

A Flexible, Process-Aware Contract Management System

Miroslav Zarić*, Zoran Miškov**, Goran Sladić*

* University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia

** PD Elektrovodina doo, Novi Sad, Serbia

miroslavzari@uns.ac.rs, Zoran.Miskov@ev.rs, sladicg@uns.ac.rs

Abstract — this paper presents an open source based, general purpose contract management system. Contract management systems are becoming increasingly important component of business application landscape, aimed at improving the management of complete contract lifecycle. Contract management systems may be viewed as specialized document management systems (DMS). Although such systems can exist and be functional without implemented business workflow, for contract management this process perspective is quite important in order to have time-efficient, yet reliable contract handling. Hence, most of the contract management systems tend to deploy, or even impose, some business workflow. The system described in this paper is developing around an open source business process management system at its core, allowing for flexible contracts management workflows, and creating an extensible platform for customization to different needs.

I. INTRODUCTION

In recent years, one of the trends in development of business applications is a focus on electronic document management (EDM) systems. EDM system deals with management of documents [1]. In business entities, different documents are often created as a result of some activities in business process or are an essential requirement for fulfilling some specific task. During the business process execution, different individuals undertake activities according to a specified sequence prescribed by some workflow. Workflow Management Coalition defines workflow as the computerized facilitation or automation of a business process, in whole or in part [2]. EDM are usually implemented as document-centric workflow management systems [3].

Any business entity has some contract management procedures in place, often involving a large amount of manual processing. In today's modern and dynamic markets, there is a growing need for time-efficient contract creation and later contract lifecycle management.

Usual lifecycle of the contract involves several phases. In initial phases, legal department is drafting a contract, making sure that the content of the paragraphs is legally clear. After this verification, contract proposal is negotiated with the client, subject to changes. Each change also needs a seal of approval from the legal department. After the contract is negotiated and comes into force, devoted contract management requires further tracking of contract implementation and contract expiration.

However, the need for prompt contract creation is often confronted by the even greater need for legal clarity and content disambiguation in order to protect business entities from wrongful legal claims. Due to time-constraints, manual processing is becoming less acceptable, and prone to errors, resulting in prolonged contract negotiations. Defining the specific contract creation and contract lifecycle management workflows can be beneficial for overall business activity of an entity. IT support for such lifecycle gave rise to specialized contract management systems. In essence, they represent document management system supported by some well-defined workflow.

Contract management is a sensitive issue, since contract content can directly bring benefits for a business entity but can also expose it to unwanted circumstances. Although standard procedures for contract creation and approval exist in any business entity, creating applicative support for them is challenging. Large companies can have enough IT resources to develop customized solutions. Most in-house solutions are created for specific contract types and don't cope well with contract type changes. Medium size and small companies are usually better served by some existing products, but then the issue of reconfigurability of chosen solution may arise.

There are different tasks contract management systems should perform:

- Contract drafting and templates creation
- Contract negotiation and approval, version control
- Milestone tracking and compliance management
- Notifications
- Completion tracking
- Document search and retrieval

Although such systems can exist and be functional without implemented business workflow, this process perspective is quite important for contract management in order to have time-efficient, flexible, yet reliable contract handling. Hence, most of the contract management systems tend to deploy, or even impose, some business workflow. The system presented here has also started as an application tailored to specific needs, but it soon become obvious that more general and process-aware approach is needed in order to create robust and flexible system.

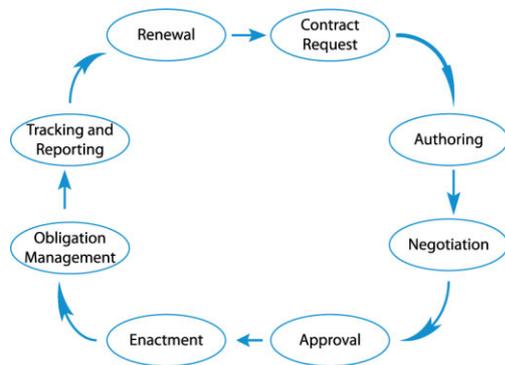


Figure 1. Typical phases of contract lifecycle

Although there are a few slightly different definitions of contract lifecycle phases, most of them agree that full contract lifecycle consists of common phases depicted in Figure 1.

There are different solutions available commercially as a standalone products (Prodiagio, Agiloft Contract Management Software, Selectica, Blueridge Software Contract Assistant...) or available as a part of a larger business suite (Microsoft Dynamics CRM, SAP CRM, Oracle Contract Lifecycle Management module...) as well as some freeware solutions. Some solutions are installable applications, some are web-based application deployable on customer servers, and some are offered on a "Software as a Service" (SaaS) model. SaaS model for contract management systems has its set of challenges as described in [4].

The development of the system presented in this paper is centered on open source business process management system Activiti [5]. This approach allows for flexible contracts management workflows while implemented document model is allowing for general purpose implementation.

The aim of the system is to support contract management throughout the whole process:

- By allowing easy contract drafting, with a repository of preapproved general purpose contract paragraphs; drafted document are also subject to approval of the legal department (resolving any outstanding issues which may arise from conflicting paragraphs)
- By allowing contract drafts negotiations with a client
- Final approval and signing of the contract
- Prompt notification of expiring contracts, allowing for contract renewal (especially important for service providing companies such as telecom operators, utility companies, and similar)
- Thanks to the process engine at its core workflow is easily adjustable to different needs
- The core of the system is built as a service-oriented application, allowing for further expansions and client application development
- One client implementation is a web application.

II. CONTRACT TEMPLATES AND CONTRACT DOCUMENT MODEL

Contract management system is, in its nature, document oriented. International standard ISO 82045-1 [6] defines principles and methods of document management. By the given definition, to facilitate document management within their life cycle and for their exchange between partners, documents shall be associated with a set of metadata, i.e. data identifying and/or describing the document [6]. In addition, we can distinguish single document (document associated with its metadata), compound document (composed of more than one document types) and associated with its metadata, document aggregation (assembly of self-contained documents), document sets, and linked documents. Version control is also one of the important issues that need proper consideration.

When contracts are taken into consideration, it is important to notice that they consist of variable data, and in most cases, fixed legal paragraphs. Variable data related to contractual parties and contract terms (such as contract validity period, special discounts granted etc.), may be embedded in paragraph text. Variable data is associated with paragraphs. Since VariableModel allows definition of variable types, they can be of any type relevant to the contract context. One important use of variables is to allow references to other paragraphs.

One approach to contract creation may be to use document templates. In manual processing, usually a Word documents are saved as templates, and manually filled with variable data during contract creation. This manual editing is error prone.

Furthermore, having a complete document template as a basic unit of work turned out not to be very flexible, since it may require a major redaction if terms of a contract is to be adjusted. For this reason, we adopted the view of document template as a sequence of paragraphs.

For contract creation, it is beneficial if these paragraphs are preapproved. That allows for new contract templates to be easily created by assembling existing paragraphs, which can be easily added, reordered, or removed. Even though templates are compiled from preapproved paragraphs, they also need final approval, since paragraphs can be internally correct, but their correlation with other paragraphs may not be acceptable. When template is approved, it can be used for fast creation of valid contracts. Usually, for most business-to-customer relations this is sufficient, and during contract preparation, only variable parts need to be filled.

However, for more complex contracts (usually business-to-business), a negotiation phase of contract lifecycle usually produces slight changes to the original text, therefore, and in this case, flexibility is needed. We added a degree of flexibility by specifying if some paragraph texts are editable. Hence, when new contract is prepared from the template, template parts are instantiated, and editable paragraphs are unlocked. Nevertheless, such an option requires that the final

contract goes through final proof-reading by legal department before final approval by contractual parties.

Basic conceptual model of contract document is given in figure 2.

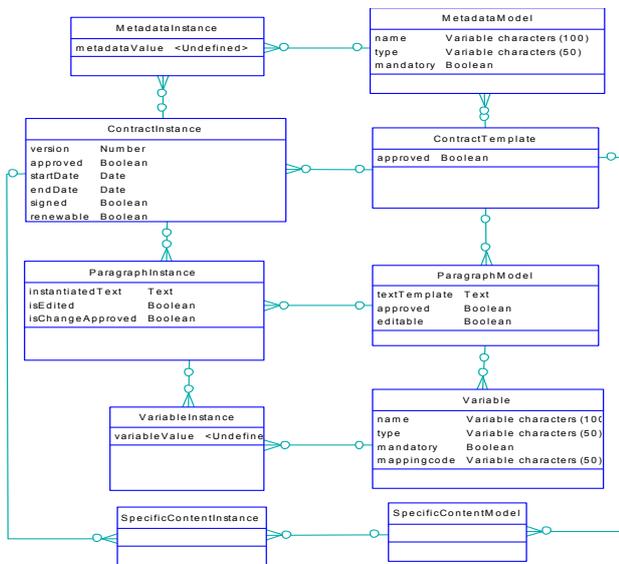


Figure 2. Conceptual model of contract document

This document model has proved to be sufficient for most purposes, especially for a creation of relatively simple contracts. However, for more complex documents, for instance, telecom operator contracts or some utility company contracts, with a vast number of tariff modules, options, and packages, such simplistic approach is not productive. In order to support complex data we added an extension point to the model, represented by SpecificContentModel entity. In the implementation, this entity is interpreted as an abstract class. Specific implementations can then inherit and create their data model to suit their specific needs. One of the specialized implementation was developed for telecom operator representative agents, responsible for contracts with individual users and small companies. In this case, contract needed to have a representational framework for offered packages, tariffs, tariff options and granted discounts.

III. MODELLING CONTRACT LIFECYCLE MANAGEMENT PROCESS

As stated earlier, contract management is a process. Although other approaches, like [7], are viable, the lack of processing support (process logic entangled in the system code) becomes a problem when support for different types of contracts or even slightly different contract management workflows needs to be added to the system. Use of some business process management (BPM) system to manage this process adds a degree of flexibility to the system. In this case, contract management workflow can be expressed as a process model in some process modelling language (BPMN, BPEL, XPD...).

Languages vary in degree of their representational completeness and their applicability to some process engine. With enough technical details entered into the model, a process engine is capable of running the process according to the "prescription" given by the process model. More details about concepts, languages and architecture of process modelling systems may be found in [8]. Some analysis of different languages is given in [9, 10]. In last few years BPMN language [11], developed by Object Management Group, has become a widely accepted standard.

Our implementation is relying on open source Activiti process engine [5]. This engine is also used by one of the leading open source document management systems – Alfresco, to add process support. Although using Alfresco with defined workflow was one solution to contract management, we wanted our own flexible document model and extension points. The choice of Activiti was driven by our prior experience with BPM systems [12], its status of open source solution, and simple API.

BPMN defines Flow Objects (events, activities, gateways), Artefacts (Data Object, Annotations), Connecting Objects (Sequence flow, Association, Message Flow), and Swimlanes as building blocks for creating a process model.

For successful process modeling, it is important to observe functional perspective of process (what needs to be done) and organizational perspective (who is responsible for activities). Also important are informational perspective (what information process execution needs), as well as IT landscape perspective, which defines a correlation to existing IT systems. Organizational perspective is in BPMN modelled through users, groups and user membership to groups. In BPMN notation, organizational aspects are depicted by Pools and Swimlanes. Existence of these symbols in a process model is a visual aid, since it doesn't have a semantic meaning for execution, i.e. activity placed in one Swimlane still needs to be explicitly assigned to a specific role.

In our system, we have implemented a basic contract management workflow suitable for most general purpose requirements. This workflow is given in Figure 3. In this process model, Contract Authoring is a complex activity that can be executed as a standalone process, and is accordingly represented as Call Activity. Similar applies to Contract implementation monitoring.

Since small business often don't require special procedures for Contract implementation monitoring, it is in a conditional branch of the process model. If monitoring is required upon contract approval, it will be executed, presumably for the whole period of contract duration. If so, a separate monitoring process should be modeled taking into account all activities required for contract context, and this model should be provided. Otherwise, process is captured by internal timer set to the contract expiration date, hence keeping the process and contract in an active state. If further actions after contract signing are not required, this timer can be set to a current time (or model changed to be skipped altogether). For

renewable contracts, a message activity is used to send notification that a contract is ending and initiate contract renewal process. Figure 4 shows the model of the contract authoring process.

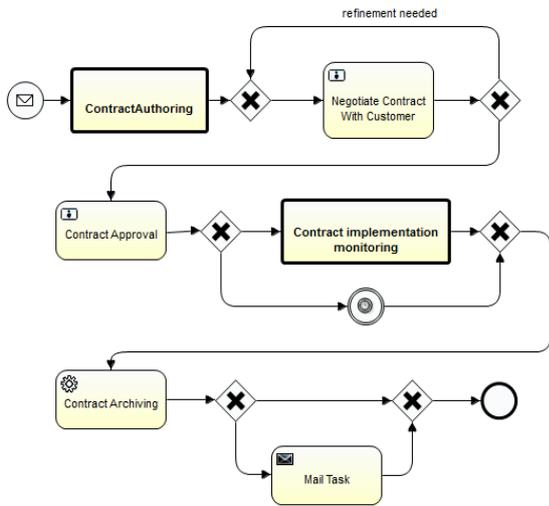


Figure 3. Basic contract management workflow

After request for contract is received, sales staff will determine appropriate contract type and check if prepared template for such contract exists. If it does, it will be used. If appropriate contract does not exist, user will be able to create a new one from existing preapproved paragraphs (possible extension is to allow free text paragraphs so that users are allowed to create new body of text, but this option should be controlled by access control rules). Such contract template proposal now has to go through approval process. After the user has got hold on approved contract template (previously existing, or newly created), it fills in contract data. During this process, if other things, rather than just variable data, are changed (text of paragraphs), the contract will be marked as “dirty” and

will require further approval.

In order to evaluate if proposed model is adaptable to different needs, we’ve modeled a process of contract management in electricity supplier (utility company). Studying their contract management activity diagrams, we have learned that, apart from more elaborate definitions of user roles, and more detailed activities or their changed order, there are no significant differences. Hence, main modelling activities in this case were concentrated on an organizational perspective, i.e. mapping organizational hierarchy of power utility company to user and groups in identity module. Also, a significant attention was given to information perspective – since the process has well-defined support documents and information objects that are used to fulfill some tasks. This detailed process is depicted in Figure 5. If an electronic signature is used on documents, then automatic mailing tasks may be used for final document delivery to the client. Otherwise, the final task needs to be performed as a standard user task. A system task (service task) is used to automatically transfer client data into a client database.

IV. SYSTEM ARCHITECTURE

At the core of our system, as mentioned before, is an Activiti BPM engine. A relational database, in our case MySQL, is used by the engine, and accordingly, by the rest of our system. That is configurable, since BPM engine can use any database supported by the Hibernate ORM system. Since Activiti is implemented in Java programming language, and due to our prior experience in development of enterprise applications in Java, we have opted to develop contract management web application also in Java. Such a choice was not mandatory nor forced upon us by the Activiti BPM engine as it exposes most of its API through REST services. Therefore, a developer can choose other options to access BPM engine. In that case, some consideration is needed if event listeners or service classes are specified in the process model. Activiti is implemented as a standard Java package and may be used as a standalone engine, or in other environments, such as a servlet or EJB containers. Since our primary implementation was intended to be a web application,

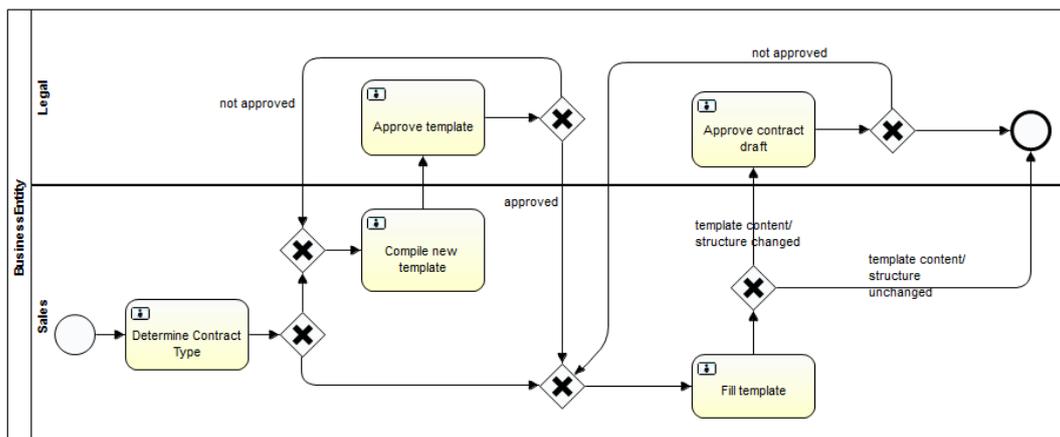


Figure 4. Contract authoring workflow

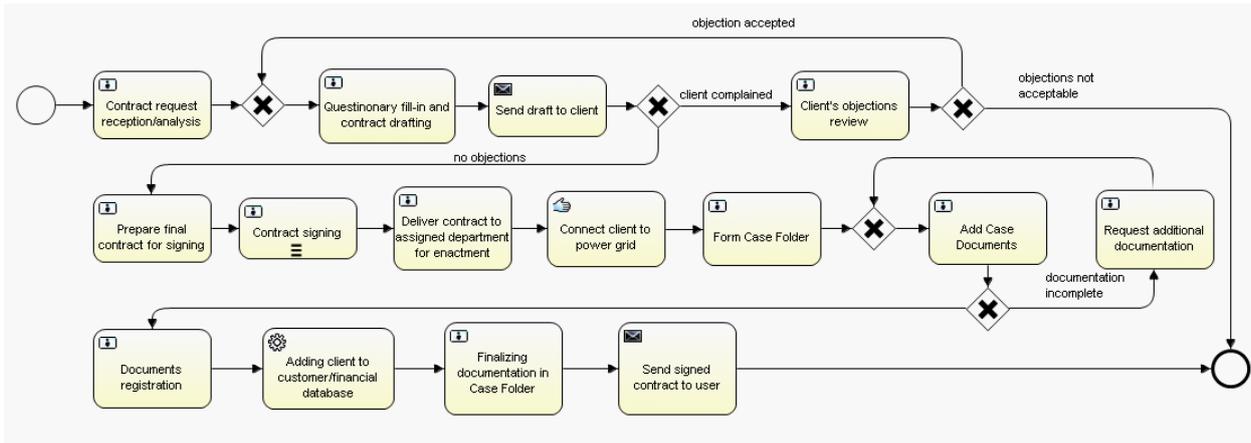


Figure 5. Contract workflow for Electricity Company

Activiti engine is also deployed on the application server. Client application, in our case web browser, is accessing the application via HTTP(s) protocol. However, since functionalities of the application may be exposed through REST services, a different type of client applications can be developed. That option is very useful when an integration of contract management into other business applications is required. If that is the case, a contract management application will use existing services, but can be implemented as a module in larger software system.

contract management process, there is a high probability that we don't have enough information, at the time of modelling, to specify form properties. However, developers can create customized forms, or even generate them dynamically, and attach them to the tasks through the form key property. This approach can be used to map our specific data to user tasks in the process.

The deployment diagram of the core modules is given in Figure 6.

V. ADDRESSING SECURITY ISSUES

Contract management is a sensitive issue, hence requiring appropriate measures to enhance contract confidentiality and access control is mandatory. Our pilot implementation is developed as a web application. In a case of in-house installation, it is a good practice to allow access only from LAN. Additionally, in order to enhance security, it is recommended that server setup enforces access through HTTPS channels. It is especially important in case of installation on publicly accessible servers. Access control is based upon Activiti BPM engine Identity module, using role-based access control. Most of the data necessary for process running is controlled by the process engine and Hibernate ORM mapper, using parametrized queries, thus lowering the possibility of SQL injection attacks. Data coming from input forms are also preprocessed for this threat. Strings that comes from user input (HTML forms in our case) are checked for XSS (Cross Site Scripting) attacks. Also, appropriate measures, to prevent CSRF (Cross Site Request Forgery) attacks, are implemented in the server application. Database servers are installed on a separate node (server), and access is allowed only through application server, with appropriate credentials.

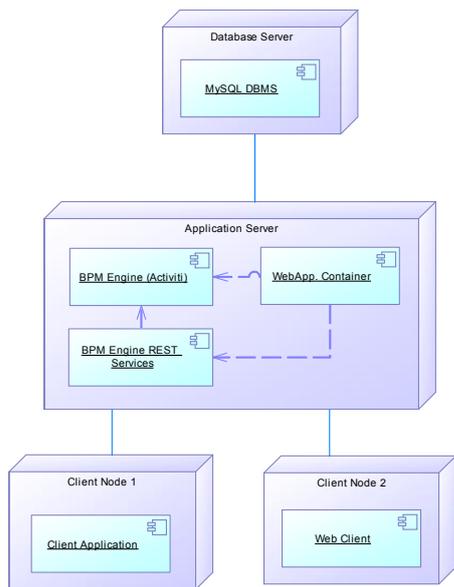


Figure 6. Deployment diagram of application modules

Activiti engine FormService features give us an opportunity to dynamically create user interface for specific tasks. If this feature is used, specific data needed to perform certain tasks may be declared in the model. Simply by reading the model, the process engine is capable of building necessary forms for users. However, if the data necessary to perform the task is dependent on contract type being processed or data created during the

VI. CONCLUSION

In this paper, a contract management system, as a specific type of document management system, is presented. A document model, suitable for general purpose contracts, but extensible for complex contract

structures is also presented. The presented system is developed using BPM engine at its core. Basic contract lifecycle management business process has been described, modelled and implemented in pilot project. Additionally, in order to assess overall functionality of the system, several real-life contract handling business processes has been modelled and implemented.

Further work on this system will be concentrated on extending access control features with COBAC (*context-sensitive access control model for business processes*) [13]. Although presented document model has been sufficient for the current implementation, it will also be enhanced by extending paragraph properties, making it possible to express more complex relationships between parts of a document. Such enhancement will bring us an opportunity to develop a system that will keep track of compatible and incompatible document parts, as designated by legal department. In that case, template creation would be simplified, and user would be warned if some parts of a document are conflicting. Additionally, since new standard "The Open Contracting Data Standard" [14] is emerging, aimed primarily at government contracting procedures, such as public procurement, this data model also needs to be taken into consideration.

Multi-tenancy, i.e. ability to offer this system as a service to multiple business entities will be further explored. In that case, special attention must be given to a strict separation of content, access rights, and also possible encryption of data may be appropriate.

REFERENCES

- [1] International Organization for Standardization (ISO). (2001): "ISO IEC 82045-1: Document Management – Part 1: Principles and Methods", ISO, Geneva, Switzerland.
- [2] Hollingsworth, David: "The Workflow Reference Model", Workflow Management Coalition, Cohasset, MA, USA (1995).
- [3] Krishnan, Rupa, Lalitha Munaga, and Kamalakar Karlapalem: "XDoC-WFMS: A Framework for Document Centric Workflow Management System". In *H. Arisawa, Y. Kambayashi, V. Kumar, H. Mayr, and I. Hunt (eds): Lecture Notes in Computer Science 2465*, Berlin: Springer, (2002). pp. 348-362.
- [4] Kwok, Thomas, Thao Nguyen, and Linh Lam. "A software as a service with multi-tenancy support for an electronic contract management application." *Services Computing, 2008. SCC'08. IEEE International Conference on. Vol. 2. IEEE*, 2008.
- [5] Activiti BPM Platform, <http://www.activiti.org/>, accessed: January 2015.
- [6] International Electrotechnical Commission, "Document management - Part 1: Principles and Methods", *INTERNATIONAL STANDARD ISO-IEC 82045-1*, 2001.
- [7] Wenwen, Ding, Chen Yan, and Jiang Zhuoren. "Design of Contract Management System Based on J2EE Architecture." *Business Computing and Global Informatization (BCGIN), 2012 Second International Conference on*. IEEE, 2012.
- [8] Mathias Weske: "Business Process Management: Concepts, Languages, Architectures", 2nd ed. 2012, XV, 403 p. 300 illus. Springer-Verlag Berlin Heidelberg 2012, Springer Link
- [9] List, Beate, and Birgit Korherr. "An evaluation of conceptual business process modelling languages." *Proceedings of the 2006 ACM symposium on Applied computing*. ACM, 2006.
- [10] Indulska, Marta, et al. "Representational deficiency of process modelling languages: measures and implications." (2008). In Golden, W. and Acton, T. and Conboy, K. and van der Heijden, H. and Tuunainen, V., Eds. *Proceedings 16th European Conference on Information Systems*, Galway, Ireland.
- [11] Business Process Model and Notation, Specification, Object Modelling Group, <http://www.bpmn.org/>, visited January 2015.
- [12] Zarić Miroslav, Segedinac Milan, Sladić Goran, Konjović Zora, "A Flexible System for Request Processing in Government Institutions", *ACTA POLYTECHNICA HUNGARICA*, vol. 11, br. 6, p. 207-227, 2014.
- [13] Gostojić, Stevan, Goran Sladić, Branko Milosavljević, and Zora Konjović: Context-sensitive Access Control Model for Government Services, *Journal of Organizational Computing and Electronic Commerce*, vol. 22 no. 2, (2012) pp. 184-213
- [14] Open Contracting Partnership, "The Open Contracting Data Standard", <http://standard.open-contracting.org/>, accessed: January, 2015.