucial for the business process and stakeholders described in the layer above. Within Online Community Application element, four functions of this application are modeled:

- Create Content– enables users to create and save content, for example questions, forum, or blog posts
- Edit Content – enables users to edit earlier created content
- Publish content – enables authorized users to publish the final version of the content, making it available online for all the members of online community and website visitors
- Rate/Comment Content – enables users to rate the content and comment on it.

The data object of this application is User Generated Content, which is the content of the online community website created and published by the authorized users.

Online Community Application provides service to business layer, described as Online Community Service, which again shows the service-oriented structure of ArchiMate. It includes the application functionality provided by online community as web based software.

Finally the business layer models the business process and involved stakeholders. In this model there is only one business process described as Online Community Usage. It's simple process consisting of two elements:

- Content Creation – including writing, editing and rating of the content. It's modeled as business collaboration, to highlight the important notion of cooperation between different groups of stakeholders. In ArchiMate, the stakeholders are represented by so called actors and in our example include: Content Manager, Community Moderator and Registered User.
- Content Reading – passive usage of the content, which is represented as action.

The product of this process is the Content of Online Community Website, associated to User Generated Content data object from the Application Layer.

The business process itself provides two services:

- Moderation service – fulfilled by Community Moderator, including tasks like: stimulation of online discussion, content monitoring and validation, content creation etc.
- Online Community Service – service given by the community as a group of collaborating people, including for example: finding the answers to questions, reading posts, discussions and recommendations, asking questions online, starting own discussions etc.

The final elements of business layers are stakeholders, who are directly involved in the business process. In our example they belong to 2 groups of actors:

- Staff – employees of the organization, here represented by two roles: Content Manager concentrating on pure content creation, and Community Moderator responsible for Moderation Service mentioned above.
- Citizen – users of the website, here characterized by two roles: Registered User, with ability to login to the website, having wider access and more rights enabling him/her to have an active role within the community, and Anonymous User with restricted access and passive use of the online community.

Shortcomings

Looking at the model above, particularly at the Business layer we will surely agree that business process is highly simplified and therefore incomplete. Content Creation, described correctly as collaboration, consists of many different activities performed by different roles, which is not visible in the current version of this model. From business process point of view it's important to distinguish those particular activities and emphasize the different kinds of roles which each particular actor holds within this collaboration.
To address this particular problem, Model 2 has been created.

Model 2: Online community in Application Layer and as business process.

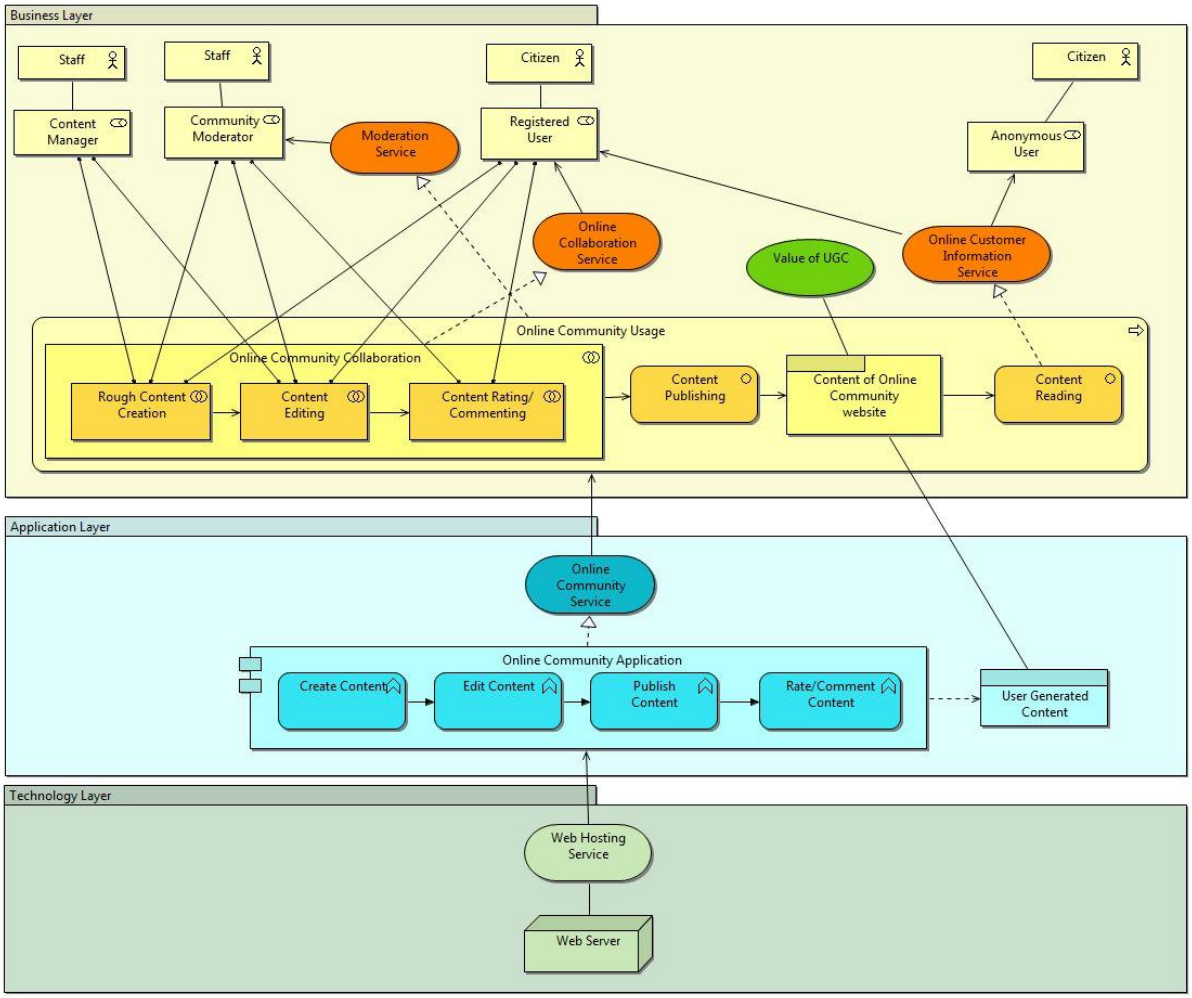

Figure 6: Online community in Application Layer and as business process.

Improvements

Concentrating on the shortcomings of the previous model, we leave the Technology and Application Layer unchanged and focus on improvements to the Business Layer. To achieve that we divide the Online Community Usage process into Online Community Collaboration and two activities: Content Publishing and Content Reading.

Let's look first at the Online Community Collaboration, which models the interactive and cooperative aspect of online community. It's divided into three different sub activities inheriting the collaboration form:

- Rough Content Creation – performed by Content Manager, Community Moderator and Registered User. It includes creating articles, starting and actively joining discussions, asking and answering questions etc. It's a work in progress, rough content created in the back-end, not officially published and hidden from the public.

- Content Editing – as above, performed by Content Manager, Community Moderator and Registered User. It consists of editing own content, but also in case of Community Moderator editing or removing low quality or inappropriate content of the others to assure high standards of the generated text.
- Content Rating/Commenting – performed mainly by Registered User with support of Community Moderator. It involves writing the comments on the existing content or using the recommendation or rating function to judge the content.

As we can see, the new representation allows us to assign different stakeholders to more refined activities within the online community, giving much clearer documentation of the business process. The Online Community Collaboration provides the Online Collaboration Service which is available for Registered Users. It grants them the possibility to take advantage of wide, pro-active character of the online community.

The next step in Online Community Usage business process is Content Publishing. It describes the action of making the final version of the content available to all visitors of the website. Through this activity the Content of Online Community website is provided as a ready product for the public use. This product has a value represented as Value of UGC (User generated Content). As stated in ArchiMate Language Primer [2] "the value of a product or service is what makes some party appreciate it. Value is often expressed in terms of money, but nonmonetary value is also essential to business, for example, practical or functional value, and the value of information or knowledge." In our case we are talking about the value of information represented by UGC, described in even more detail in the master thesis "Business Value of Online Community in public sector" [9].

The Content of Online Community website is provided as a source for the final activity which is Content Reading, which has much more passive form of use. With Online Customer Information Service it gives the Registered and Anonymous users the possibility to read the content in search for information, showing pure informative role of online community.

With the improvements described above we achieved much more detailed description of the business process. Crucial online community activities being part of this process has been extracted and assigned to different stakeholders. The whole business process has been divided in two parts, showing the interactive and cooperative aspect as well as the static, informative one.

Yet another important fact should be noticed. While in the previous model the online community and its functionality was represented only as a web application and modeled in Application Layer, Model 2 shows online community in the form of human cooperation in the Business Layer. With that, we just made the first step in representing application as a business process, which we will continue in the following models, to prove the assumption of this paper.

Shortcomings

The application of online community is described in four functions, but in the model above it provides only one general Online Community Service to the Business Layer above. As the business process in the Business Layer has been extended to provide more detail, it's now possible to align the application functionality with the business process activities. To achieve that, additional application services have to be derived to create the linking between those two layers. Keeping that in mind, we will continue the redevelopment of the model.

Model 3: Online community in Application Layer (multiple services representation) and as business process.

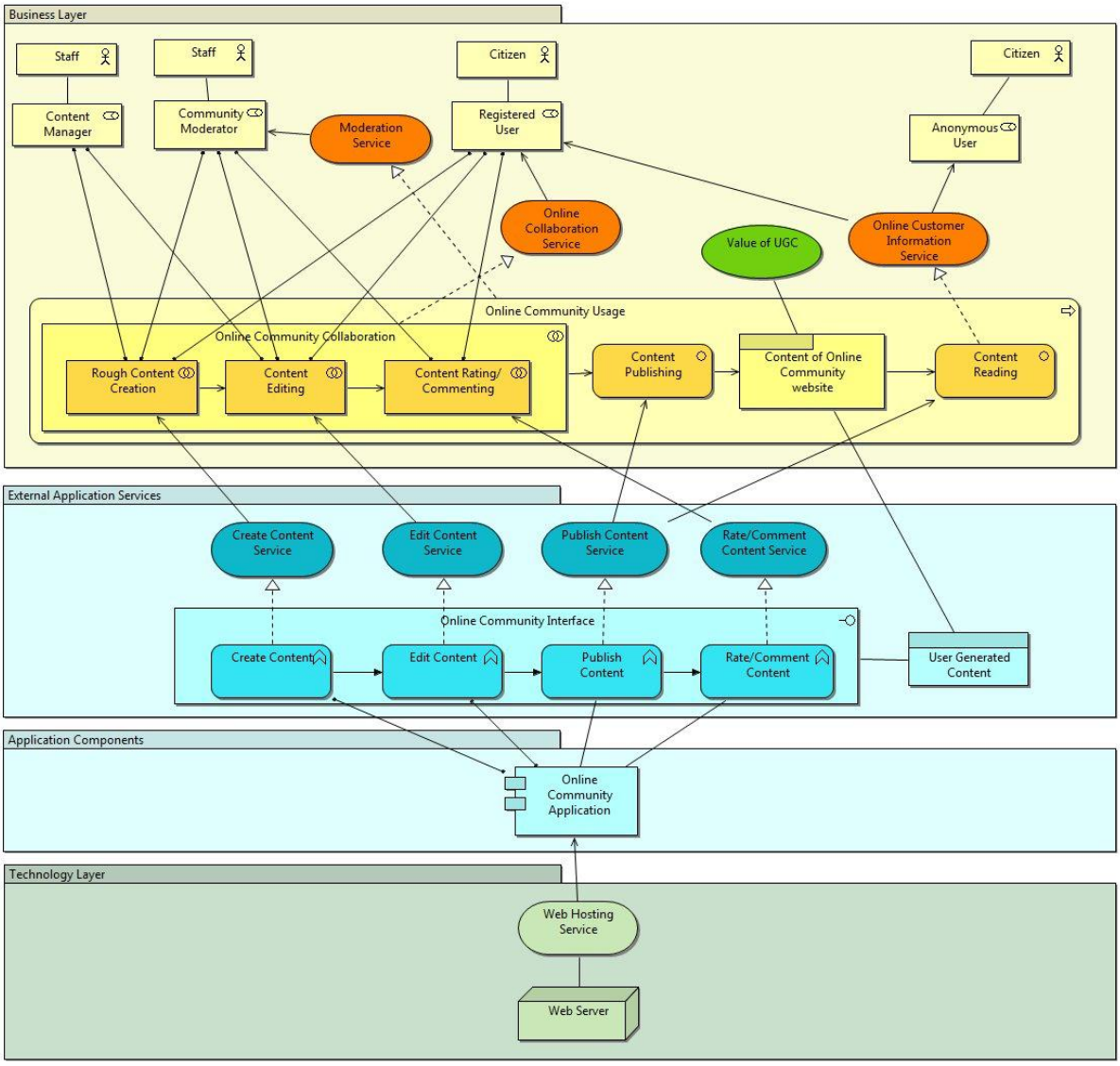

Figure 7: Online community in Application Layer (multiple services representation) and as business process.

Improvements

In the ArchiMate Language Primer [6] we read that "application interface is the (logical) location where the services of a component can be accessed. In a broader sense (as used in, among others, the UML definition), an application interface also has some behavioral characteristics: it defines the set of operations and events that are provided by the component, or those that are required from the environment." The four functions of the application fit the description above as provided by the application component and directly required by the environment, in our case stakeholder. For this reason those functions were shifted from application to the interface representation.

For each of those functions, separate services were assigned:

- Create Content Service
- Edit Content Service
- Publish Content Service
- Rate/Comment Content Service

Those services are provided to each activity (collaboration or action) within the business process, creating the linkage between application and business layer and aligning the operations of the interface with related elements of the business process.

ArchiMate gives us the option to make a distinction between the externally visible behavior of application components in terms of application services, and the internal behavior of these components to realize these services [6]. Following this concept we divided Application Layer into two sub-layers:

- External Application Services - grouping the online community interface and services
- Application components – containing single online community application

Looking at the ArchiMate View in Figure 8 achieved from applying slight modifications to the model in Figure 7, we can clearly see that there is a nearly 1:1:1 alignment between functions of the application, services of the application and activities in the business process. We can talk about the mimicking of application functions and services by business process activities. From that we can see that the function of application can be modeled within business layer and absorbed by business process, proving the theory stated in this paper.

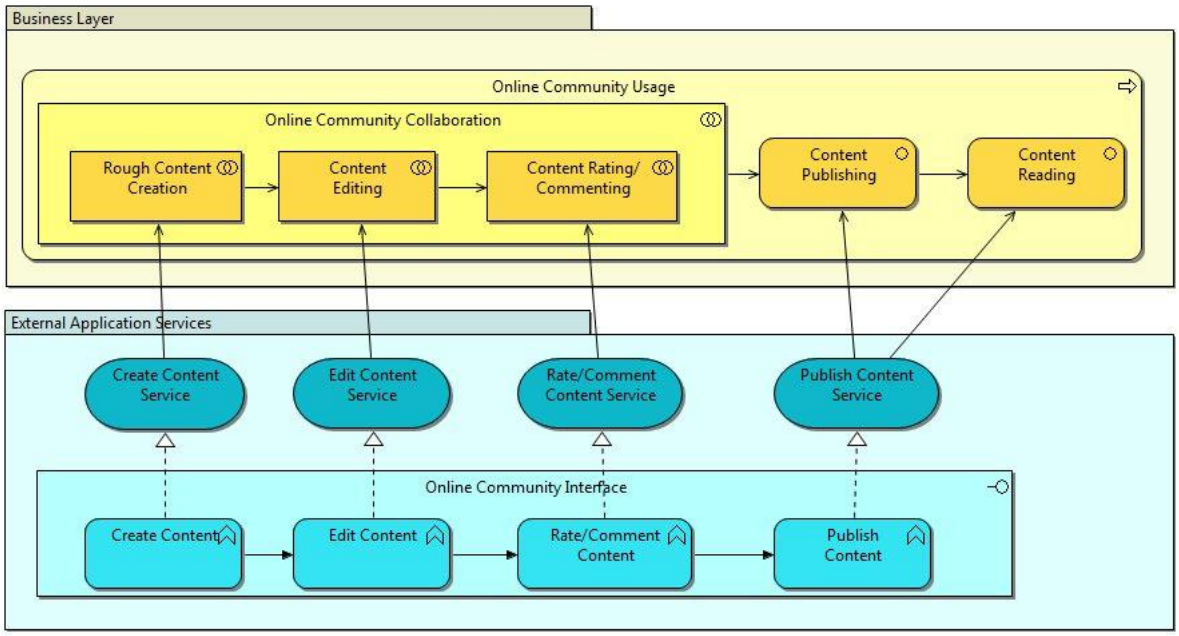

Figure 8: The mimicking of application functions and services by business process activities.

Shortcomings

Modeling the online community in the way presented in Model 3 [Figure 7], makes the system too repetitive and some of its parts like for example external Application Services seem redundant. We will try to make the final model as optimal as possible.

Model 4: Online community in Business Layer.

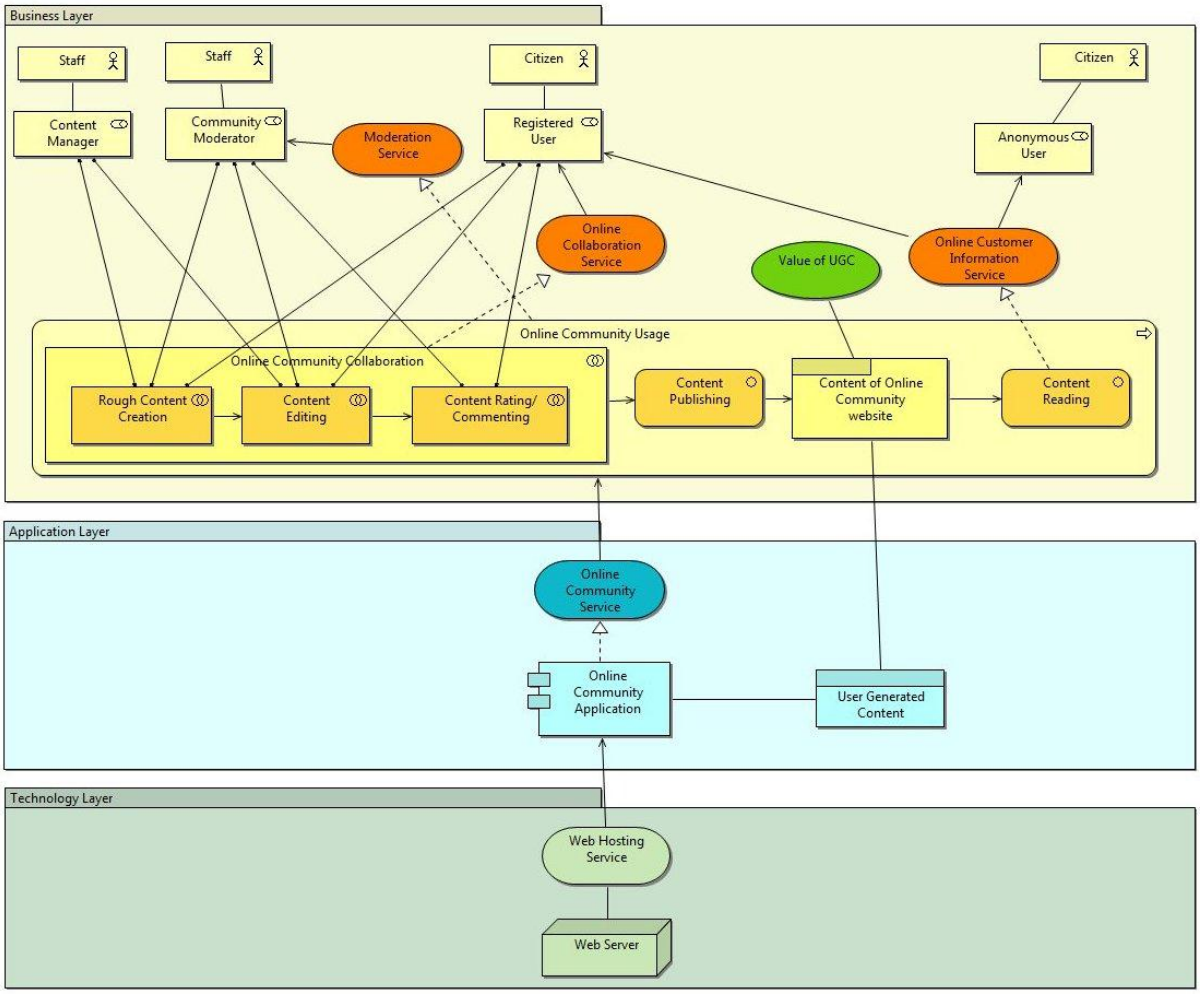

Figure 9: Online community in Business Layer.

In the final version of our model we decided to discard the External Application Services layer, going back to initial Application Layer consisting of Online Community Application, User Generated Content data object and single Online Community Service. Making this final step we completed the transfer of the application functionality to business process.

This representation concentrates on actors and their specific role within the business process. It suggests that online community is more importantly the business process and people around it, than only a web application.

5 Conclusions and further research

In this chapter we will summarize the findings from the chapters above and try to form the final conclusions. We will also point to topics which are still open and interesting to research in another paper.

5.1 Conclusions

The goal of this research was to examine if it's possible to model application as a business process in ArchiMate modeling language. Because the ICT-intensive organizations continue choosing applications as they core business processes, the problem presented in this paper is very actual and interesting not only from the academic point of view but also for the business science.

Application as business process provides number of advantages: IT-enabled business process innovation, quick and large scale process roll-out, robustness and many more described in detail in section 2.3.

In chapter 4 we went through several modeling stages, finally succeeding in moving online community from Application Layer to Business Layer. Doing that, we changed the meaning of the online community from being support software to business process involving different stakeholders. We are not trying to say that core business applications should be always modeled in business layer. However, this technique might be applied when the business value of the application and the relations between stakeholders engaged in this process are more important than the software's functionality.

ArchiMate proved to be a very suitable modeling tool to illustrate this problem. The property of ArchiMate which allows using the same logic and similar elements in each layer and model the same behavior regardless the layer, made it possible to achieve 1:1 relation between the business process elements and the functionality of the software, visualized in Figure 8. This discovery let us conclude that it's possible to model online community in application as well as in business layer.

In this paper, we chose to model online community, however, based on our research we strongly believe that the same modeling technique could be used for other applications, for example for e-commerce (online shops) or e-learning.

5.2 Further research

As mentioned above, we believe it should be possible to apply similar modeling techniques for e-commerce or e-learning applications. It would be interesting to examine if it's also possible and relevant (from business point of view) for software different than online application or web 2.0 type.

Also it might be interesting to research if modeling application as business process can be achieved in other modeling languages like UML, for example. Even though UML was developed for modeling of software-intensive systems, it doesn't mean we can't use it to our liking :-).

References

1. D. Chappell (2008): Application Lifecycle Management as business process, Whitepaper

2. D. Chappell (2008): What is application lifecycle management?, Whitepaper

3. T. Davenport (1993): Process Innovation: Reengineering work through information technology. Harvard Business School Press, Boston

4. S. Jones (2008): FT Alphaville exclusive: Moody's error gave top ratings to debt products

5. M. Kirchmer (1999): Business process oriented implementation of standard software, page 25-26

6. M. Lankhorst (2004): ArchiMate Language Primer

7. A. McAfee, E. Brynjolfsson (2008): "Investing in the IT That Makes a Competitive Difference", Harvard Business Review

8. Rummler, Brache (1995): Improving Performance: How to manage the white space on the organizational chart. Jossey-Bass, San Francisco

9. B. Sobolewska (2011): Business value of online communities in public sector: ArchiMate based approach