CMDB BASED ASSET MANAGEMENT WITH INTEGRATED OPEN SOURCE SYSTEMS

Zarić Miroslav[1], Sladić Goran[1]
{miroslavzaric, sladicg}@uns.ac.rs
[1]*Fakultet tehničkih nauka, Novi Sad*

Paroški Milan[2], Andrić Sanja[2]
{milan.paroski, sanja.andric}@vojvodina.gov.rs
[2]*Sektor za informatičke poslove Vlade APV*

Abstract *– This paper describes integration of different Open Source systems, in an attempt to provide a system capable of fullfilling the needs of asset management. Integration is based around the CMDBuild platform, serving as an asset register, while other systems provide added functionality (document management, workflow management, spatial data storage and representation). Brief introduction of ITIL and its Configuration Management Database (CMDB) part, as well as short overview of available commercial and Open Source CMDB solutions is given. Data model for a property management (for governmental institutions) and implementation of such model is given as a case study.*

1. INTRODUCTION

Successful operation of almost any business process is highly dependant on the efficient resource management. At the present time, success of a business process is also critically linked with the support provided by IT infrastructure and services. One of the services, that IT sectors in any type of organization (business or governmental) must provide, is the resource management, or in a broader view, the asset management for any type of assets involved in a business process. Any resource or capability that is used to perform business process or to deliver some business service is considered as an asset.

Furthermore, IT services needs to keep pace with growing demands – so there is a need to keep track of available and required IT resources needed for the successful operation of a primary business.

Asset management and maintenance implies the existence of a database that contains all relevant data describing assets. In addition, for some of assets it is necessary, or at least desirable, to have additional documentation. In some cases, a proper resource/asset management requires specific process workflow to be followed. For some assets (hardware, software, network resources, sensor data etc.), an automatic data gathering and storage is possible and should be available in the asset management system.

Therefore, a comprehensive resource or asset management system should consist not only of relational database and of application for data input/output, but also of some document management system, and usually some workflow engine. Optionally, such system could provide more functionality, depending on resource types, if it would include some system for an automatic data gathering.

In our paper we are proposing an asset management system based on configuration management database (CMDB) that can easily be extended by document management system, workflow engine, automatic data gatehring module(s) and, in particular, support for managing geospatial aspects of the assets.

The rest of the paper contains following sections. Section 2 is about ITIL (Information Technology Infrastrucure Library), Configuration management database and its relation to asset management. Section 3 briefly presents the role and selected open source solution for the document management system. Section 4 is the central section of the paper. It presents a case study – Integration of CMDBuild and Alfresco for Property Management for local/provincial government institutions. The basic case study is further extended by adding spatial extensions to the assets database. Section 5 brings concluding remarks and outlines further development of the system.

2. ITIL AND CONFIGURATION MANAGEMENT DATABASE

Looking at IT resources, today there is a proposed standard set of concepts and techniques for managing IT infrastructure, its development and functionality. This set of standards has been discussed in series of books that cover the area of IT management technologies, and is known as the IT Infrastructure Library - ITIL (Information Technology Infrastructure Library) [1].

An important part of the recommendations of ITIL are the models and guidelines for forming the configuration database, database that will hold information on all essential resources, especially regarding IT infrastructure (hardware, software and human resources), but also other resources. ITIL, as we know it today, was developed under the title “Managing the IT infrastructure in public administration” (Government Information Technology Infrastructure Management Methodology - GITMM). However, it was not until the late 90's, of the last century, that ITIL recommendations have come into widespread use. In December 2005., a revised version of ITIL was published, better known as ITIL v3. It has entered into service in May 2007. The specification of the version originally contained five documents:

1. Strategy service (Service Strategy)
2. Design Services (Service Design)
3. The transition services (Service Transition)
4. Functionality of services (Service Operation)
5. Continuous improvement of service (Continual Service Improvement)

One of the important things ITIL has introduced is a concept of Configuration Management Database (CMDB).
It is a specialized database designed for keeping, updating and searching for information on the IT configurations, and is known under the acronym CMDB.
In version 3 ITIL changed the name of a configuration **unit** (property), which is now regarded as an **asset**, or simply a configuration item (CI). The very concept of the CMDB has been redefined. ITIL v3 defines CMDB as "A database used to store Configuration Records throughout their lifecycle. The Configuration Management System maintains ***one or more*** CMDBs, and each CMDB stores Attributes of CIs, and Relationships with other CIs. System Configuration Management (The Configuration Management System) maintains one or more of the CMDB database(s) and each CMDB database contains attributes of configuration items (Configuration Items), and their relationships with other configuration units." Version 3 explicitly allows for the appearance of more CMDB databases.

CMDB and Asset Management

Although developed as a specialized database for storing information about IT resources and their configurations, these bases are in fact suitable for the storage and manipulation of information about any other resources/assets of interest. The very definition of Asset and related terms in ITIL v3 [2] is not limited to IT resources and services:

Asset - Any Resource or Capability. Assets of a Service Provider include anything that could contribute to the delivery of a Service. Assets can be one of the following types: Management, Organization, Process, Knowledge, People, Information, Applications, Infrastructure, and Financial Capital.

Asset Management - Asset Management is the Process responsible for tracking and reporting the value and ownership of financial Assets throughout their Lifecycle. Asset Management is part of an overall Service Asset and Configuration Management Process.

Asset Register - A list of Assets, which includes their ownership and value. The Asset Register is maintained by Asset Management.

There are several available commercial CMDB systems in the market. They are either stand-alone software packages or a part of a larger integrated system. Currently the most developed, and most commonly used commercial CMDB systems are: BMC Atrium CMDB [3], CA Enterprise IT Management - CA CMDB [4], IBM Tivoli [5], The HP Universal CMDB [6]
For an organization that considers implementing ITIL recommendations, or at least its CMDB part, there are also Open Source alternatives, interesting in terms of price, but also in terms of customization to specific needs. Some of more frequently used Open Source CMDB Systems are: Lokomo OneCMDB [7], CMDBuild [8] and RapidOSS [9]. More details about these systems is given in [10].

3. DOCUMENT MANAGEMENT SYSTEMS

The document management system (DMS) is an information system that enables the creation, storage, transfer, organization, retrieval, exchange, manipulation and update of documents [11]. In other words, it is an information system used to store and track electronic documents or scans of paper documents.
The basic functionalities of document management systems are [11][12]:

- support different structural forms of documents and metadata association to documents,
- business processes definition and document lifecycle management,
- indexing and retrieval of documents,
- optical character recognition,
- users collaboration and version control,
- security, and
- integration with other systems.

As with CMDB, there is a choice between different commercial DMS, as well as Open Source alternatives for such systems.
Vendors recognized by the 2009 Gartner ECM Magic Quadrant include Alfresco, Autonomy, Day Software, EMC, Ever Team, Fabasoft, HP, Hyland Software, IBM, Laserfiche, Microsoft, Newgen Software Technologies, Objective Corporation, Open Text, Oracle, Perceptive Software, SAP, Saperion, Siav, SpringCM, SunGard, Systemware, Xerox and Xythos Software.

Alfresco is one of the most widely adopted Open Source solutions. The Alfresco system is the web-based DMS system that is suitable for content management in small and medium organizations as well as in large, complex geographically distributed organizations. It is licensed under the *GNU General Public License*. Alfresco uses only technologies of the open source software and the open standards. It allows relatively easy customization and expansion of its functionality through the use of XML configuration documents and integration with other applications using the open standards.

4. CASE STUDY – Integration of CMDBuild and Alfresco for Property Management for local/provincial government institutions.

As a case study, we will present data model suitable for property management for local/provincial government institutions. Data model will be applied to a CMDB system, and documents related to specific assets will be stored in the Alfresco repository. We will describe the necessary steps to achieve integration between these two systems.
Property management is one of the vital activities for local and provincial government institutions. A complete registry of all properties in use, or own, by the local / provincial government would allow them for better control their sources of revenues (through tax and/or rent collection), as well as their expenditures.

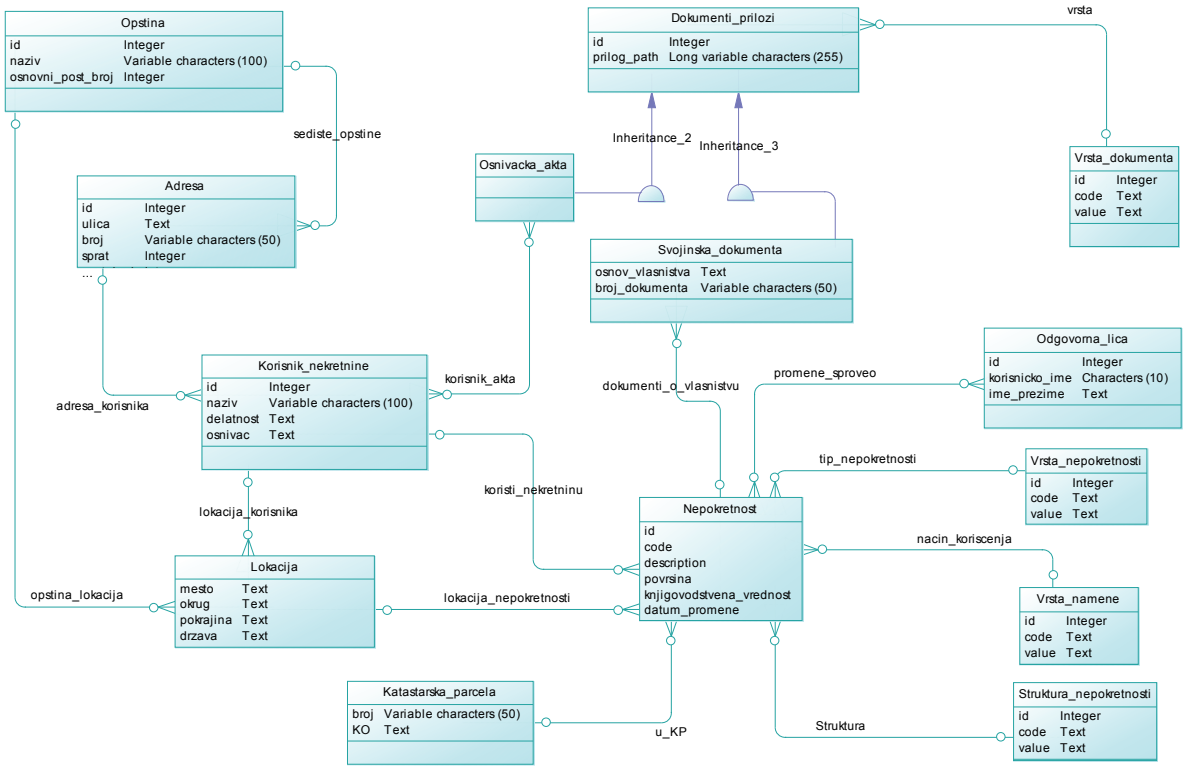

Figure 1. The data model for the property management application

Such a registry would also allow for simplified access to information about properties owned, their respective values, available office spaces, property occupants etc. This information can be crucial for future planning of a resource usage.

It is important to note that such a registry is not intended to replace official Land Registry (cadastre), but instead to be a tool that would allow local authorities easy accessible overview of their assets.

First step in this process was to define an appropriate data model. After few rounds of consultation, the data model shown in Figure 1 was determined as sufficient.

Next step in this process was to implement the data model in an appropriate database that will be used as asset/property registry.

Although for each asset there is a different set of data (attributes) that describes it, concept of CMDB is well suited for most of them. Most of the CMDB systems will allow for creation of a specific data model, not limited to IT-related items. CMDB allows users to create their own classes for assets, with required attributes, and new types. Instead of building a custom application with more or less predefined database structure (data model), using CMDB and its tools will allow users to gradually develop their own data model (types and classes, and relationship between them). In that process, for each new type or class, an appropriate table in the database is created, and for every relation defined in a CMDB application an appropriate relationship is generated in the underlying database management system.

For our case study we opted to use CMDBuild as the asset registry. Although any Open Source CMDB solution would be acceptable, CMDBuild was chosen because of our prior knowledge of it, its relatively high level of completeness, simple installation and most of all its embedded capabilities for integration with other systems, primarily with Alfresco and Enhydra Shark. CMDBuild is bundled with an Open Source report building engine - JasperReport, allowing for creation of simple or complex reports exported as CSV, ODF or PDF files. Also in a new version of CMDBuild (ver > 1.3), there is a possibility to declare spatial (georeferenced) data as attributes of classes, feature that is very well suited to the problem of property management. One very important feature of CMDBuild is its support for internationalization – allowing us to create user interface in the Serbian language (both Latin and Cyrillic scripts).

CMDBuild is relying on PostgreSQL as its database management system. PostGIS, a spatial extension to the PostgreSQL database is used to enable spatial data as attributes.

The CMDBuild application is separated into two modules:

- Data Administration Module – used by administrators to create new types, define classes and their attributes, define relationships, users and groups, and setting up entire application;
- Data Management Module – used by regular users, for entering, editing, viewing, and deleting data.

Any type of asset that will be managed through CMDBuild will be represented by its own class, with appropriate attributes. All classes created in CMDBuild are derived from one base class, named *Class*, that contains basic attributes *Id*, *Code*, *Description* and *Notes*. The attribute *Id* is hidden and is never displayed to the users, since it is maintained as the primary key by the underlying DBMS itself. *Code*, *Description*, and *Notes* are always present, but they can be "turned off", i.e. made inactive if necessary.

While transferring our data model to CMDBuild, some entities were created as classes with their respective attributes, while others (like *Struktura_nepokretnosti, Vrsta_namene, Vrsta_nepokretnosti, Vrsta_dokumenta)* were created as new ***Lookup*** types. Lookup types are used in CMDBuild whenever there is an entity that appears as attribute in other classes, and there is a need to restrict its values to previously determined list of allowed values. For defined *Lookup* type, a list of items is then created.
Besides defining classes and their attributes, CMDB allows for defining relations between classes. Such relationships are referred to as *Domains*. By creating a new domain, a user can define relationships existing between two classes. Relation between two classes can be of cardinality 1:1, 1:N, N:1, N:N. There are slight differences in a way user interacts with the system, depending on the type of relationships. Relationships of type 1:1 and 1:N can be directly expressed as an attribute of a class, with type set to the special type REFERENCE. Other relationships are later handled not by changing single attribute, but instead by explicitly defining existing relations.
Some entities in our model (*Nepokretnost, Korisnik*) contain references to electronic versions of important documents related to them. Although it would be possible to implement some kind of a local document repository, better solution is to use some of existing, available systems for document management.
Instead of creating our own class for relevant documents, an already supported concept of "attached document" is used to link relevant documents with their respective asset. Attached documents are stored in Alfresco DMS.

In our case study, **Alfresco** DMS has been used for a number of reasons:

- CMDBuild, which is used as the asset registry, already comes with the embedded capability to connect to the Alfresco system;
- The Alfresco application server was already installed and was in use at the location, so users have prior knowledge of it;
- The combination of CMDBuild and Alfresco would allow for more versatile user roles – some user would have access rights only to CMDBuild, while others could have access only to Alfresco, or both systems;
- Integration of these two systems is easily achieved.

To achieve interaction between CMDBuild as the asset registry and Alfresco as the document repository, the following steps have to be performed:
On the CMDBuild side:

- New *Lookup type* needs to be created, containing definition of document types that will be allowed to be used as attachments for any asset. In our case this new type was called *AlfrescoCategory* – containg items (*Kupoprodajni ugovor, Ugovor o zakupu, Osnivački akt, Izvod iz zemljišne knjige)*.
- Set the connection parameters of the Alfresco server. It is necessary to set the URL of the Alfresco Web service connector, address of the FTP server provided on the Alfresco server instance, as well as specific paths to access the proper workspace created on Alfresco for this purpose.

On the Alfresco side:

- A new Alfresco **aspect** needs to be created to support metadata sent from CMDBuild and to attach those metadata to documents received during upload from CMDBuild. The XML files with **aspect** definition are provided with the CMDBuild installation.
- An appropriate workspace needs to exist, in which the document uploaded from CMDBuild are to be stored (in our case a new one has been created specifically for this application, although any existing one could have been used).
- A new category, with the same name as one defined in CMDBuild has to be created.

A screenshot displaying the administration module (using the Serbian localization) is depicted in Figure 2. In this case, the module is in the state of managing *Lookup* types.

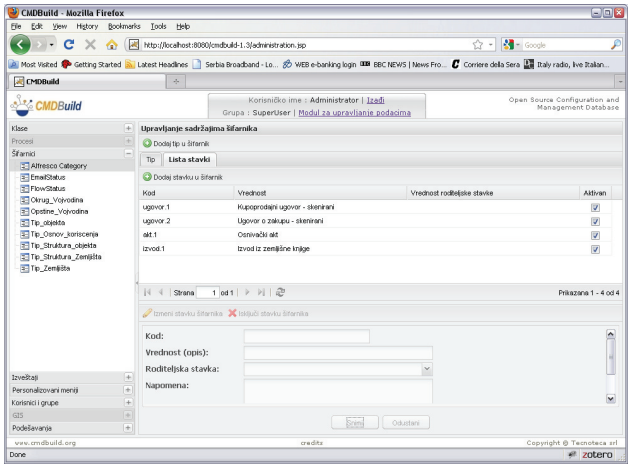

Figure 2. The lookup type "AlfrescoCategory" and its item list

Architecture of integrated CMDBuild and Alfresco system is displayed in Figure 3.

The communication between CMDBuild and Alfresco is dual, attached documents are transferred via FTP, while accompanying metadata is sent to Alfresco via the SOAP protocol i.e. web service interface.
If the web service communication is not well configured, file transfer may be successful, document could end up in the Alfresco workspace, but no relation would be established, since its metadata would not be correlated to the file itself. This would result in an "orphaned" document on the Alfresco side. Documents related to

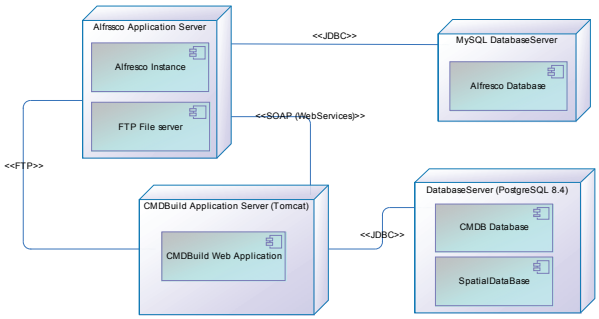

Figure 3. Deployment diagram of integrated CMDBuild-Alfresco system

some assets are visible as "Attachments" in the Data Management module of CMDBuild.

Although integration of CMDBuild and Alfresco will enable functional asset management, with all accompanying documents, additional systems can be incorporated as well.

Since this setup in our case study was primarily intended to be used as a property registry, ***georeferenced (spatial) data*** would be a very welcomed addition. Fortunately, adding GIS support is already incorporated in CMDBuild, although that functionality is not enabled by default. To enable use of spatial data as some asset attributes the following steps are needed:

- The PostGis extension of PostgeSQL must be installed;
- Since the CMDBuild database installed with the system does not contains „spatial data", a few scripts need to be executed on the database to enable GIS features. Scripts for the Linux installation of PotgreSQL are distributed with CMDBuild, while scripts for the Windows version of PostgreSQL can be found in the installation directory of PostgreSQL itself;
- In CMDBuild administration module, GIS support needs to be activated; optionally administrator can set initial latitude and longitude and zoom for map views;
- In the GIS tab of the administration module of CMDBuild the additional configuration options can be set:
    - Icons used for displaying different kind of spatial data;
    - Connection to external services (maps providers). Already supported are GoogleMaps, Open Street Maps and Yahoo Maps, as well as support for connections to Geoserver;
    - Setting up layers and Geoserver layers.
- Finally, in the CMDBuild administration module, for desired class, a new attribute needs to be created under the group "Geographical attributes". The geographical attribute can be of one of the following types:
    - POINT
    - LINE
    - POLYGON

Figure 4. displays one object with its spatial attribute (*Pozicija)* set, and displayed in the map view, using the external Open Street Maps as a map provider (data displayed here is randomly entered, not representing any real object).

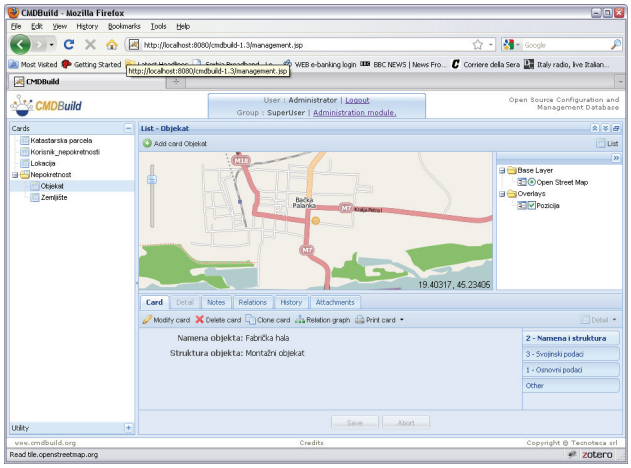

Figure 4. Map view of a geographical attribute of an object

Best usage of the GIS support would probably be to use custom set up, with ***GeoServer*** containing detailed layers of spatial data, as a service provider.

Another option available directly from CMDBuild is to connect it to a workflow engine, namely ***Enhydra Shark***. This approach, where data manipulation in the CMDB database is steered through some controlled process is also recommended by ITIL guidelines.

Yet another option, very interesting when assets managed in CMDBuild are on the level of IT infrastructure, is to connect CMDBuild to ***OCSInventory,*** the software capable of automatic data gathering about computer hardware and software configurations, network equipment and other resources available through a local network. Integration of CMDBuild with OCSInventory is discussed in [13].

Property management was first, but the necessary phase of comprehensive asset management for the provincial government. There are plans to use the data gathered in CMDBuild to proceed to the next level of detail. For some properties (buildings) which are used by the provincial government, more detailed data would be added (floor plans, offices, occupants, and complete IT infrastructure present at the location).
In this case data model would gradually expand, as well as a user base. In addition, new geographical attributes would be needed for more detailed plans. In this case, the existing system would have to contain one node running GeoServer instance, holding additional layers of spatial data. With growing complexity of the system and growing number of users, user and group management needs to be carefully planned, and some well-defined processes deployed.
Deployment diagram presenting such a comprehensive, fully integrated system with CMDBuild at its core is given in Figure 5.

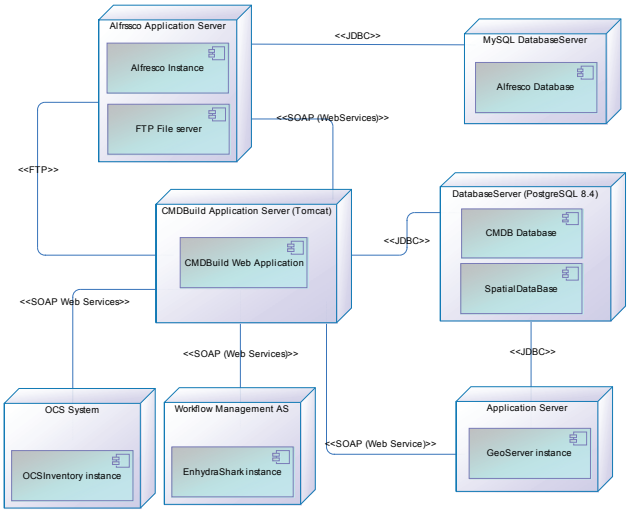

Figure 5. Deployment diagram of the integrated Open Source systems

CONCLUSION

CMDBuild is one of the Open Source products designed following ITIL guidelines and aimed at providing the efficient CMDB system. Although CMDB databases are designed to be used as the specialized databases for storing data about IT configurations, the basic concept they were founded on allows for their usage for management of any assets.

Applying appropriate data model will allow CMDBuild to be used as a registry of arbitrary assets. In our case study, we developed the appropriate model for property management, intended primarily for the IT sector of the provincial government. Such a registry will enable the IT sector to provide valuable services to other local and provincial institutions, allowing them easy access and insight of their properties, their locations, current values, and information on how that properties are currently used.

The intention of such a service is not to replace any function of the official Land Registry (cadastre), but to be a helpful tool for day-to-day operations of local/provincial institutions.

Since such an asset registry is primarily focused on properties, a lot of accompanying documents is necessary to properly maintain and manage those assets.

CMDBuild natively support integration with Alfresco DMS, and integration was preffered, and better option compared to creation of a local document repository.

Integration of these two systems was easy to achieve and gave users the benefits of both worlds – data management in CMDB database, with attached documents permanently stored and visible with their metadata in Alfresco.

Further development of this system is envisioned, with addition of new types of assets to CMDB (internal structure of some buildings, IT and human resources), as well as integration with additional Open Source systems, such as workflow management system and automatic inventory system for IT resources. Addition of GeoServer to this integrated system would enable more detailed maps of objects and infrustructure layouts to be linked with related assets.

REFERENCES

[1] "Information Technology Infrastructure Library," Official ITIL® Website
Available: http://www.itil-officialsite.com/.

[2] "ITIL_Glossary_V3_1_24.pdf."
Available:http://www.best-management-practice.com/gempdf/ITIL_Glossary_V3_1_24.pdf

[3] "BMC Atrium CMDB - Configuration Management System - BMC Software," BMC Atrium CMDB Available: http://www.bmc.com/products/product-listing/atrium-cmdb.html.

[4] "CA CMDB - CA," CA Enterprise IT Management – CA CMDB, Available: http://www.ca.com/us/cmdb.aspx.

[5] "IBM - Integrated Service Management software, IBM Tivoli."
Available: http://www-01.ibm.com/software/tivoli/

[6] "HP Universal CMDB software - HP - BTO Software." Available:
https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-15-25^1059_4000_100__

[7] "Lokomo Systems AB," One CMDB Overview Available:
http://www.lokomo.com/products/onecmdb_overview.shtml.

[8] "Home — CMDBuild," CMDBuild open Source configuration and management database
Available: http://www.cmdbuild.org/.

[9] "RapidOSS | iFountain.com", Rapid OSS Insight into IT Operations Management
Available: http://www.ifountain.com/rapidinsight.

[10] M. Zarić, A. Kovačević, and Z. Konjović, "Sistemi otvorenog koda za upravljanje IT konfiguracijama - Open Source CMDB Systems", Zbornik radova YU Info 2009, Kopaonik, Srbija. Available: http://www.e-drustvo.org/proceedings/YuInfo2009/html/pdf/165.pdf

[11] ISO IEC 82045-1, Document Management – Part 1: Principles and Methods, International Organization for Standardization, 2001.

[12] M. Shariff, "Alfresco Enterprise Content Management Implementation," 2006.

[13] S. Nikolić, V. Penca, and M. Zarić, "Rešenje za upravljanje it resursima bazirano na otvorenom kodu / Open Source Based Solution to IT Resources Management," Zbornik radova YU Info 2010,
Kopaonik, Srbija: 2010. Available: http://www.e-drustvo.org/proceedings/YuInfo2010/html/pdf/173.pdf

Acknowledgments

Results presented in this paper are part of the research conducted within the Grant No. III-44010, Ministry of Science and Technological Development of the Republic of Serbia.