Architecture and flexibility of digital libraries systems

Michal Kökörčený

Abstract**—***In the last decades digital libraries play an important role in knowledge sharing. Suitable architecture of any information system (not only a digital library) is important aspect for all institutions. From the point of view of an institution managing collection(s) of digital objects it is important to choose such system and architecture, which represents low costs of implementation and maintenance of the digital library system, possibility to easily integrate the system into the IS/ICT environment of the institution and sufficient flexibility of the information system for future changes. In this paper we will compare six widely used software systems from perspective of architecture of information systems. We will point out several problematic areas of contemporary digital libraries systems and we will propose certain possible trends of future development.*

Keywords**—**CDS Invenio, Digital library, DILLEO, DSpace, EPrints, Fedora, Greenstone, Service oriented architecture

I. Introduction

Comparison of digital libraries systems was discussed for example in the paper [1]. There were selected five open source systems and were defined several features of any digital library [1]. Selected systems were compared based on the previously defined characteristics.

In this paper we will compare several systems for building of digital libraries from perspective of architecture of information systems. In general, digital libraries systems are – in point of view of architecture – a kind of information systems. We will focus on comparison of these systems and theirs flexibility instead of comparing theirs features and functionalities.

II. Problem Formulation

Digital libraries are one of the most common services for seeking of information, especially in the education and scientific area. Digital libraries are controlled collections of information (digital objects) [2], including services for storing, manipulation, accessing, searching etc. The collections of information contain scientific, education, business or personal data [2] and digital objects can be represented as text documents, images, audio, video or other media files, executable applications and other digital content. Digital library systems provide specific functions for manipulating with these digital objects.

These features and specific requirements have influence on architecture and design of whole information system. From the point of view of an institution managing collection(s) of digital objects is important to choose such system and architecture, which represents: low costs of implementation and maintenance of the digital library system, possibility to easily integrate the system into the IS/ICT environment of the institution and sufficient flexibility of the information system for future changes. Almost in no organization we have information systems and applications without cooperation or communication with other systems. There are many systems and applications in every organization, nevertheless most of these systems are not well integrated [3], which causes many problems not only for users. Especially integration of legacy systems is very problematic. Nowadays, integration of existing and new implemented information systems is a key subject of interest for every organization.

In this paper, we have selected for comparison six widely used free digital library systems: Fedora, DSpace, Greenstone, EPrints, CDS Invenio and DILLEO. Furthermore we will discuss other approaches to architecture of (distributed) digital libraries systems.

III. Comparison of digital libraries systems

A. Fedora

Fedora (Flexible Extensible Digital Object Repository Architecture) was developed at Cornell University. It is architecture for storing, managing and accessing digital content in the form of digital objects [4]. Fedora is based on the principles of the Kahn-Wilensky Framework [5].

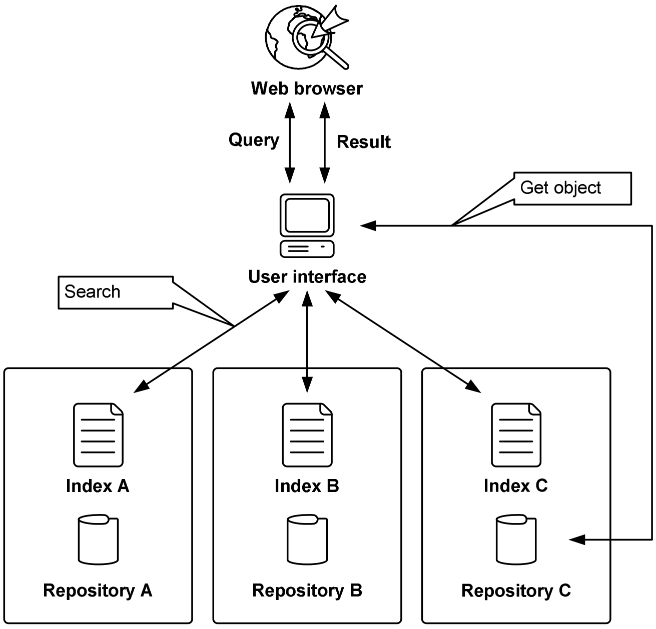

Fig. 1 Kahn-Wilensky architecture

In the Kahn-Wilensky framework have been defined basic elements and components of a digital library. In this architecture have been separated user interface and repository (repositories). Each repository provides operations for storing and accessing of digital objects through Repository Access Protocol (RAP).

The Fedora Repository Project is an open source implementation of the Fedora architecture. Nevertheless, Fedora does not contain the user interface (presentation tier). Fedora is not a complex system, which is possible simply to install and to use. An institution should implement the user interface by other means. Fedora system is implemented in Java language and can be run on many platforms such as Windows, Linux, Mac OS and others [6].

The main difference between other systems is that Fedora is based on principles of service oriented architecture (SOA). Service oriented architecture is a concept for building and integration of information systems and applications [7]. Fedora provides repository services exposed as web services with well-defined application interfaces via REST or SOAP protocols [4]. The system consists of three layers [8]:

- Web Services Exposure Layer,
- Core Subsystem Layer,
- Storage Layer.

The Web Services Exposure Layer consists of three related web services described using Web Services Definition Language (WSDL) [8]:

- Management Service (API-M) which defines an interface for administration of the repository,
- Access Service (API-A) which defines an interface for accessing digital objects stored in the repository,
- Access-Lite Service (API-A-Lite) which defines a reduced version of the Access Service (API-A).

The key feature of Fedora is that repository can store all types of digital content and its metadata [4]. Fedora provides high level of flexibility of digital content – not high level of flexibility of whole system. Fedora ensures access to the data objects through services described in behavior objects. Behavior objects contain metadata that describes operations and runtime binding [8]. Fig. 2 shows the principle of this object model.

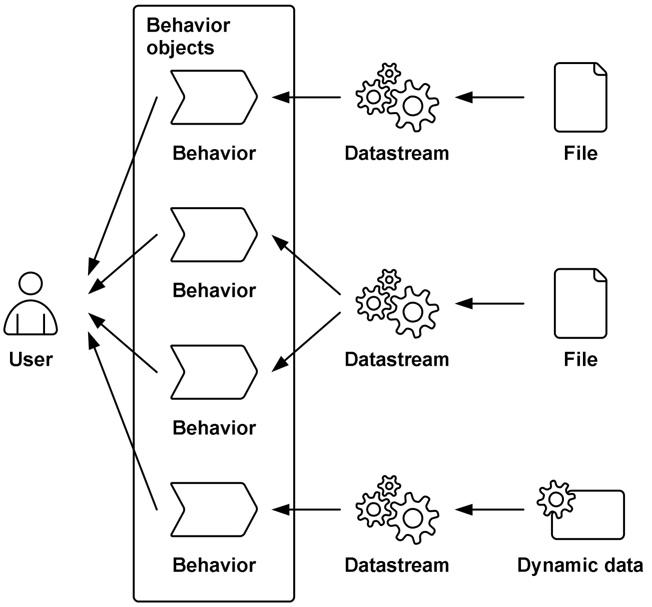

Fig. 2 Fedora object model [8]

Another difference to other systems is multi version support. It is possible to store and access several versions of a digital object in the repository. Fedora automatically preserves all versions of digital objects.

Repository supports Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH). Metadata are stored using Dublin Core (DC) standard and digital objects can be exported into METS (Metadata Encoding and Transmission Standard) format or into own FOXML (Fedora Object XML) format [6].

B. *DSpace*

DSpace is open source software for building of digital libraries, primarily intended for academic and research institutions. The system was developed in cooperation MIT (Massachusetts Institute of Technology) and HP (Hewlett Packard) [9]. DSpace is licensed under the BSD Open Source License and it is currently available for UNIX or OS/X platform [12]. Customization of the system requires comprehensive knowledge of this environment. DSpace does not work on Windows, Linux and other widely used platforms.

DSpace – in opposite to Fedora – is a complex software solution that includes repository and complete user interface. Therefore it is possible to deploy the software in a relatively short time period. DSpace uses as unique identifier Handle system (via URN), metadata are stored using Dublin Core (DC) standard [10], and metadata can be exported into METS format [11].

DSpace supports metadata searching as well as full text searching. Indexing of documents for full text searching is possible for these file formats:

- Plain text (TXT),
- Microsoft Word (DOC),
- Adobe Portable Document Format (PDF),
- Hyper Text Markup Language (HTML).

All these file formats – except plain text – are for purposes of indexing converted using filters into plain text. Indexing of documents is always performed on plain text data. If the system is extended with new filters, it will be possible to automatically index other file formats and document types [6].

DSpace supports “collections” for storing of data objects. Each data object must be inserted into at least one collection (or more). Fig. 3 shows the architecture of DSpace repository.

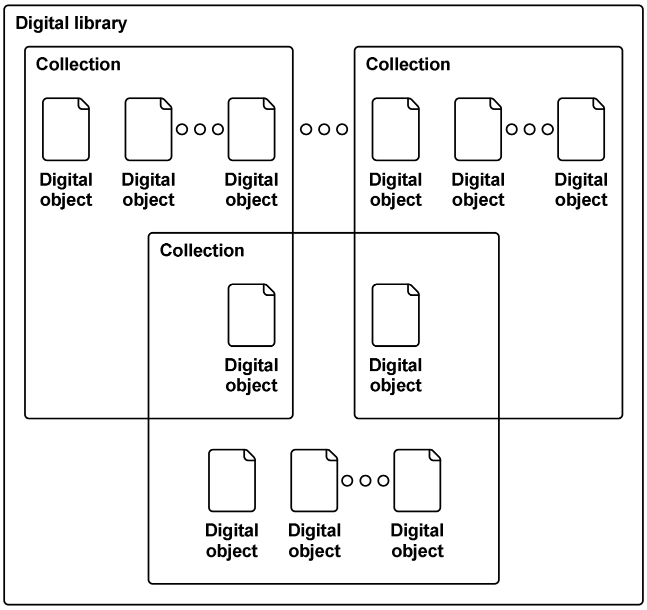

Fig. 3 Storing digital objects in DSpace

Furthermore, these collections may be divided into “communities”. Communities are organized into tree structure. That way DSpace provides means for dividing of digital sources into logical domains [6]. Due to these features DSpace differs from most of other digital libraries systems.

DSpace also supports simple workflow system for inserting new digital objects into repository. Newly inserted object is not published automatically and immediately – it must pass through an approval process, which contains a few predefined phases/steps (metadata approval, content approval). For each phase/step of the approval process is responsible certain (defined) user or certain group of users. The built in workflow system consists from five steps:

- Submit,
- Step 1 – Accept/Reject,
- Step 2 – Accept/Reject/ Edit metadata,
- Step 3 – Edit metadata,
- Archive.

It is possible to use any combination of steps 1, 2 and 3. Assigned user is notified by e-mail asking him to review the submission – the user will perform an action, the submission can be rejected or approved. Finally, when all steps were proceeded, the digital object is archived in repository and published. If no steps 1, 2 or 3 are defined, the digital object is archived and published immediately. The scheme of the current workflow mechanism that is built in DSpace is shown in Fig. 4.

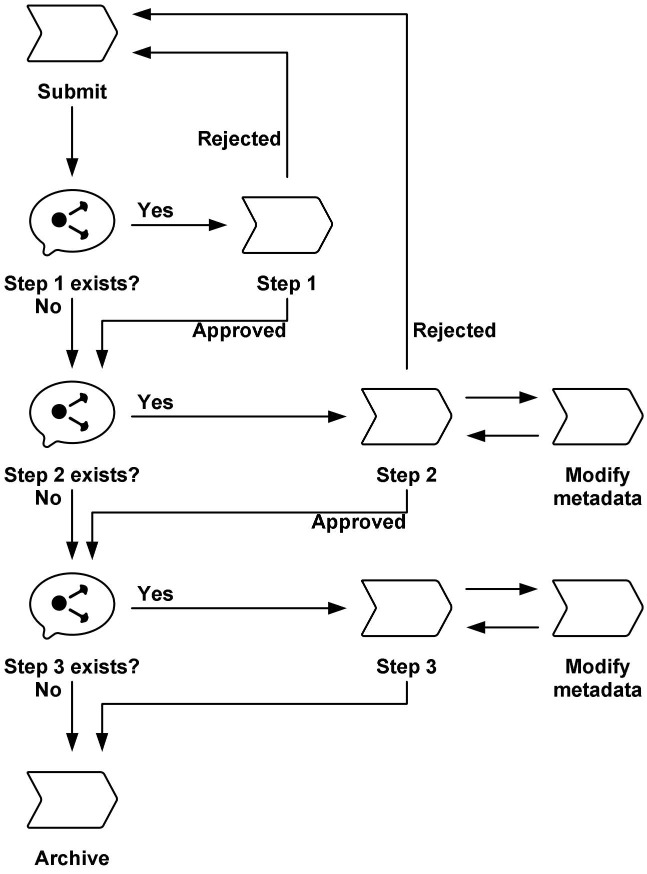

Fig. 4 DSpace approval process

An approval process is defined for a collection – not for whole repository or certain community. It means that each collection can have its own approval process [6]. A disadvantage of the DSpace digital library system is that the workflow system is built in and it is not possible to modify the approval process. Thus, you cannot define your own workflow mechanism.

The biggest disadvantage of DSpace digital library is low flexibility of whole system. DSpace is a monolithic system, which does not allow easy modifying certain parts of the software. DSpace is not based on principles of service oriented architecture (SOA).

C. Greenstone

Greenstone is a system for building and distributing digital library collections [13]. It is open source software released under the GNU General Public License. Greenstone has been developed at the University of Waikato in cooperation with UNESCO [13]. This system provides a way for organizing and publishing information on the internet as well as on removable media (e.g. DVD, CD etc.) – just this feature is not common for other digital libraries systems. Greenstone is the only widely used system, which supports distributing collections via removable media.

Greenstone supports Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH). Metadata are stored using Dublin Core standard and digital objects can be exported into METS format. Greenstone allows to use own metadata schema either by extending an existing one or by defining an entirely new one using a metadata set editor [13].

Digital objects are organized into "collections", which contain relating objects. A library contains one or more collections. Fig. 5 shows the architecture of Greenstone repository.

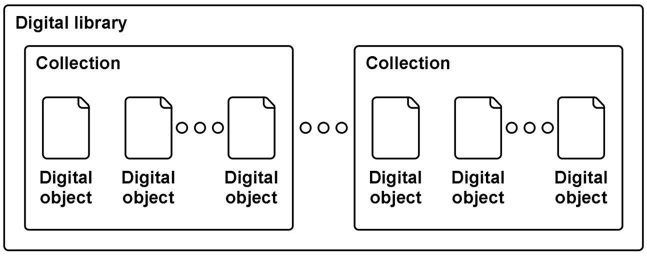

Fig. 5 Storing digital objects in Greenstone

Greenstone provides multi-language user interface containing several tens of language localizations [12].

D. EPrints

EPrints is open source software for building repositories of digital sources [14], primarily intended for scientific publications. EPrints has been developed at the University of Southampton.

For metadata storing the system uses its own internal format. In opposite to other digital library systems EPrints supports multiple "archives" under one instance of the software. An archive stands for standalone logical (not physical) digital library. So, you can have multiple digital libraries running on one instance of the software [6]. Fig. 6 shows the architecture of EPrints library.

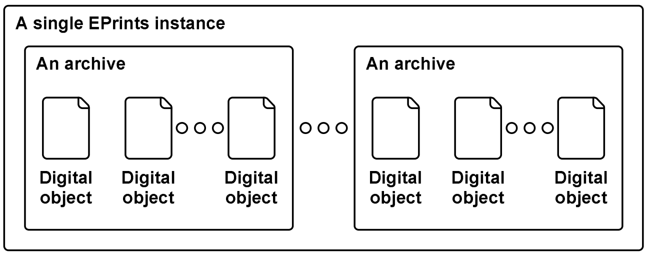

Fig. 6 Storing digital objects in EPrints

EPrints digital library system is based on statically generated pages of user interface (partially). This is just because of intention of the software – publishing of scientific resources. In this case it is not supposed, that digital objects are inserted or updated very often [6]. When a digital object was inserted or updated in a repository, all changes are visible after regenerating of user interface (static pages). Therefore there is a delay between editing and publishing of a digital object.

E. CDS Invenio

CDS Invenio is a complex system for building and managing digital libraries. CDS Invenio is developed at the CERN and it is primarily intended for purposes of this institution. It is free software licensed under the GNU General Public Licence (GPL) [15].

CDS Invenio uses MARC [16] format for metadata storing and allows defining mapping between other metadata formats. Similar to EPrints, this system uses statically generated pages of user interface [6]. All changes are visible after regenerating of user interface.

CDS Invenio is a comprehensive solution; nevertheless for many functions it is necessary to install third party products. This system is not based on principles of service oriented architecture (SOA). There are strong dependencies on other products, which produces tight coupling relations [6]. Furthermore it brings problems with incompatibility between different versions of these products. CDS Invenio is relatively flexible product, but on the other hand, modification of the system is usually complicated and often very expensive. In opposite to other digital library systems, CDS Invenio allows changing of searching algorithm including results presentation.

F. DILLEO

DILLEO is a digital library of sharable teaching materials primarily intended for faculty and students of higher education [17]. DILLEO was developed at University of Hradec Králové in cooperation with other partners and universities under project E-DILEMA as a part of European Union program SOCRATES/MINERVA.

DILLEO is designed and implemented as multi-language digital library system and provides user interface in different languages. Digital objects have SCORM compliant metadata format, interoperability is accomplished by OAI-PMH and SRU/SRW protocols.

The information system is based on three-tier architecture with thin client as web browser. The digital library is implemented on Microsoft .NET platform, presentation tier is created in Microsoft ASP.NET language, data tier is realized on Microsoft SQL Server. Fulltext searching is implemented by means of Microsoft Index Server. Logic tier consists from set of application objects, which ensure processing of requests received from presentation tier [17]. Fig. 7 shows the architecture of DILLEO digital library.

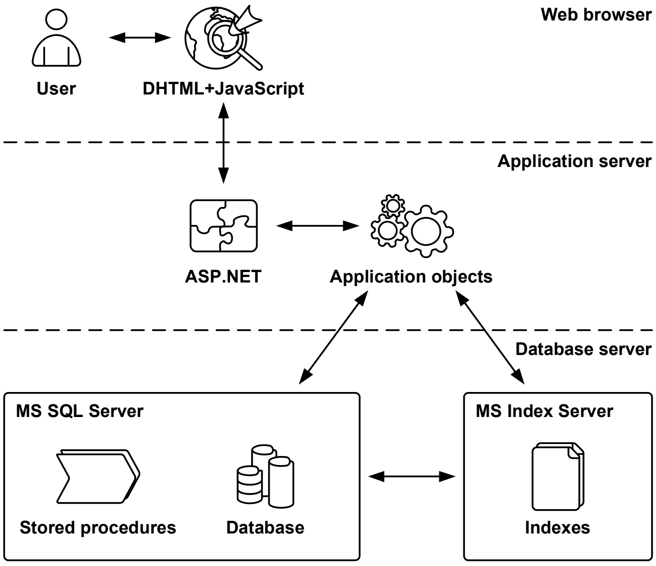

Fig. 7 The DILLEO architecture [17]

This is typical architecture of many contemporary information systems which produces tight coupled relations between application objects (components). Flexibility of such system is very low; it is very complicated and relatively expensive to customize certain parts and behavior of the software.

G. Other approaches

Generally, there exist many other approaches to architecture of digital libraries systems, especially of distributed digital libraries systems. Nevertheless there are only a few systems based on principles of service oriented architecture (SOA).

One of these approaches was proposed and presented for example in the paper [18]. Functionality of the system is partitioned into several well defined services with a well defined public interfaces, which define allowed requests and possible responses and exceptions [18]. The functionality of whole digital library is represented as the union of all functionalities of all individual services. Such system is much more flexible than typical monolithic systems. Fig. 8 shows basic architecture of the presented digital library system, which consists of these layers [18]:

- Front-End Layer,
- Repositories Layer,
- Index & Query Layer.

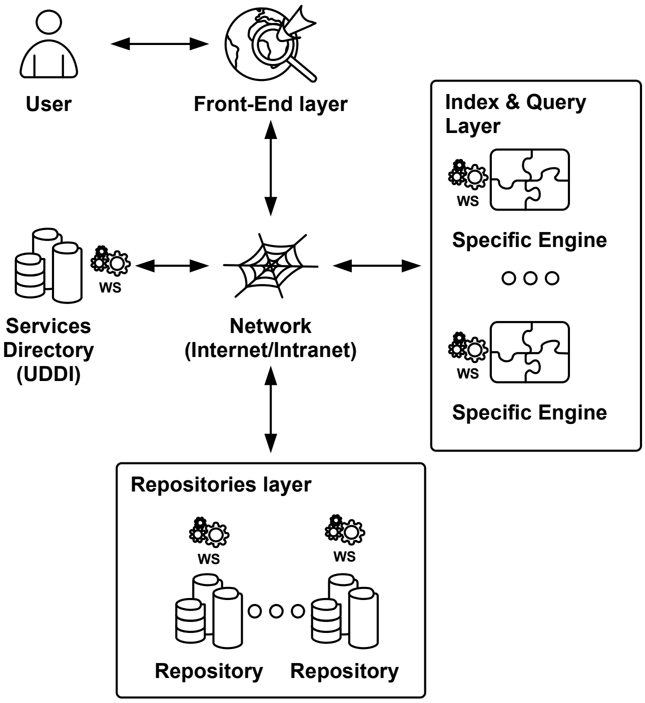

Fig. 8 Digital library architecture based on services [18]

The repositories layer consists of several repositories of digital objects, which can be based on various technologies. A repository has two main functions: to assure safe storage and preservation of the resources and to assure access to the resources [19]. Therefore each repository provides operations for storing and accessing of digital objects.

The index and query layer consists of several specific engines, which provide operations for indexing and searching on digital objects stored in the repositories layer [18].

The front-end layer consists of two types of interfaces. The first one represents the user interface of the system accessible via web browser. The second one is machine interface accessible through web services (WS). This layer is responsible for processing of all requests from the user and machine interfaces. The front-end layer implements the DAISS (Distributed Archive Index and Search System), which is responsible for requests distributing and results merging [18].

The services directory provides operations for registration and discovering of all web services. All services in the system are described using Web Services Description Language (WSDL) and are registered in the Universal Description, Discovery and Integration (UDDI) directory service.

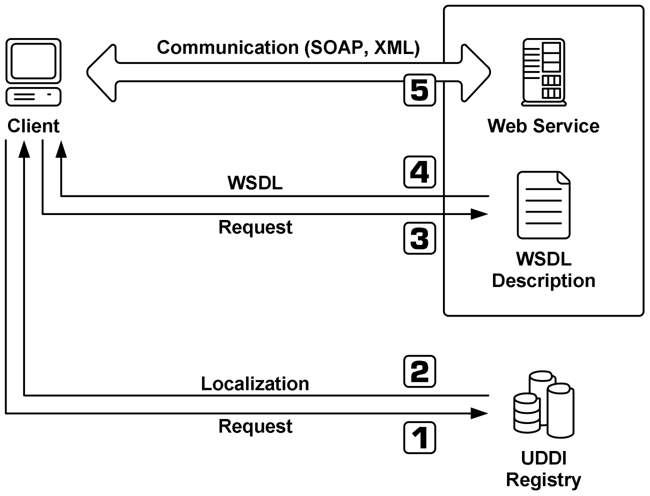

Fig. 9 Finding web services using UDDI registry

If a service consumer needs to call a service, first of all, the service is located and the communication protocol is determined using UDDI registry. Provided WSDL is passed to the consumer. WSDL describes provided operations, request messages, response messages and exceptions. Then, the service consumer directly connects to the service provider and sends a request message. The service provider sends the expected response message back to the service consumer. This mechanism is in service oriented architecture called "find-bind-execute" paradigm. Whole architecture is based on standard technologies, such as HTTP, SOAP, WSDL, UDDI, XML etc.

In this digital library framework all services communicate with each other according to the above described mechanism. In this way it is possible to dynamically discover what services are provided and are available. For communication can be used internet or intranet network.

This architecture allows building distributed digital library system, which is very flexible and allows easy customization its functionalities with low costs of implementation. It is possible to dynamically add new services as well as to modify current services. Architecture of this system produces loosely coupled relations between components (service providers and service consumers).

The Fedora is also based on principles of service oriented architecture (SOA); nevertheless there are significant differences to above described approach. Fedora provides repository services exposed as web services with well-defined application interfaces [4]. Fedora repository can store all types of digital content and its metadata [4] and provides high level of flexibility of digital content. The key difference between these approaches is, that Fedora – as whole system – is not built as distributed system using dynamically located services, which allows customizing any component and functionality of digital library.

IV. Conclusion

Most of the presented systems are specific oriented (e.g. EPrints or CDS Invenio), whereas usually respect needs and requirements of institutions, where were created. There are a few digital library systems, which are complex; nevertheless the software solution and architecture is not optimal and satisfactory. Table 1 contains comparison of architecture and features of the selected digital libraries systems.

TABLE I - COMPARISON OF DIGITAL LIBRARIES SYSTEMS

	Fedora	DSpace	Greenstone	EPrints	CDS Invenio	DILLEO
Multi-language support	-		*			*
Versioning	*					
User interface		*	*	*	*	*
Dynamically generated pages	*	*	*			*
Customizable searching					*	
Customizable metadata formats	*				*	
Flexibility of the system	*				*	
SOA principles	*					
Process based approach		*				
Removable media (DVD, CD…)			*			

In general, flexibility of described digital libraries systems is very low. Only Fedora is based on SOA principles, CDS Invenio is a flexible system; nevertheless customization is relatively complicated and may be rather expensive. None of these systems (except DSpace) support process based approach, typically for the approval process. Very problematic is also customization of searching algorithm and modifying of used metadata format. Any customization and modification of these systems may be complicated and represents high costs for an institution.

Furthermore, other problematic area is integration of a digital library system into the IS/ICT environment of the institution, for example integration with other learning management systems, document or content management systems etc. Digital libraries systems based on principles of service oriented architecture, such as Fedora or other SOA approaches; usually provide capability for integration through services exposed as web services with well-defined application interfaces via standard protocols (SOAP, REST etc.). Nevertheless there are a few other systems, such as DSpace or Greenstone, where integration can be accomplished for example through a standard message passing mechanism, typically as XML format over SOAP protocol [18].

A. Basic characteristics of digital libraries systems

Fedora – it is a flexible system based on principles of service oriented architecture (SOA). Nevertheless, flexibility is concerning only to content of the repository, Fedora is capable to store and access any digital content – this does not stand for flexibility of whole information system. Fedora does not contain user interface, it must be implemented by other means.

DSpace – is a complex system that contains repository including user interface. Nevertheless, DSpace is not flexible system – it is not possible to customize certain parts and behavior of the software.

Greenstone – is a multi-language digital library system and supports distributing collections via removable media. In this paper we discuss service oriented architecture, process based approach etc. – nevertheless contemporary technologies do not allow creating applications, which are based on these principles, which are multi-platform and which supports running from removable media.

EPrints – is primarily intended for scientific publications, where digital objects are not added or modified very often. This system uses statically generated pages of user interface. EPrints supports multiple archives under one instance (multiple logical libraries).

CDS Invenio – is a complex and flexible system, which uses third party products for certain functionalities. Customization of the system is complicated and may be rather expensive. There are strong dependencies on other third party products, which can cause problems with incompatibility between different versions. CDS Invenio is not based on principles of service oriented architecture.

DILLEO – is primarily intended for educational purposes and it is designed and implemented as multi-language digital library. The system represents typical three-tier architecture with thin client. The application consists from set of components producing tight coupled relations. DILLEO – similarly to the most of other digital library systems – does not support service oriented architecture and does not stand for flexible system.

Other approaches – there exist many other approaches to architecture of (distributed) digital libraries systems. We have described the architecture which was presented in the paper [18]. This approach is completely based on principles of service oriented architecture (SOA) and whole system is much more flexible than typical monolithic systems and allows easy customization its functionalities. It is possible to dynamically add new services as well as to modify current services.

B. *Summary*

Typical disadvantages of contemporary digital library system in opposite to contemporary company information systems are:

- not enough flexible architecture,
- tight coupling of application components (instead of loose coupling architecture),
- higher costs of customization of a digital library system,
- digital libraries usually cannot share application components with other information systems,
- difficult integration of a digital library system with other information systems and applications in an institution,
- higher costs of this integration,
- usually there is not process based approach.

Contemporary digital libraries are very often monolithic systems/applications, which cannot fulfill new requirements on information systems.

C. *Recomendations*

New digital libraries systems and theirs architecture should follow ways and trends of development in the area of company information systems. Actually it comprises these concepts and principles:

- service oriented architecture (such as Fedora),
- process based approach (such as DSpace),
- automated processes including human workflow (human tasks),
- standard infrastructure and technologies (e.g. Enterprise Service Bus etc.).

Service oriented architecture (SOA) is a widely accepted approach for building and integrating of information systems and applications [20]. Service oriented architecture is a set of principles used during analysis, design, implementation and integration phases of information systems development, which produces coarse grained, loosely coupled services (code). Service oriented architecture differs from the more general client/server model in its emphasis on loose coupling between software components [21].

Services represent basic building blocks (components), which implement certain functionalities accessible over a computer network. These services can be combined to create new systems, reflecting new requirements on functionalities of software solutions. In the area of digital libraries these approaches were not applied (except a few systems).

Service oriented architecture can improve the flexibility of the software solution and management of an organization with lower costs of implementation and maintenance. Service oriented architecture provides an opportunity to achieve broad scale interoperability while offering flexibility to adapt to changing technologies and requirements [20]. In general, service oriented architecture can offer the following benefits [20]:

- loosely coupled applications and location transparency,
- application connectivity and interoperability,
- alignment of IT around the needs of the business,
- enhanced reuse of existing assets and applications,
- process centric architecture,
- parallel and independent development,
- better scalability and graceful evolutionary changes,
- reduced costs of application development and integration,
- easier maintenance.

One of the many possibilities how to achieve high flexibility of the whole system is to implement a digital library as a composite (SCA) application. Service component architecture is a relatively new concept or framework for creating applications (composites) built from services [22]. Fig. 10 show basic SCA concept how it is possible to build applications from services.

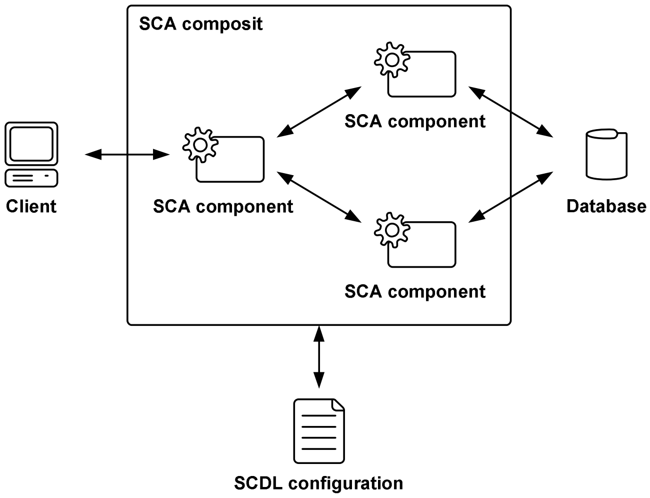

Fig. 10 Basic SCA concept [22]

Services are basic elements of SCA composites which perform specific functionalities. Services may be implemented using different technologies and different programming languages [22]. SOA/SCA based architecture provides loosely coupled suite of services which can reduce costs of system modification. See [22]-[24] for more details about this concept.

REFERENCES

[1] G. Pyrounakis, M. Nikolaudou, “Comparing Open Source Digital Library Software“ in *Handbook of Research on Digital Libraries: Design, Development, and Impact*, IGI Global, 2009, pp. 51-60.

[2] E. F. Martinez, S. Y. Chen, “Evaluation of User Satisfaction with Digital Library Interfaces“ in *Proceedings of the 5th WSEAS International conference on Simulation, Modeling and Optimization*, Corfu, 2005, pp. 172-177.

[3] S.-Y. Chen, C.-N. Chang, “Architecture for Personal Digital Library“ in *ICCOMP'05 Proceedings of the 9th WSEAS International Conference on Computers*, Wisconsin, 2005.

[4] Fedora, Flexible Extensible Digital Object and Repository Architecture [Online]. Available: http://fedora-commons.org

[5] R. Kahn, R. Willensky, “A Framework for Distributed Digital Object Services“ [Online]. Available: http://www.cnri.reston.va.us/home/cstr/arch/k-w.html

[6] V. Krejčíř, “Systémy pro tvorbu digitálních knihoven“ in *Proceedings of the INFORUM 2006: 12. konference o profesionálních informačních zdrojích*, 2006.

[7] J. Davies, D. Schorow, S. Ray, D. Rieber, *“The Definitive Guide to SOA: Oracle Service Bus, Second Edition“*, Apress, 2008.

[8] T. Staples, R. Wayland, S. Payette, ”The Fedora Project – An Open-source Digital Object Repository Management System” in *D-Lib Magazine*, vol. 4, number 4, Apr. 2003.

[9] DSpace, Digital Archive Project [Online]. Available: http://dspace.org

[10] Dublin Core Metadata Initiative [Online]. Available: http://dublincore.org

[11] Metadata Encoding & Transmission Standard [Online]. Available: http://www.loc.gov/standards/mods/

[12] I. H. Witten, D. Bainbridge, R. Tansley, C. Huang, K. J. Don, “StoneD – A Bridge between Greenstone and Dspace” in *D-Lib Magazine*, vol. 11, number 9, Sep. 2005.

[13] Greenstone digital library software [Online]. Available: http://www.greenstone.org

[14] Open Access and Institutional Repositories with EPrints [Online]. Available: http://www.eprints.org

[15] CDS Invenio [Online]. Available: http://cdsware.cern.ch/invenio/index.html

[16] MARC Standards [Online]. Available: http://www.loc.gov/marc

[17] S. Mikulecký, *“Digital library of learning objects“*, University of Hradec Králové, 2005.

[18] H. T. Zagalo, J. P. Martins, P. M. Almeida, J. S. Pinto, “A Contribution of Open Source Technologies to Support Distributed Digital Library’s Repository and Index Services“ in *Proceedings of 4th WSEAS Conference on Mathematics and Computers in Business and Economics*, Tenerife, 2003.

[19] J. Borbinha, “An Infrastructure for a National Digital Library“ in *Proceedings of the 5th WSEAS International conference on Simulation, Modeling and Optimization*, Corfu, 2005, pp. 146-151.

[20] Z. Mahmood, “Service Oriented Architecture: Potential Benefits and Challenges“ in *Proceedings of the 11th WSEAS International Conference on Computers*, Crete Island, 2007, pp. 497-501.

[21] S. Kambhampaty, S. Chandra, “Service Oriented Architecture for Enterprise Applications“ in *Proceedings of the 5th WSEAS International conference on Software Engineering, Parallel and Distributed Systems*, Madrid, 2006, pp. 48-54.

[22] J. Marino, M. Rowley, *“Understanding SCA (Service Component Architecture)“*, Addison-Wesley, 2010.

[23] SCA Service Component Architecture, Building Your First Application – Simplified BigBank [Online]. Available: http://www.osoa.org/download/attachments/28/SCA_BuildingYourFirstApplication_V09.pdf

[24] SCA Service Component Architecture, Assembly Model Specification [Online]. Available: http://www.osoa.org/download/attachments/35/SCA_AssemblyModel_V100.pdf