Current usage of Component based Principles for Developing Web Applications with Frameworks: A Literature Review

Novak, Matija; Švogor, Ivan

Source / Izvornik: **Interdisciplinary description of complex systems, 2016, 14, 253 - 276**

Journal article, Published version
Rad u časopisu, Objavljena verzija rada (izdavačev PDF)

https://doi.org/10.7906/indecs.14.2.14

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:880413

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**



Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository





Interdisciplinary Description of Complex Systems 14(2), 253-276, 2016

CURRENT USAGE OF COMPONENT BASED PRINCIPLES FOR DEVELOPING WEB APPLICATIONS WITH FRAMEWORKS: A LITERATURE REVIEW

Matija Novak* and Ivan Švogor

Faculty of Organization and Informatics, University of Zagreb
Varaždin, Croatia

DOI: 10.7906/indecs.14.2.14
Review article

Received: 4 September 2015.
Accepted: 3 March 2016.

ABSTRACT

Component based software development has become a very popular paradigm in many software engineering branches. In the early phase of Web 2.0 appearance, it was also popular for web application development. From the analyzed papers, between this period and today, use of component based techniques for web application development was somewhat slowed down, however, the recent development indicates a comeback. Most of all it is apparent with W3C's component web working group. In this article we want to investigate the current state of web application development with component approach. Most of all we are interested in which way components are used, which web development frameworks are being used, for which domains is component based web development most popular and successful, etc. How many current web development frameworks explicitly refer to component-based approach? To answer this question, we performed a literature review.

KEY WORDS

component based development, CBD, Web application, WWW, Frameworks

CLASSIFICATION

ACM: H.1.2, H.3.4., H.4.m
JEL: L86

*Corresponding author, η: matnovak@foi.hr; +385 42 390 864;
FOI, Pavlinska 2, HR – 42 000 Varaždin, Croatia

INTRODUCTION

Creating complex software architecture by (re-) using smaller, more manageable, software elements is the main goal of component based development (CBD). In many cases it has proven to simplify software design and have a positive impact on extra-functional properties of software products, e.g. better maintainability, scalability, reliability, usability, etc. Since the research on World Wide Web related development is constantly growing, we find interesting to verify in which way are these two research areas are related. To satisfy our curiosity we conducted a *literature review* in which we analyzed how web researchers and practitioners apply existing component based development techniques to create architecture of their web applications. Since most of the web applications are currently developed using different web development frameworks, we are also interested how many frameworks are based on CBD. Therefore, the main research question is: "*How many of the current web application development frameworks explicitly refer to application of component-based approach?*" – Based on this question we derive several more specific questions:

- Q_1 – In which way is CBD used for web application development?
- Q_2 – What is the relation between CBD and web application development?
- Q_3 – Which component models are used for web application development?
- Q_4 – In which web application development domains is CBD used?

The rest of the article is organized as follows: in Section 2 describes the review protocol and all the related methods used to perform a literature review. Section 3 provides an overview of the paper selection process. In Section 4 we present a detailed analysis of the results and discuss them. Finally, Section 5 concludes the article.

REVIEW PROTOCOL

The review protocol of this study is based on the work of Breivold et.al. [1] and suggestions by Kitchenham [2]. It consists of the following steps (Figure 1): a) motivation statement, b) research goal statement, c) defining the search terms, d) providing information on restrictions and selection criteria, e) database selection, f) paper search process, g) paper quality assessment, h) paper data extraction, i) data synthesis. Since steps a) and b) are given in the previous section, here we proceed with steps c) – e) which are given in the Table 1.

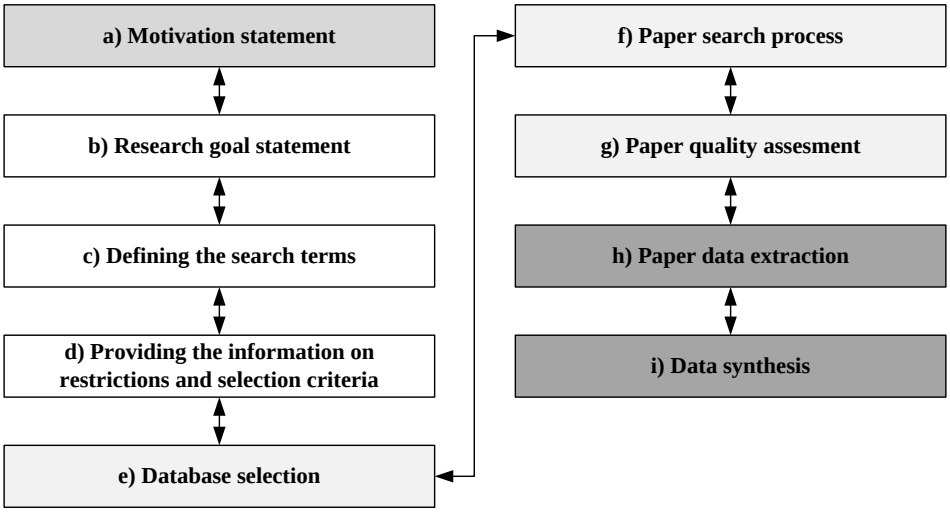


Figure 1. Overview protocol.

SEARCH TERMS, SELECTION CRITERIA AND DATABASE SELECTION

For more effective selection of relevant papers we used Mendeley, a reference management tool [3]. Every database listed in Table 1provides a way to extract and import the search results into such tools. Since ACM is an exception, we imported the search results manually. Duplicated were automatically excluded or merged.

To additionally ensure that the quality of the papers is satisfactory, there are some additional criteria [1]: a) paper must provide evidence for claims and theoretical reasoning in data analysis, b) paper must have the description of the context in which the study was conducted, c) the research method must be described or easily inferred, and finally d) goals of the study need to be described or easily inferred.

Table 1. Literature review protocol steps c)-e).

Search terms (using OR operator)	**Databases**	**Restrictions**
"component based" and web and development *"component based" and web and architecture* *"component-based" and web and development* *"component-based" and web and architecture* *"CBD" and web and development* *"CBD" and web and architecture*	SCOPUS ACM Digital Library IEEE Xplore Science Direct ISI Web of Science	Papers that are included are form 2004 to 2014 (including papers that are in the databases at 15 January 2014) Papers that are written in English Only full papers are considered No duplicates should be included Peer-reviewed papers in journals, conferences and workshops Only studies that explicitly refer to use component based approach in web application development Only studies that explicitly refer to using or making a framework Search process & quality assessment

DATA EXTRACTION

Once the final list of papers was obtained, a content analysis and review was performed while considering [1]:

(a) general paper information (title, authors, publication year, source, publication type, citation information, research methodology, and data analysis type) and

(b) content related information (research focus, area of CBD application, applied CBD models, software application domain (what kind of software, who uses it, etc.), programming languages used, problems with applying CBD, area of future work).

The result of reading the papers are synthesized in the following sections.

SEARCH OVERVIEW

DOMAIN SPECIFIC REVIEW CRITERIA

Based on the research area we decided to exclude papers which refer to service oriented architecture (SOA), semantic web and ontology and finally papers which refer to service oriented computing (SOC).

Since the search terms resulted in 166 papers related. Also, as stated by Bano and Ikram, SOA is a shift of paradigm in software development, as it can be seen in application of web services instead of using commercial off-the-shelf software [4]. Having this in mind we decided to exclude SOA related papers as a whole new literature review can be performed with SOA as a main topic.

Semantic web and ontology related papers are also removed because these principles are used for all kind of applications, not necessarily web applications. Although, some papers corresponded to the initial criteria, we decided to leave out ones which relate to semantic web, as this research area is growing and very specific.

Finally papers which refer to service oriented development, service oriented computing, web-service based applications and development of web-services were left out as they are mostly related to development techniques and SOA. We conclude that similarly as for SOA, this research area would deserve its own separate literature review.

Since some papers are excluded in this phase because they are out of scope, they will not be addressed further. However, we would like to refer the interested readers to the sources: [5-8].

DATABASE QUERIES AND SEARCH RESULTS OVERVIEW

Table 2 shows the overview of the search results. By applying the exclusion criteria from the initial result of 1132 papers only 29 were selected for full reading.

At this point we also verified the validity of research queries. We performed a search without "CBD" term, as presented in Table 2. One can notice that there is no significant difference when this term is left out and "component-based" is used. Finally, by using the terms "component" or "components" alone 7696 results were found in Scopus only. One can easily

Table 2. Search overview.

Database	**Findings without CBD keyword**	**Findings without restrictions (Language, Year) with "CBD" keyword**	**2004 - 2014**	**Only English**
Scopus	677	691	567	548
ACM	291	296	211	211
IEEE	325	326	240	240
Science Direct	33	34	24	24
Web Of Science	141	146	109	109
Total	*1467*	*1493*	***1151***	***1132***
After removing of all duplicates, Mendeley automatically				*761*
After removing of all duplicates, Mendeley "Detect Duplicates" option				*733*
After removing of covers (journal, conference proceedings covers)				*621*
After removing all duplicates, found manually by reading titles and abstracts				*610*
After removing all unrelated papers, found manually by reading titles and abstracts				*116*
After second reading of abstracts and titles				*106*
After removing all papers which do not explicitly refer to application of a framework (reading introduction and conclusion)				33
After the second complete reading of all papers				***27***
Number of papers which are accessible (full text of 3 papers wasn't found in any database)				***24***

conclude that this terms cannot be used by themselves and considering only the keywords the following query was used:

TITLE-ABS-KEY((web AND development) OR (web AND architecture)) AND TITLE-ABS-KEY("component based" OR "component-based" OR "component" OR "components")

The exact search queries with all the restrictions are available in Appendix A of this paper. Appendix B contains the list of final 29 papers labeled [A1] – [A29].

From the final 29 papers which were selected for full reading, 3 of them could not be accessed for full reading. These papers are [A1]-[A3] and because they were unavailable, they are not present in all parts of this review. Also, after all the papers were fully read, two additional ones were excluded which left as with final 24 papers [A4]-[A27]. The two excluded papers are out of scope. Paper [A28] is related to quality assessment and not web development while paper [A29] is business oriented, but mentions frameworks.

RESULT ANALYSIS

DATABASES AND YEAR DISPERSION

As it can be seen by the right data bar in Figure 2, considering the number of papers found, Scopus is the most inclusive database with 22 of final 24 papers (and with 7 found only in Scopus), while Science Direct was the most exclusive with only 1 paper. The remaining two papers not found in Scopus, first was found only in IEEE [A23], while the second one [A20] was in ACM and Web of Science. The number of papers included in the database which were selected for final reading (the original 27 papers, which includes 3 inaccessible) is shown on the left data bar. Here, Scopus was also the most inclusive database with 25 of 27 papers. The two missing papers are the same as previously and both of them are found in IEEE Xplore. Therefore, combining Scopus and IEEE Xplore gives most of the relevant papers in this research area.

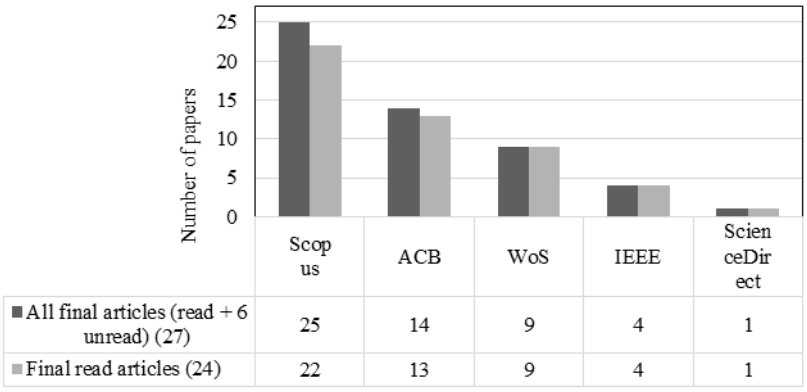
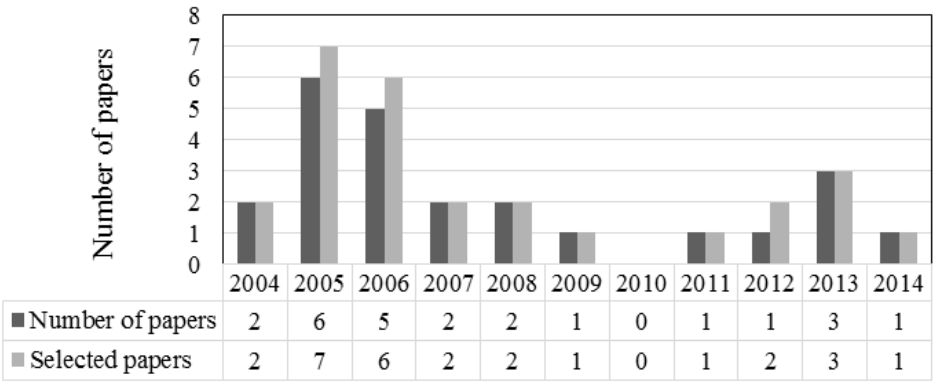


Figure 2. Papers in database.

The database search was performed in mid-January of 2014 and it was set to include papers from previous 10 years, i.e. 2004-2014. Figure 3 shows the dispersion of papers between years. As it can be seen, most of the papers, i.e. around 50% of them, were written between 2005 and 2006. This should be a consequence of Web 2.0 which was popularized by Tim O'Reilly in late 2004 at the O'Reilly Media Web 2.0 conference. At that time there was an increase of web application development and since web applications started growing more complex, it was necessary to ease the development process and perform research in this direction. The solution was found in web development frameworks and component based techniques (as one of the options). Earlier papers are focusing on theoretical aspects while later on they became more practical providing different benchmarks and case studies.

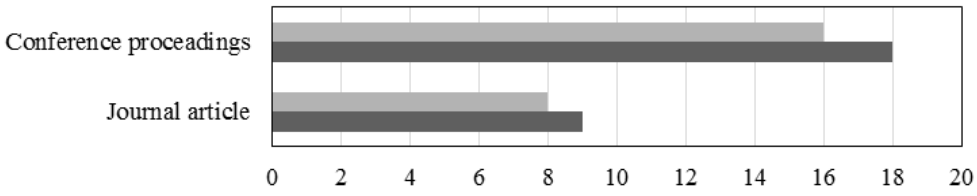


	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
Number of papers	2	6	5	2	2	1	0	1	1	3	1
Selected papers	2	7	6	2	2	1	0	1	2	3	1

Figure 3. Papers per year.

PUBLICATION TYPES

Figure 4 shows publication types. Two-thirds of selected papers are conference proceedings while one-third of remaining papers were published in journals. List of conference proceedings and journals is presented in Table 3 and Table 4.



	Journal article	Conference proceedings
Selected papers	8	16
All papers	9	18

Figure 4. Papers per publishing type.

It is interesting to notice that every paper which was selected comes from a different journal and therefore there isn't any conclusion on which journals to follow for this particular topic. However, the conferences are a bit more conclusive. Table 3 shows that two papers come from 13th International WWW Conference Proceedings 2005 while 6 papers are published in Springer's Lecture Notes. This indicates that Springer database should also be included in future investigation of this area.

Table 3. List of journals.

	JOURNALS
1	Journal of Shanghai University
2	Computer Science - Research and Development
3	Cartographica
4	IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)
5	Journal of Intelligent Information Systems
6	Decision Support Systems
7	Digital Creativity
8	International Journal of Computer Integrated Manufacturing
	JOURNALS OF PAPERS THAT WERE NOT READ IN FULL
9	Advanced Materials Research

Table 4. List of conference proceedings.

	CONFERENCE PROCEADINGS	PUBLISHER
1	Thirteenth International World Wide Web Conference Proceedings, WWW2004	ACM
2	Lecture Notes in Computer Science	Springer
3	Proceedings - International Conference on Next Generation Web Services Practices, NWeSP 2005	IEEE
4	Web3D Symposium Proceedings	ACM
5	Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining	ACM
6	Proceedings of the ACM Symposium on Applied Computing	ACM
7	Lecture Notes in Computer Science	Springer
8	Lecture Notes in Engineering and Computer Science	Springer
9	Thirteenth International World Wide Web Conference Proceedings, WWW2004	ACM
10	Software Engineering and Advanced Applications, 2005. 31st EUROMICRO Conference	IEEE
11	Lecture Notes in Computer Science	Springer
12	ENASE 2013 - Proceedings of the 8th International Conference on Evaluation of Novel Approaches to Software Engineering	Springer
13	2008 34th Euromicro Conference Software Engineering and Advanced Applications	IEEE
14	Lecture Notes in Computer Science	Springer
15	5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering,Software Architecture and Reuse (ICIS-COMSAR'06)	IEEE
CONFERENCE PREOCEADINGS OF PAPERS THAT WERE NOT READ IN FULL		
16	Proceedings of the IASTED International Conference on Internet and Multimedia Systems and Applications	IASTED
17	ICEIS 2005 - Proceedings of the 7th International Conference on Enterprise Information Systems	ICEIS

RESEARCH METHODS AND TYPES

Figure 5 presents the information about most common research methods used. In 5 papers authors explicitly state that they are performing a case study while in other papers, the applied research method was inferred from the context. It turns out that case study is used by 70 % of the papers, followed by theoretical reasoning, experiment and action research.

Considering the type of the study presented in papers, Figure 6 shows that qualitative reasoning is the most popular one. 20 papers use qualitative reasoning, while of the remaining 4, 2 of them use quantitative study and 2 of them use a mixed approach, i.e. qualitative and quantitative.

Table 5 provides an insight into the relation between the type of the study and the research method used. As it can be seen in the Table 5, most papers use qualitative study type performed on a case study.

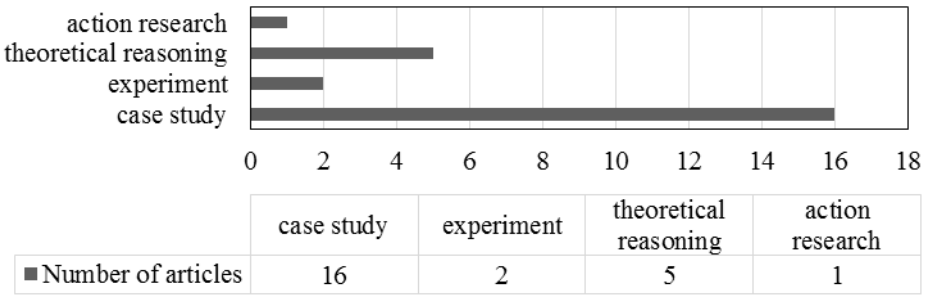


Figure 5. Research methods.

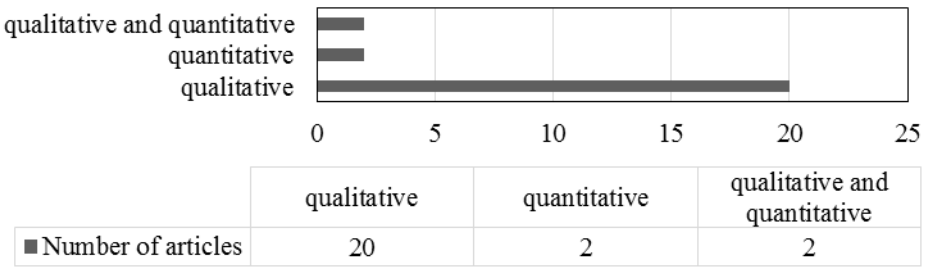


Figure 6. Study types.

Table 5. Type of study and research method by paper (continued on p.261).

Paper	Year	Type	Research method	Validation
[A10]	2004	qualitative	theoretical reasoning	AMACONT project used for developing component-based adaptive web presentations. Theoretical descriptions.
[A15]	2004	qualitative	case study	Simple demo applications
[A7]	2005	qualitative	case study	Authors created one e-learning course
[A11]	2005	qualitative	case study (explicitly mentioned)	Framework was developed and used on prototype.
[A16]	2005	quantitative & qualitative	action research	Authors developed a framework and used it in Rila Solution Company in Finance and Crediting domain.
[A17]	2005	qualitative	case study	Developed a prototype of a framework for Real Time Distributed Control Systems (RTDCS)
[A21]	2005	qualitative	theoretical reasoning	Theoretical explanations
[A25]	2005	qualitative	theoretical reasoning	Theoretical proposal of a framework
[A8]	2006	qualitative	case study	Prototype was created
[A9]	2006	qualitative	case study (explicitly mentioned)	Demonstration through various examples.
[A13]	2006	qualitative	case study	GOWARN concept from 2003 extended to new locations (AIS (atlas information system) for Campi Flegrei, a volcanic area near Naples, Italy).

Table 5. Type of study and research method by paper (continuation from p.260).

Paper	Year	Type	Research method	Validation
[A23]	2006	qualitative	case study	Math Test site was built as a prototype.
[A24]	2007	qualitative	case study	Method was used on prototype system National Agriculture Decision Support System (NADSS).
[A27]	2007	qualitative	case study	"…, several Web-Based product customization systems are developed for typical industrial enterprises." [A27]
[A19]	2008	qualitative	case study	Real world applications, exp. "Portal e-Arte"
[A20]	2008	quantitative	case study (explicitly mentioned)	Large scale web application DSS that includes 2000 component and 500 pages.
[A5]	2009	quantitative	Experiment	Benchmark in using Jade Framework for construction of self-repairable cluster systems.
[A6]	2011	quantitative & qualitative	Experiment	Authors performed tests (login case).
[A14]	2012	qualitative	case study	Prototype was built "scrap exchange system"
[A12]	2013	qualitative	case studies (explicitly mentioned)	Various case studies in BMD Systemhaus GmbH company
[A18]	2013	qualitative	theoretical reasoning	Theoretical explanation of the model
[A26]	2013	qualitative	theoretical reasoning	Theoretical explanation
[A4]	2014	qualitative	case study	Prototype was built eCommerce, shopping chart application

CITATION

The citation count of the selected papers (including the ones not accessible) is shown in the Table 6. Citing information was taken from Google Scholar in late April of 2014. The most cited paper is [A26] which has 56 citations. Several of the following papers have relatively good citation record, but on average there are 8 citations per paper. If we exclude the papers published in last four years (because they are fairly recent), the average number of citations per paper is 11. Therefore, we can conclude that research in component based web applications needs some further investigation due to relatively low publication and citation count. However, due to exclusion of SOA related papers which is becoming hugely popular research area, it is most likely that this effected the number of publications related to CBD and web development frameworks. Also, another limiting factor is the strong search criteria which states that the framework must be explicitly mentioned.

Table 6. Papers sorted by citation level.

Paper	Title	Year	Cited by
[A24]	A software architecture and framework for Web-based distributed Decision Support Systems	2007	56
[A10]	Design and Implementation of Component-based Adaptive Web Presentations	2004	26
[A15]	XVM: A Bridge between XML Data and Its Behavior	2004	16
[A11]	Towards end user development of Web applications for SMEs: A component based approach	2005	16
[A8]	Using the Amacont Architecture for Flexible Adaptation of 3D Web Applications	2006	15
[A19]	The WebComfort Framework: An Extensible Platform for the Development of Web Applications	2008	11
[A13]	Modular web-based atlas information systems	2006	9
[A4]	openMVC: A non-proprietary component-based framework for web applications	2014	9
[A22]	Modeling a web-based remote monitoring and fault diagnosis system with UML and component technology	2006	8
[A27]	Realization of a development platform for Web-based product customization systems	2007	7
[A20]	Shared Content Management in Replicated Web Systems: A Design Framework Using Problem Decomposition, Controlled Simulation, and Feedback Learning	2008	6
[A5]	A self-repair architecture for cluster systems	2009	6
[A9]	A component-based framework for knowledge discovery in bioinformatics.	2006	3
[A21]	Structural engineering: Processes and tools for developing component-based open hypermedia systems	2005	2
[A23]	A Framework for Developing Wireless Mobile Online Applications	2006	2
[A14]	A Web Application Framework for End-User-Initiative Development with a Visual Tool	2012	2
[A26]	The pataphysics of creativity: Developing a tool for creative search	2013	2
[A12]	Composing user-specific web applications from distributed plug-ins	2013	1
[A25]	A study on frameworks of component integration for web applications	2005	0
[A7]	Developing Customizable Web-based Educational Applications through a Component-based Framework	2005	0
[A16]	Towards development and use of in-house component framework: Results and expectations	2005	0
[A17]	Loose integration of COTS tools for the development of real time distributed control systems	2005	0
[A1]	A non proprietary framework for policy controlled management of the model in the MVC design paradigm (NOT READ)	2005	0
[A2]	A circuit board approach to building web applications (NOT READ)	2006	0
[A6]	Support for development and test of web application: A tree-oriented model	2011	0
[A3]	Component based webGIS development framework (NOT READ)	2012	0
[A18]	Nested web application components framework: A comparison to competing software component models	2013	0

In Figure 7, which presents number of cited papers by year, it can be noticed that most of the citations are from between 2005 and 2006, more than 50 %. But surprisingly the most cited paper is from 2007, and the more recent one, from 2014 is cited 9 times. It will be interesting to see if the rising trend as seen from 2011 up to 2014 will continue.

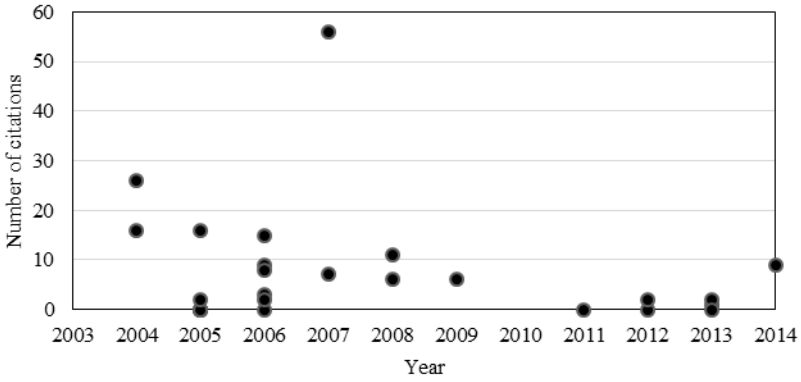
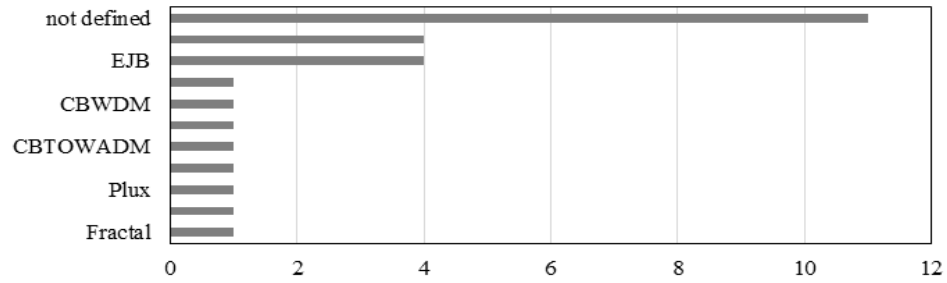


Figure 7. Citation level of papers by years.

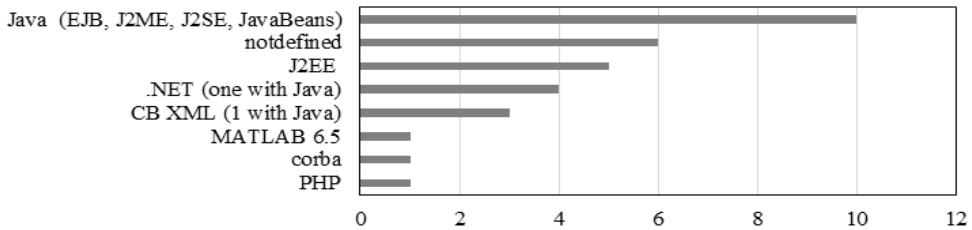
USED CBD MODELS AND PROGRAMING LANGUAGES

In Table 7 and Figure 8 one can notice that 11 out of 24 papers haven't defined a component model. The most used component model is some variation of JavaBeans consequently making Java the most popular development language (usually J2EE).



	Fractal	Corba	Plux	COM	CBTOWADM	CONTIGRA	CBWDM	XVM	EJB	JavaBeans	not defined
Component model	1	1	1	1	1	1	1	1	4	4	11

Figure 8. Component models.



	PHP	corba	MATLAB 6.5	CB XML (1 with Java)	.NET (one with Java)	J2EE	notdefined	Java (EJB, J2ME, J2SE, JavaBeans)
Programing language	1	1	1	3	4	5	6	10

Figure 9. Programming languages.

Both JavaBeans and EJB are used in 4 papers. In 3 of them, JavaBeans and EJB are used simultaneously, 2 papers use COM and Corba component models which are well known, and the remaining ones use custom component models.

Considering programming languages (Figure 9), Java is most widely used, with 12 out of 21 papers using it. Among these, 10 of which using explicitly Java and in 2 paper Java is an option (papers [A21] and [A12]). Paper [A15] also uses Java, but only to build XML which is then used to develop web applications, therefor it is not counted since this XML can be generated in many programming languages.

Table 7. Overview component model & programing language.

Paper	Component model	Programing language
[A4]	not defined	.net, J2EE
[A5]	Fractal	Java
[A6]	CBTOWADM	note defined
[A7]	JavaBeans, EJB	J2EE
[A8]	Contigra	XML
[A9]	not defined	J2SE
[A10]	not defined	XML
[A11]	not defined	not defined
[A12]	Plux (plug and play like OSGi or SOFA 2.0)	.net (can be implemented in Java)
[A13]	EJB	Java, EJB
[A14]	not defined	Java Servlets with JSON
[A15]	XVM	Java for build XVM Framework, XML for building Web applications
[A16]	JavaBeans, EJB	J2EE, EJB
[A17]	JavaBeans, EJB	J2EE, EJB
[A18]	not defined	PHP + Smarty
[A19]	not defined	ASP .NET
[A20]	not defined	MATLAB 6.5 and a discrete-event simulator
[A21]	not defined	not defined, (any language is possible to use)
[A22]	COM	HTML, ASP, ActiveX, COM
[A23]	JavaBeans	J2ME, JavaBeans
[A24]	Corba, (possible to use DCOM, EJB)	Corba
[A25]	CWBDM	not defined, but proposed architecture is based on Java
[A26]	not defined	not defined, (any language is possible to use)
[A27]	not defined	not defined, (any language is possible to use)

CBD BEST PRACTICES

Most papers do not explicitly report on any major problems while using the component approach, but rather they report suggestions for future researchers and practitioners concerned with CBD. All the suggestions are aggregated and presented in Table 8.

OVERVIEW OF FRAMEWORK USAGE

All the selected papers authors use some type of a web development framework, which can be divided in two groups; general and specific. General frameworks are used to develop any kind of web application, i.e. they can be used in many domains, while specific frameworks

Table 8. Suggestions for future component-based web framework development.

Paper	Suggestion
[A11]	Components should capture domain knowledge of web application development and hide complexities from End User
[A11]	Components should not capture application domain specific knowledge into Components. Rather those specific application needs should be abstracted and generic Components (Tools and Engines) should be created that can be used across many application domains."
[A11]	Components should be easy to use by End Users, yet they need to be complete so that it aid full capture of all the necessary 'Components parts' of the application such as front end pages, back end processing logic and database information
[A16]	It is not an easy task to develop an in-house component framework or to integrate available preexisting COTS in enterprise applications. It actually needed far more efforts and investments than it was foreseen in the beginning (approximately 50% more work than expected)."
[A16]	It is not one time effort but continuous process, which needs considerable investment in time and resources.
[A16]	The percentage of reusability changes from application to application and often needs component modification and reconfiguration."
[A16]	The major benefit of an in-house Component framework development surprisingly is not the project cost and time reduction based on business logic and business functions reusability (our but the company knowledge sharing and the creation of business function components
[A14]	Component composition: Each component is designed to achieve some special task; several components can be composed together in a dependent series to achieve a larger task.
[A14]	Problem in distributed systems is distributed component management.
[A18]	Problem is redesign of components to be more generic, simple and fast integration procedure with arbitrary Web applications.

are specialized (or limited) to only a certain type of web application, i.e. a certain domain. As it can be seen in Figure 10, authors tend to use general frameworks, however the number of specific ones is fairly significant. Table 9 and Table 10 present a detailed overview of frameworks which are in the selected papers.

In Figure 11 the time dispersion of framework types is given. As it can be seen, in last four years general frameworks are preferred. Although there is one exception, one can notice that there seems to be the stabilization of the research domain. Initially, there was a lot of specific frameworks but due to growing complexity of web applications, researchers seem to use existing and already proven frameworks.

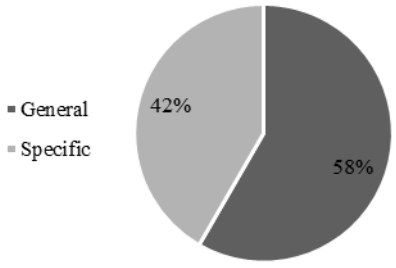
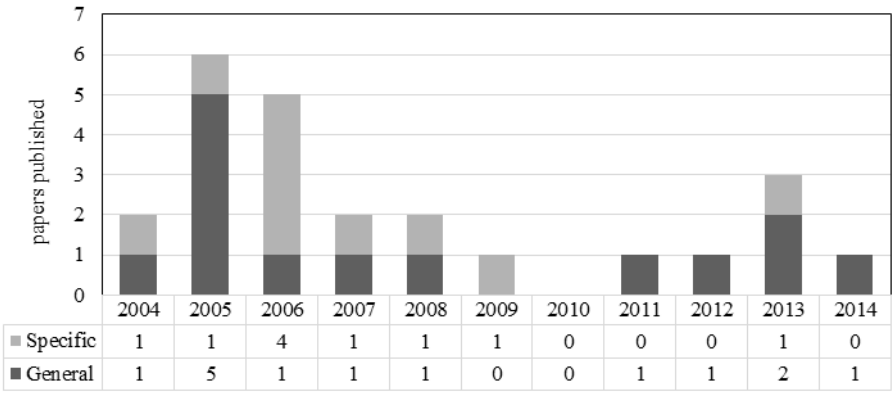


Figure 10. Framework types.



	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
Specific	1	1	4	1	1	1	0	0	0	1	0
General	1	5	1	1	1	0	0	1	1	2	1

Figure 11. Type of frameworks per year.

Table 9. Overview of general frameworks (continued on p.267).

Paper	Framework description
[A4]	New model openMVC is a framework which enables building web applications that then can change Style Information, Layout and validation constraints updated without coding. eCommerce shopping cart application was built as a prototype.
[A6]	Framework for developing and testing web applications. Model (CBTOWADM) that is used simplifies the difficulty of web application testing. Authors focus on functional testing based on a UML model. Model can also be used for developing web applications.
[A11]	Framework which end-users use to develop web applications using components developed by web developers. This framework targets SME's (small and medium enterprise) web applications
[A12]	Plug-in framework Plux, for integrating components into web applications.
[A14]	Web app framework for end-user development, for end-users to quickly implement simple sites with backend logic, like using a database. Framework is meant for SME web applications.
[A15]	XVM is XML virtual machine used as framework for developing and deploying XML-based applications, it is not a programing language. XML application container is built on top of XVM.
[A16]	Component framework in which different components are plugged (containing specific business logic) in and then are used to build Web applications. Currently Finance and Crediting web applications was built. Some components are domain specific like portfolio management.
[A17]	Real Time Distributed Control Systems (RTDCS) is a Framework for loose integration of COTS tools. Idea is integration of domain specific COTS tools, in the sense of automatic interchange of formally expressed information through standard and free software middleware. A prototype was built, which integrates several COTS tools aimed to develop RTDCS.
[A18]	Framework that can easily be configured to work and integrate into an arbitrary application, and by configuring the framework, we configure all the components created using it and make those accessible to the host application.
[A19]	WebComfort (Framework) a dynamic component-based CMS platform which allows users to manage and operate complex web applications in a dynamic and integrated fashion.
[A21]	Framework for developing component based open hypermedia systems (CB-OHS).
[A23]	Framework that makes the modeling, implementation, and maintenance of wireless mobile online applications intuitive and easy, especially for students and beginners. Framework decomposes a complex online application into modules. Each module is a plug-and-play unit. The components in the libraries can be directly called and used.

Table 9. Usage of CBD and connection to architecture (continuation from p.266).

Paper	Usage of Components	Connection to architecture
[A25]	Framework for developing web applications using COTS components.	
[A27]	Framework for developing any kind of web applications used in typical industrial enterprises. Framework gives a repository of components that can be used and customized to build new web applications, especially suitable for small and mid-sized enterprises with low IT expertise.	

Table 101. Overview specific frameworks.

Paper	Framework description	Domain, kind of applications
[A5]	Jade Framework for construction of self-repairable cluster systems. System analyses itself, and when failure is detected it start analyzing and repairs (reconfigures) from these failures based on system map.	distributed cluster systems of any kind
[A7]	WebMODE is component-based framework for developing e-learning web applications. WebMODE is semi-complete application that can produce specialized applications (eLearning).	eLearning applications
[A8]	Framework that uses adaptive hypermedia architecture AMACONT together with the component- oriented 3D document model CONTIGRA to achieve various types of 3D adaptation within Web pages.	3D Graphics Web applications (e-Commerce, e-learning, 3D collaboration, …)
[A9]	Enhance reusability and productivity for shot timeline projects in the field of bioinformatics to integrate all knowledge discovery. Paper gives architecture description for IKDD (interactive knowledge discovery and data mining) and demonstration through examples	biomedicine (bioinformatics)
[A10]	Developing component-based adaptive web presentation. Developing web presentations out of components.	adaptive web presentations
[A13]	Framework for building web based modular AIS systems.	Atlas Information Systems
[A20]	Systematic framework for designing and evaluating large-scale, component-based replication architectures for Web systems. Framework employs a combination of problem decomposition, configuration evaluation through controlled system simulations, and a neural-network-based feedback learning mechanism in the exploration of the design space.	decision support tool for a system designer to systematically explore design options and select an appropriate design configuration that best meets the design objectives
[A22]	Reusable WRMFDS model, aims at making such advanced information technologies be used widely in the condition monitoring and fault diagnosis domain, it can give developers a paradigm to accomplish the similar systems.	monitoring web applications
[A24]	Method to help design and implement a Web-based Decision Support Systems (DSS) in a distributed environment. Method was used on prototype Framework.	web based DSS (decision support systems)
[A26]	Method to build creative web search engines using the pataphysics.	Creative web search engines

Table 11. Usage of CBD and connection to architecture (continued on pp.269-271).

Paper	Usage of Components	Connection to architecture
[A10]	Components are documents, content-units or media types, which are used for construction of web presentations in component-based way.	Pipeline-based architecture is used which uses the components to generate final presentations. Everything is based on XML.
[A15]	"Components are responsible for realizing the application logic related to the associated element and providing services to other components." [A15]	XVM architecture is component-based architecture and also enables dynamic composition of components. XVM manager is responsible for handling all components. "In the XVM architecture, the key idea is a mapping between XML elements and software components, which associates XML elements with software components." [A15]
[A7]	Component are business rules, but because WebMODE is based on J2EE architecture, many other parts are also components.	WebMODE is based on J2EE architecture and uses the MVC design pattern. "J2EE web components are used to implement the presentation logic that connects the client interface to the business rules. "[A7]
[A11]	Two types of Components: " (a) Tools that allow End Users to create and assemble applications and (b) Engines that could be used to run these applications." [A11]	Architecture not particularly described. Framework is built so that end-users can use existing components to build new applications, or developers can create new component include into Framework and this can be further used again by end-users.
[A16]	Components contain business logic or presentation logic, with connectors to HTTP, JDBC, JNDI, CORBA, RMI	Architecture is multi-tier (client, web (presentation), business, and database). Components are used at web and business tier.
[A17]	Components are generated by different tools. These components are then used to build new applications. Enterprise Java Beans are a group of classes responsible for achieving the tasks implied in the business logic (the implementation of the services offered over RTDCS data).	The Model Collaboration Engine (MCE) architecture is based on View Model Approach. "Each of the Domain-Specific Models shows only the information about the system relevant to a specialist (or tool). Four models or views (more can be added) are identified as Domain-Specific: Control System (architecture independent system functionality), Distribution System (network topology and services), Real Time System (software architecture and temporal issues) and Software Engineering (code and documentation generation)." [A17]
[A21]	"Component/Structure server: Reifies the domain specific abstractions, providing the domain specific services to clients. They are semi-autonomous components, since they rely on the infrastructure services for common functionality. They establish a well-defined interface for communication with client applications." [A21]	Layered architecture is used in CB-OHS. These tiers are: client, structure server (components) and infrastructure.

Table 11. Usage of CBD and connection to architecture (continuation from p.268, continued on pp.270-271).

Paper	Usage of Components	Connection to architecture
[A8]	Component are used for document modeling.	AMACONT architecture with the component-oriented 3D document model (CONTIRGA). "AMACONT architecture processes component-oriented XML documents within the adaptation pipeline." [A8]
[A9]	Components are part of the Core architecture of IKDD Framework. Components are integrated in the third layer. "All components have to be sub-classes of the abstract class AComponent. … The AComponent class provides an intuitive and complete set of methods to program a new component." [A9]	Layered architecture is used (Database, Core, third layer, application layer). "The core package from iKDD models the component abstraction, implements a graph-based processing module, and covers basic components e.g. XML import / export." [A9]
[A13]	Components have specialized functionalities to client modules that require server-based functionality (e.g. data analysis or computation of visualizations that require large data set).	Client-server architecture is used. Components are used on server side.
[A22]	"Using component-based programming, we developed a highly maintainable system, which contains three components packages: Monitoring Controls Package based on ActiveX, Analysis Controls Package based on ActiveX, and Diagnosis Algorithms Package based on COM." [A22]	"A four-tier model based on the Microsoft's tier concept is adopted in the WRMFDS, which consists of the Presentation Services Tier, the Application Services Tier, the Data Access Services Tier and the Database Services Tier." [A22]
[A23]	Layered structure of components is used. The top layer consist of components which are used to build final applications (exp. welcome component, login components …).	Three-tired-architecture is used (MVC pattern). The tires are: server, client and databases. Components are used on server side. "A request-response pair contains three parts (Model, View, controller) and forms a unit. Each unit is implemented by reusing component libraries in the layered component structure and each unit can be plug-and-play into the system." [A23]
[A24]	"There are three major roles in the 3CoFramework: component implements or wraps the domain-specific computational logic or data access; a connector implements the component interaction; a coordinator implements the distributed components and connectors management." [A24]	Layered software architecture is used (Data, Information, Knowledge, Presentation) and components are used in each of these layers

Table 11. Usage of CBD and connection to architecture (continuation from pp.268-269, continued on p.271).

Paper	Usage of Components	Connection to architecture
[A19]	Components are different platform features.	In the paper they say that Component-based architecture is used. Architecture consist of: (1) Modules; (2) toolkits; (3) extenders; (4) data repository access; (5) module actions; and (6) the WebComfort API. As small connections between components as possible. Paramount to some of these aspects was the usage of the Provider pattern, which is a mix of the Abstract Factory, Strategy and Singleton patterns.
[A20]	Components are content created form three layers. "A Web application can usually be described in three layers. Presentation layer, business logic layer, and database layer. Each layer can be partitioned and distributed among the CDN's replica servers; In such replication approaches, *content elements drawn from the three layers are structured into components that are replicated*. The components are then dynamically assembled and delivered from the replica servers when they are requested."[A20]	Hierarchical component-based content architecture is used. Where components are at the lowest layer. Hierarchy, top down: application, site view- web page, components.
[A5]	Component is every tier in JEE multi-tiered architecture which is wrapped with JADE Framework.	"In a JEE multi-tiered architecture, the Web server is classically divided in several tiers: the HTTP daemon (Apache), the servlet engine (Tomcat), the EJB business server (JOnAS), and the database tier (MySQL for e.g.). Each tier is independently wrapped in a component." [A5]. Entire JADE framework is based on FRACTAL component-model. "Fractal is used in Jade in three main ways: (i) to construct the Jade software framework itself; (ii) to construct managed systems or to wrap legacy managed elements; (iii) to construct the System Map – a causally connected representation of the managed system model." [A5]
[A6]	Web application is divided into modules, module into sub module and sub module into components which realizes some function. "A component of CBTOWADM can be a dynamic link library, a web page, the logic function of a web page, a class, a web service, a database or a table of a database, an image file, and so on." [A6]	Component-based and tree-oriented web application development model (CBTOWADM) describes the software architecture.

Table 11. Usage of CBD and connection to architecture (continuation from pp.268-270).

Paper	Usage of Components	Connection to architecture
[A14]	Component implement backend functionalities, which end-users can use when building web applications (exp. search component for searching some data table). Two types of components: Components that implement functionalities that are not domain dependent and components that are domain-dependent.	Three-tired-architecture (client, application server, DB server) with MVC pattern is used. Components are used in application server.
[A12]	Component can be anything. Every user can add his own components. Components can be server-side that are installed and executed on server. Client-side installed and executed on client and use local resources. And sandbox components installed on server and downloaded to client on demand and executed on sandbox on the client.	Component based plug-in architecture. Different components are combined by end users and web applications created.
[A18]	Framework itself is a component that can be integrated into other web applications, but also consist of components which consist of components "nested components".	MVC architecture, components are used on all MVC layers.
[A4]	Components are used in all parts of the architecture for implementing all kind of functionalities, like styling component, validation constraint component, etc.	Five layers architecture (client, presentation logic, business logic, data abstraction, database), each layer has components for some specific functionalities.
[A25]	There are 4 kinds of components (domain, common business, base business) each with their own sub components which are used in new web applications build from end users.	Each component type is a specific module in the overall architecture. "We have identified web components and layered on ABCD architecture"[A25]
[A26]	Components are used to implement algorithm for creative web searching. "More advanced students could also develop their own components to test out theories and improve their understanding of the base concepts of not just search engines but the various fields that play a role in information retrieval systems." [A26]	"… component-based software architecture has been proposed which will allow for a range of different style systems to be developed with little overhead, thereby improving the chance of creative out-comes occurring in a different way." [A26]
[A27]	Components are used for building new web applications. They are used by end-user when they build their applications and users can customize the components with some parameters depending on their needs.	There are component-based application modules (parameter based, configuration based document based, description based), this modules is responsible for using of components based different types of customization. For example XML descriptors are used in the parameter based customization.

Table 11 shows for which purpose authors used component based development and how did it affect the software architecture of their web applications. There are three ways of component approach usage which one can distinguish:

- Components are used for creating web development frameworks – in this approach authors create component based frameworks which are used to create web applications, which can, but don't need be component based.
- Components used as application building blocks – in this approach components are used to create component oriented web applications without the underlying framework.
- Mixed approach – both framework and web application developed with this framework are component oriented.

In all the above cases the architectural decision is made solely by the end user, and all papers report only on developing prototypes (weather it is a framework or a web application). While most of the authors use component approach on the server side to implement various services, on the client side, 11 papers report using n-tier architecture, thus making it the most common.

SUGGESTIONS FOR FUTURE RESEARCH

If one is interested component approach and web development frameworks the most relevant scientific databases are Scopus, Springer and IEEE which will cover most of the related publications. Currently, the most relevant publications (2/3 of them being conference proceedings) were published between 2005 and 2006 which is most likely due the popularization of Web 2.0.

According to the selected papers here are some interesting research directions for the future:

- Graphical tools for creating application models which are then exported to XML schemas and automatically translated into component templates for creating web applications [A11].
- Enhancement of security, creating security models, and develop a complete XML virtual machine (XVM, XMLVM) development process model (analysis, deployment evaluation, performance evaluation, etc.) [A15], [A12].
- Refine components to reduce end-user effort to develop web applications, minimize faults, handle exceptions [A11].
- Research into component approach and mixed-media web applications [A8].
- Implementation of unified conceptual models and component libraries [A22].
- Model driven development approach for component web applications [A19].
- Research into verification models and tools for building component based web applications [A6].

It is apparent that component based approach is becoming a serious architectural direction and there is a very recent working groups focused solely on component based development for web, including the one from the W3C [9, 10].

CONCLUSION

In this paper we presented a literature review on component based development relation to web application frameworks. The original pool of related publications had 1132 papers which were, by the strict set of rules filtered out to 27 papers. Since three of them were inaccessible, 24 of them made it through to the full analysis which resulted in answers to the research questions $Q_{1\text{-}4}$.

Q_1: *In which way is CBD used for web application development?* - There are three main approaches: a) component approach is used for creating component based frameworks which are then used for creating web application (not necessarily component oriented), b)

component approach is used for building components which are the building blocks of web applications, and c) a mix of two previous approaches. In approach a) and c) the end user decides whether to use component approach for web application development while in b) component approach is imposed to the end users.

Q_2: *What is the relation between CBD and web application development?* – Component approach is used mostly for server side applications. Using it on the client side is less common, but there are cases and end-users aren't constrained to use it. Most widely used architecture is n-tired with components used inside different layers. For any future researchers and practitioners it is strongly suggested to plan component approach right from the start of the application design process. Although it requires more time, true benefits (separation of concerns, better maintainability, scalability, replaceability, single point of edit, etc.) are apparent later.

Q_3: *Which component models are used for web application development?* – Based on the reviewed papers it is obvious that EJB and Java beans are most preferable component models, hence Java being also the most popular programming language for this purpose. Although, it should be noted that there are a lot of custom models also. Since Java/J2EE is a leader in this field future researchers and practitioners have a choice to make, weather to expand the existing Java based component models or create new ones which requires more time, however offers new possibilities independent of a single technology.

Q_4: *In which web application development domains is CBD used?* – It is hard to recognize distinct domains however there are two types of web application development frameworks presented in the selected publications: a) general; used for any kind of web applications and b) specific; for developing special purpose web applications (e.g. eLearning, 3D graphics, monitoring, etc.). Majority of selected papers (14) describe the general framework.

Finally, the answer to the overall research question (i.e. how much of the current web application development frameworks explicitly refer to application of component-based approach) is hardly intuitive. It is apparent that most of the papers which are selected explicitly referee to component based approach, however this is the result of the selection process. While making decision, one should keep in mind the fact that the answers arise from a small number of processed papers (after filtering process). Nevertheless the answers are interesting and give an indication what is happening in the presented field. Review with more papers included should be performed to have a broader overview of the field.

There is a lot of publications dismissed which are related to SOA, and it would be very useful to perform an additional literature review with the same focus but focused on SOA. Considering the number of papers published, it is apparent that in the past there was a minor setback of this research area however, there is a growing trend. Although component based approach is more popular in other software engineering domains, with appearance of SOA, and Web 2.0 the number of component model is growing and we envision it will still grow, especially with W3C involvement. Therefore we reckon that any future web framework researchers and practitioners should be acquainted with component based development techniques, as it will become more popular in the near future.

APPENDIX A. EXTRACT OF SEARCH QUERIES

Here are the search queries for each database:

- SCOPUS (http://www.scopus.comhttp://www.scoupus.com/)
  - TITLE-ABS-KEY((web AND development) OR (web AND architecture)) AND TITLE-ABS-KEY("component based" OR "component-based" OR "CBD")
  - **691 findings without restrictions (Language, Year)**, 677 without "CBD" keyword

- ACM Digital Library (http://portal.acm.org)
  - **(((Title:web and Title:development) OR(Title:web and Title:architecture)) and (Title:"component based" OR Title:"component-based" OR Title: "CBD")) OR (((Abstract:web and Abstract:development) OR (Abstract:web and Abstract:architecture)) and (Abstract:"component based" OR Abstract:"component-based" OR Abstract:"CBD"))**
  - **296 findings without restrictions (Language, Year),** 291 without "CBD" keyword
- IEEE Xplore (http://ieeexplore.ieee.org)
  - **((web AND development) OR (web AND architecture)) AND ("component based" OR "component-based" OR "CBD")**
  - **326 findings without restrictions (Language, Year),** 325 without "CBD" keyword
- ScienceDirect (http://www.sciencedirect.com)
  - TITLE-ABSTR2-KEY((web AND development) OR (web AND architecture)) and TITLE-ABSTR-KEY("component based" OR "component-based" OR "CBD")
  - **34 findings without restrictions (Language, Year)**, 33 without "CBD" keyword
- ISI Web of Science (http://www.isiknowledge.com)
  - TS=(((web AND development) OR (web AND architecture)) AND ("component based" OR "component-based" OR "CBD")) OR TI=(((web AND development) OR (web AND architecture)) AND ("component based" OR "component-based" OR "CBD"))
  - **146 findings without restrictions (Language, Year)**, 141 without "CBD" keyword
- TOTAL (All databases) **1492 findings without restrictions (Language, Year)**, 1467 without "CBD" keyword

APPENDIX B. PAPERS OF THE STUDY

[A1] A. Jackson and J. G. Keating, "A non proprietary framework for policy controlled management of the model in the MVC design paradigm," in ICEIS 2005 - Proceedings of the 7th International Conference on Enterprise Information Systems, 2005, pp. 451–454.

[A2] H. Avoyan and A. Mehrabyan, "A circuit board approach to building web applications," in Proceedings of the IASTED International Conference on Internet and Multimedia Systems and Applications, 2006, vol. 2006, pp. 35–40.

[A3] X. Chen, X. Li, and Y. Zhan, "Component based webGIS development framework," in Advanced Materials Research, vol. 542–543, Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, Guangxi, 541004, China, 2012, pp. 1286–1289.

[A4] R. Barrett and S. J. Delany, "openMVC: A non-proprietary component-based framework for web applications," in Thirteenth International World Wide Web Conference Proceedings, WWW2004, 2004, pp. 1196–1197.

[A5] F. Boyer, N. De Palma, O. Gruber, S. Sicard, and J.-B. Stefani, "A self-repair architecture for cluster systems," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2009, vol. 5835 LNCS, pp. 124–147.

[A6] M. Cao, Z. Cao, and H.-Q. Li, "Support for development and test of web application: A tree-oriented model," J. Shanghai Univ., vol. 15, no. 5, pp. 357–362, 2011.

[A7] E. Q. Da Silva and D. De Abreu Moreira, "Developing customizable web-based educational applications through a component-based framework," in Proceedings - International Conference on Next Generation Web Services Practices, NWeSP 2005, 2005, vol. 2005, pp. 394–399.

[A8] R. Dachselt, M. Hinz, and S. Pietschmann, "Using the Amacont architecture for flexible adaptation of 3D web applications," in Web3D Symposium Proceedings, 2006, pp. 75–84.

[A9] J. Etienne, B. Wachmann, and L. Zhang, "A component-based framework for knowledge discovery in bioinformatics," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, vol. 2006, pp. 916–921.

[A10] Z. Fiala, M. Hinz, G.-J. Houben, and F. Frasincar, “Design and implementation of component-based adaptive web presentations,” in Proceedings of the ACM Symposium on Applied Computing, 2004, vol. 2, pp. 1698–1704.

[A11] J. A. Ginige, B. De Silva, and A. Ginige, “Towards end user development of Web applications for SMEs: A component based approach,” in Lecture Notes in Computer Science, 2005, vol. 3579, pp. 489–499.

[A12] M. Jahn, R. Wolfinger, M. Löberbauer, and H. Mössenböck, “Composing user-specific web applications from distributed plug-ins,” Comput. Sci. - Res. Dev., vol. 28, no. 1, pp. 85–105, 2013.

[A13] B. Jenny, A. Terribilini, H. Jenny, R. Gogu, L. Hurni, and V. Dietrich, “Modular web-based atlas information systems,” Cartographica, vol. 41, no. 3, pp. 247–256, 2006.

[A14] J. Li and T. Chusho, “A web application framework for end-user-initiative development with a visual tool,” in Lecture Notes in Engineering and Computer Science, 2012, vol. 1, pp. 816–822.

[A15] Q. Li, M. Y. Kim, E. So, and S. Wood, “XVM: A bridge between XML data and its behavior,” in Thirteenth International World Wide Web Conference Proceedings, WWW2004, 2004, pp. 155–163.

[A16] V. Lilov and S. Ilieva, “Towards development and use of in-house component framework: Results and expectations,” in Software Engineering and Advanced Applications, 2005. 31st EUROMICRO Conference, 2005, vol. 2005, pp. 12–17.

[A17] J. Portillo, O. Casquero, and M. Marcos, “Loose integration of COTS tools for the development of real time distributed control systems,” in Lecture Notes in Computer Science, 2005, vol. 3412, pp. 191–200.

[A18] S. Prstačić and M. Žagar, “Nested web application components framework: A comparison to competing software component models,” in ENASE 2013 - Proceedings of the 8th International Conference on Evaluation of Novel Approaches to Software Engineering, 2013, pp. 149–156.

[A19] J. de S. Saraiva and A. R. da Silva, “The WebComfort Framework: An Extensible Platform for the Development of Web Applications,” in 2008 34th Euromicro Conference Software Engineering and Advanced Applications, 2008, pp. 19–26.

[A20] R. Sharman and R. Ramesh, “Shared Content Management in Replicated Web Systems: A Design Framework Using Problem Decomposition, Controlled Simulation, and Feedback Learning,” IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev., vol. 38, no. 1, pp. 110–124, Jan. 2008.

[A21] M. Vaitis, M. Tzagarakis, G. Gkotsis, and P. Blachogeorgakopoulos, “Structural engineering: Processes and tools for developing component-based open hypermedia systems,” in Lecture Notes in Computer Science, 2005, vol. 3511, pp. 113–128.

[A22] X. Wu, J. Chen, R. Li, W. Sun, G. Zhang, and F. Li, “Modeling a web-based remote monitoring and fault diagnosis system with UML and component technology,” J. Intell. Inf. Syst., vol. 27, no. 1, pp. 5–19, 2006.

[A23] C. Xu, “A Framework for Developing Wireless Mobile Online Applications,” in 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering,Software Architecture and Reuse (ICIS-COMSAR’06), 2006, pp. 231–237.

[A24] S. Zhang and S. Goddard, “A software architecture and framework for Web-based distributed Decision Support Systems,” Decis. Support Syst., vol. 43, no. 4, pp. 1133–1150, 2007.

[A25] H.-K. Kim, H.-S. Yang, and R. Y. Lee, “A study on frameworks of component integration for web applications,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2005, vol. 3647 LNCS, pp. 58–70.

[A26] A. Hugill, H. Yang, F. Raczinski, and J. Sawle, “The pataphysics of creativity: Developing a tool for creative search,” Digit. Creat., vol. 24, no. 3, pp. 237–251, 2013.

[A27] X. Zhang, Y. Yang, S. Liu, and F. Liu, “Realization of a development platform for Web-based product customization systems,” Int. J. Comput. Integr. Manuf., vol. 20, no. 2–3, pp. 254–264, 2007.

[A28] M. Á. Moraga, C. Calero, J. Garzás, and M. Piattini, “Assessment of portlet quality: Collecting real experience,” Comput. Stand. Interfaces, vol. 31, no. 2, pp. 336–347, 2009.

[A29] I. Marmaridis and A. Ginge, "Paradigm shift in inter-organisational collaboration : A framework for web based dynamic ecollaboration," in ICSOFT 2007 - 2nd International Conference on Software and Data Technologies, Proceedings, 2007, vol. ISDM, no. WSEHS/-, pp. 178–185.

REFERENCES

[1] Breivold, H.P.; Crnkovic, I. and Larsson, M.: *A systematic review of software architecture evolution research*.
Information and Software Technology **54**(1), 16-40, 2012,
http://dx.doi.org/10.1016/j.infsof.2011.06.002,

[2] Kitchenham, B.: *Procedures for performing systematic reviews*.
Keele **33**, 1-26, 2004,

[3] Mendeley Ltd.: *Mendeley 2008*.
http://www.mendeley.com, accessed 1st February 2014,

[4] Bano, M. and Ikram, N.: *Issues and challenges of Requirement Engineering in Service Oriented Software Development*.
5th International Conference on Software Engineering Advances, 2010,
http://dx.doi.org/10.1109/icsea.2010.17,

[5] Alexander, A: *Scala Cookbook: Recipes for Object-Oriented and Functional Programming*.
O'Reilly Media, Sebastopol, 2013,

[6] Wang, A.J.A. and Qian, K.: *Component-Oriented Programming*.
John Wiley & Sons, Hoboken, 2005,
http://dx.doi.org/10.1002/0471713708,

[7] Dashorst, M. and Hillenius, E.: *Wicket in Action*.
Manning Publications Company, Greenwich, 2008,

[8] Brutzman, D. and Daly, L.: *X3D: Extensible 3D Graphics for Web Authors*.
Morgan Kaufmann Publishers, Monterey, 2007,

[9] W3C Working Group: *Introduction to Web Components*.
http://www.w3.org/TR/components-intro, accessed 15th February 2015,

[10] –: *WebComponents*.
http://webcomponents.org, accessed 15th February 2015.