Contents lists available at ScienceDirect



Information and Software Technology



journal homepage: www.elsevier.com/locate/infsof

Process models for service-based applications: A systematic literature review

Stephen Lane*, Ita Richardson

Lero, the Irish Software Engineering Research Centre, University of Limerick, Ireland

ARTICLE INFO

ABSTRACT

Article history: Received 8 June 2010 Received in revised form 6 December 2010 Accepted 12 December 2010 Available online 17 December 2010

Keywords: SOA Service-based application Software process Systematic literature review *Context:* Service-Oriented Computing (SOC) is a promising computing paradigm which facilitates the development of adaptive and loosely coupled service-based applications (SBAs). Many of the technical challenges pertaining to the development of SBAs have been addressed, however, there are still outstanding questions relating to the processes required to develop them.

Objective: The objective of this study is to systematically identify process models for developing service-based applications (SBAs) and review the processes within them. This will provide a useful starting point for any further research in the area. A secondary objective of the study is to identify process models which facilitate the adaptation of SBAs.

Method: In order to achieve this objective a systematic literature review (SLR) of the existing software engineering literature is conducted.

Results: During this research 722 studies were identified using a predefined search strategy, this number was narrowed down to 57 studies based on a set of strict inclusion and exclusion criteria. The results are reported both quantitatively in the form of a mapping study, as well as qualitatively in the form of a narrative summary of the key processes identified.

Conclusion: There are many process models reported for the development of SBAs varying in detail and maturity, this review has identified and categorised the processes within those process models. The review has also identified and evaluated process models which facilitate the adaptation of SBAs.

© 2010 Elsevier B.V. All rights reserved.

Contents

		~ =				
1.	Introduction					
2.	2. Background					
	2.1. Processes models for service-based applications	25				
	2.2. Taxonomy	25				
3.	Research method	26				
	3.1. Systematic literature review protocol 4	26				
	3.1.1. Research questions 4	26				
	3.1.2. Data sources	26				
	3.1.3. Search string	26				
	3.1.4. Study selection	27				
3.1.5. Data extraction						
	3.1.6. Data synthesis	28				
4.	Results	28				
	4.1. Summary of selected studies	29				
5.	RQ-1: Mapping study	30				
	5.1. Software process meta-model					
6.	Systematic review					

* Corresponding author.

E-mail addresses: stephen.lane@lero.ie (S. Lane), ita.richardson@lero.ie (I. Richardson).

	6.1.	RQ-1: I	Process categories	431
		6.1.1.	End to end models	431
		6.1.2.	Analysis and design	431
		6.1.3.	Construction	432
		6.1.4.	Deployment and Provisioning	433
		6.1.5.	Execution and Monitoring	433
		6.1.6.	Formal Methods	433
		6.1.7.	Adaption	434
	6.2.	RQ-2: /	Adaptation process models	434
		6.2.1.	ProDAOSS	434
		6.2.2.	PLASTIC	434
		6.2.3.	Muliview SOAD	435
		6.2.4.	BCDF	435
		6.2.5.	Chang et al.	435
		6.2.6.	СЅОМА	435
		6.2.7.	CSOA	435
		6.2.8.	Dino	435
7.	Discu	ssion		436
	7.1.	Literatu	ure Gaps	436
	7.2.	Limitat	tions of the Review	436
8.	Concl	lusions a	nd future work	436
	Ackn	owledgm	nents	436
	Appe	ndix A.	Selected studies	437
	Refer	ences		438

1. Introduction

Repeatable software development process are a key component in the development of high quality software [1]. Implementing a repeatable process ensures that all of the necessary development tasks get completed, and in the correct sequence. In order to make a process repeatable it needs to be documented so that project stakeholders can adhere to the process. In software engineering it is common to document a process by means of a process model. Software process models come in many forms from high level models such as the traditional waterfall model [2] to comprehensive improvement models such as CMMI [3] or the ISO-15504 [4]. Many domains such as the medical device or automotive industries require that software used in their products follow one of these comprehensive process models.

With the recent emergence of Service-Oriented Computing (SOC) many industries are interested in reaping its benefits. These benefits include the loose coupling of services which facilitates software reuse and the flexibility of service-based applications (SBAs) which allows them to be adapted with minimal effort. Unfortunately, because of the diversity of business requirements and operating contexts that accompanies the development of Service-Oriented Architectures (SOA), older software development paradigms cannot be blindly applied [5]. Many existing process models were designed with object-oriented or component based development in mind [6], therefore, since SOC requires a new development approach it will also require new supporting process models.

In this work we will have completed a systematic literature review in order to document the state of art in development processes models for developing service-based applications (SBAs). This will provide a useful starting point for further research in the area. Section 3 outlines the research method for this review, followed by Section 4 which outlines the results of the review. Section 5 contains a meta-model of the processes identified in the study, with Section 6.1 providing detailed descriptions of each of the process from the meta-model. Section 6.2 looks in detail at each study that facilitates adaptation in response to the second research question of the study. Section 7 contains a discussion of the results, followed by the conclusions in Section 8.

2. Background

2.1. Processes models for service-based applications

There have been many service-oriented process models proposed by proponents of SOC, for example IBM have proposed the SOMA [7] methodology while a group of industry practitioners have proposed the Service-Oriented Architecture (SOA) practitioners guide [8]. Similarly there have been efforts to tailor the RUP framework so that it can be used for SBA development. There are many SBA process models reported in grey literature such as white papers or technical reports. Lane et al. [9] report many of these, however, there may also be many other proprietary models maintained by organisations who keep them confidential for competitive advantage. Apart from proprietary models or the models reported in grey literature, there are many models published in scientific publications such as conference or workshop proceedings as well as peer reviewed journals. In order conduct an exhaustive review of relevant SBA process models using these sources, a systematic literature review (SLR) was conducted by the authors. A SLR is a comprehensive review carried out based on a set of predefined search criteria, which follows documented and repeatable guidelines. This is a popular review method in areas such as evidence based medical research and is being conducted increasingly in software engineering. The particular SLR procedure used for this review was proposed by Kitchenham and Charters [10] to meet the requirements of a software engineering review. In the next sections the review method and results of the review will be described in detail.

2.2. Taxonomy

Within the research areas of Software Process and Service-Oriented Computing (SOC) the terminology used is often ambiguous and non-standard. Having a consistent set of terms is important when evaluating or comparing different studies during a systematic review. In this section a taxonomy is presented where a brief description of the most common terms used throughout the review will be given with reference to synonyms found in the literature. The software process definitions used in the review are from Derniame et al.'s (1999) "Software process: principles, methodology, and technology" [11] which gives a comprehensive view of the software process research area. The SOC terms used in this review are defined the S-Cube project's knowledge model [12].

- Software process refers to the organisation, management, measurement and improvement of the activities involved in software development. Every software development project has a process whether or not it is explicitly or implicitly defined. At a bare minimum, software development will consist of a set of related activities whether or not they are measured or improved. Synonyms in the literature for software process include: software development process or software life-cycle [11].
- Software process models or process models are representations of real world software processes. Ideally process models should perfectly reflect the software processes they are trying to represent. Organisations often measure and improve their software process by documenting it and comparing it to reference process models such as CMMI or ISO-15504. Similarly, organisations may choose to implement an existing process model from the many types available [11].
- Service-Oriented Computing (SOC) refers to the computing paradigm that has evolved from objectoriented and component based design. The SOC paradigm facilitates ad-hoc, platform independent communication between unintroduced parties through the use of software services [12].
 - Services play a central role in SOC, they are self-contained loosely coupled computational elements that expose the functionality of underlying software. Services can be combined or composed into more complex services or they can be used in combination with other services to form service-based applications. Services can be published on public registries and made available for use by third party clients [12].

Service-based applications (SBAs) are applications that are composed of software services, the

composed of software services, the services may be controlled by the application developers or they may be third party services. A common example given for a SBA is the travel booking application: this is composed of a hotel booking service, a car rental service and an airline reservation service each of which are provided by third parties. Synonyms in the literature include service-oriented systems or service-based systems [12].

3. Research method

This review follows the guidelines set out in Kitchenham and Charters' [10] guide for SLRs in software engineering. This research method provides a verifiable method of summarizing existing approaches as well as identifying gaps in the current research.

3.1. Systematic literature review protocol

A review protocol is a written plan that is completed before the review begins. Therefore, the success of the review depends on the quality of the protocol. The protocol also provides a means by which the review itself can be repeated or updated at a later date to include subsequent publications. The protocol describes every aspect of the review from searching electronic databases to creating the final report and lays the groundwork for an unbiased systematic review Because the review is a snapshot in time it is impossible to replicate exactly. It would, however, be possible to update an existing review by including only new studies from a subsequent literature search.

3.1.1. Research questions

The review protocol begins with the construction of the research question or questions. Every systematic review has at least one primary research question with the possibility of one or more secondary questions. The research questions can be freely formed or they can contain a Population, Intervention, Comparison, and Outcomes as per Kitchenham and Charters' guide. The following are the primary and secondary research questions for this study:

- 1. What software processes models are proposed for developing SBAs ? (RQ-1)
 - Population: software
 - Intervention: process for SBAs
 - Comparison: process for traditional software
 - Outcomes: set of processes and techniques.
- 2. Do proposed SBA development processes models facilitate SBA adaptation ? (RQ-2).

3.1.2. Data sources

For this SLR, the electronic data sources in Table 1 were searched as these are the primary sources for software engineering research publications. Other software engineering SLRs such as [13],[14] use similar or the same sources. Other resources such as CiteSeer and Google Scholar are powerful search tools which return many of the same results as the databases mentioned. This duplication as well as the volume of non-relevant publications means that they were excluded as sources for this review.

3.1.3. Search string

The primary research question was be broken down to give a list of search terms to use in the electronic databases. Since the secondary question is based on the first, it is not necessary to have separate search terms. The search string will use the logical operator OR to include synonyms for each search term, and the logical

Table 1	
Literature	sources

Source	URL
IEEE Explore	http://leeexplore.ieee.org
ACM Digital Library	http://portal.acm.org
ISI Web of Knowledge	http://www.isiknowledge.com
SpringerLink	http://www.springerlink.com
ScienceDirect	http://www.sciencedirect.com
Wiley InterScience	http://www3.interscience.wiley.com

Table 2

Search	synonyms.
--------	-----------

Development techniqueSBADevelopment approachSOSEDevelopment processSOADevelopment life-cycleService-basedDevelopment life cycleService orientedDevelopment methodologyService orientedProcess modelService-centricProcess frameworkService computingDevelopment frameworkService engineeringsdlccmmcmmi15504	Software process	Service-based
Development approachSOSEDevelopment processSOADevelopment life-cycleService-basedDevelopment life cycleService orientedDevelopment methodologyService orientedProcess modelService-centricProcess frameworkService computingDevelopment frameworkService engineeringsdlccmmcmmi15504	Development technique	SBA
Development processSOADevelopment life-cycleService basedDevelopment life cycleService-orientedDevelopment methodologyService orientedProcess modelService-centricProcess frameworkService computingDevelopment modelService computingDevelopment frameworkService engineeringsdlccmmcmmi15504	Development approach	SOSE
Development life-cycleService basedDevelopment life cycleService-orientedDevelopment methodologyService orientedProcess modelService-centricProcess frameworkService centricDevelopment modelService computingDevelopment frameworkService engineeringsdlccmmcmmi15504	Development process	SOA
Development life cycleService-orientedDevelopment methodologyService orientedProcess modelService-centricProcess frameworkService centricDevelopment modelService computingDevelopment frameworkService engineeringsdlccmmcmmi15504	Development life-cycle	Service based
Development methodologyService orientedProcess modelService-centricProcess frameworkService centricDevelopment modelService computingDevelopment frameworkService engineeringsdlccmmcmmi15504	Development life cycle	Service-oriented
Process modelService-centricProcess frameworkService centricDevelopment modelService computingDevelopment frameworkService engineeringsdlccmmcmmi15504	Development methodology	Service oriented
Process frameworkService centricDevelopment modelService computingDevelopment frameworkService engineeringsdlccmmcmmi15504	Process model	Service-centric
Development modelService computingDevelopment frameworkService engineeringsdlccmmcmmi15504	Process framework	Service centric
Development framework Service engineering sdlc cmm cmmi 15504	Development model	Service computing
sdlc cmm cmmi 15504	Development framework	Service engineering
cmm cmmi 15504	sdlc	
cmmi 15504	cmm	
15504	cmmi	
	15504	
spice	spice	
12207	12207	
15288	15288	

Table 3

Inclusion/exclusion criteria sources.

Included

- Peer reviewed research papers from electronic databases.
- Papers that describe more than a single specialised process.
- Must describe processes for developing SBAs.

Excluded

- Non-English papers.
- Papers specific to a particular technology.
- Papers that only address once specific process area.
- Papers that are application domain specific.
- Papers which are obviously not related to the research questions in this protocol.

- Letters and editorials.

- Duplicate publications on the same approach.

operator AND to link together each set of synonyms. Table 2 lists all of the search synonyms used in the construction of the search string. When concatenated using the appropriate boolean expressions the following generic search string was produced:

("software process" or "software engineering process" or "engineering process" or "engineering methodology" or "sose framework" or soad or soma or "architecture framework" or "architecture frameworks" or "engineering methodology" or "development technique" or "development approach" or "development methodology" or "development process" or "development processes" or "development lifecycle" or "development life cycle" or "process model" or "process framework" or "development model" or "development framework" or sdlc or cmm or cmmi or 15504 or spice or 12207 or 15288) and (sba or sose or soa or "service-based" or "service based" or "service centric" or "service computing" or "service engineering").

Some of the databases such as IEEEXplore, can take this string directly while others, such as Web of Science, required slight modification. SpringerLink has a 10 term length on search strings which meant that a series of strings had to be used instead of one generic string. In all cases the string was applied to the abstracts contained in the databases. Applying the string to titles produced too few results while applying the string to full texts produced thousands of inaccurate results.

3.1.4. Study selection

The studies that were selected for inclusion in this study were identified from on-line electronic databases during October 2009. In order to determine whether or not a study is included, the abstract was evaluated based on the inclusion/exclusion criteria in Table 3. The inclusion and exclusion criteria we selected in order to exclude irrelevant publications while keeping studies of interest. Most of the criteria are self explanatory, however, some of the criteria need a little explanation. The criterion: "Papers that describe more than one process" was included because hundreds of the the results returned in the initial search detailed single low level processes that were too specialised to be applied generally by practitioners. We were looking for studies that were aimed at guiding the high level development process such as life-cycle models or process models with several high-level processes. The criterion: "Must describe service-oriented process activities/practices" was included to filter out studies that do not contain processes/activities that can be used to develop SBAs.

In many cases it was not possible to determine whether or not a study should be included due to the quality of its abstract. In these cases the studies were either included or excluded based on their entire full text. After the initial set of studies were selected their details were recorder using the citation management tool Zotero.¹ Full texts were obtained where possible for the data extraction part of the SLR. It is also important to note that many papers were excluded based on their full text as it became apparent that they failed to meet the inclusion/exclusion criteria during the data extraction process.

The study selection process is one of the most critical processes of the review process and also one of the most time consuming. It is therefore vital that the search string only returns studies that are closely related to the research questions to prevent unnecessary study selection effort. It is also vital not to exclude important relevant studies by using search strings with an overly narrow focus. To strike a balance between too many and too few results of the selection process, the search string was validated using a selection of 5 known relevant publications. The search string was tested against the publisher's websites for those known publications to ensure that the studies were returned in the publisher's search engine results.

The selection process was undertaken by the first author, with a random selection of 20 of the original results audited by the second author in order to ensure minimal selection bias. There was a close correlation between the studies chosen for selection by the two authors which indicates minimal selection bias.

3.1.5. Data extraction

After the full text of the studies were retrieved the data extraction phase begun. For this activity a custom data extraction form was created using MS Access™. This allowed for convenient data entry as a paper's full text was being reviewed. Another benefit of using this approach was the ease with which reports could be generated from the underlying data table. Meta-data such as author, title and publication source was collected with descriptive data fields such as study type and focus during this activity. Table 4 shows the data fields that were extracted from each study along with their descriptions and their associated research questions. The first research question, RQ1, which refers to the identification of process models for developing SBAs requires all of the data extraction fields for analysis. The data fields are used to illustrate frequency distributions as well as to map the various processes for developing SBAs. Some of the more content-oriented fields such as "Processes" and "Study Focus" are used to determine whether the identified processes facilitate SBA adaptation, in answer to the second research question, RQ2.

Like the study selection process, the data extraction process was conducted by the primary author and audited by the secondary

¹ http://www.zotero.org/.

Tab	ole	4

Data extraction fields.

Data field	Purpose	Research questions
Reviewer	Name of reviewer	RQ1
Title	Name of study	RQ1
Authors	Study authors	RQ1
Publication source	Where the paper is	RQ1
	published, name of journal, conference, etc.	
Publication type	Is the paper a conference	RQ1
	paper, journal paper, etc.	
Year of publication	When study was published	RQ1
Primary/secondary	Deos the study use primary	RQ1
	or secondary data	
Study type	Type of research methods used	RQ1
Study context	Industry, lab based study,	RQ1
Ctudu nonulation	etc.	DO1
Study population	students, academics	RQI
	industry exports atc	
Research questions	Research question(s) of the	RO1 RO2
Research questions	study	KQ1, KQ2
Study focus	Primary objective of the	RQ1, RQ2
	study	
Processes	Process described in the	RQ1, RQ2
Findings/conclusions	Main conclusions from the	RO1 RO2
	study	
IS valid	Was it a valid study	RQ1, RQ2
	•	

author. A random selection of 10 of the selected studies were reviewed by the second author and the appropriate data fields extracted. The fields that were extracted by both authors were compared and found to be sufficiently similar not to warrant further validation.

3.1.6. Data synthesis

The data synthesis phase of this review aims at addressing the two research questions posed at the beginning of the review. The first research question was to identify process models for developing SBAs. In order to address this research question, details of the process models contained in the selected studies were extracted. In order to get an idea of the scope of the process models identified, the individual processes identified within each process model were synthesised into a meta-model. This meta-model provides an overview of the area and will also highlights gaps in the research. Once the meta-model is completed a narrative summary of each of the process areas is provided based on the studies that make up each category. The second research question asks whether or not the development processes that are identified address adaptation of SBAs. In order to answer this question studies which contain SBA adaptation related processes are also critically reviewed through a narrative summary.

A meta-model is similar to a map from a mapping study as outlined by [15,16]. Rather than looking at each study in detail, the aim is to scope out the breath of the research area. The narrative summary of each process within the meta-model is more typical of SLRs than mapping studies. For this reason the data synthesis process is divided into two parts, a mapping study to scope the area and an SLR to evaluate the included studies. Petersen et al. [15] recommend that rather than use these synthesis approaches in isolation, they can be used to complement one another. For example, in this case, the systematic map is first created, then the specific processes are critically evaluated through the narrative summary. A noteworthy exclusion from this study is a quality assessment, as these are often present in systematic reviews. However, in this study there are many qualitative studies included which makes it difficult to apply a non-subjective quality score.

For this study, a UML² (Unified Modeling Language) meta-modeling approach was taken as an alternative to the bubble plot approach often used in mapping studies. The processes used in this meta-model were constructed using Noblit and Hare's reciprocal translation technique [17]. This is the synthesis technique suggested by Kitchenham [10] when researchers are attempting to create an additive summary of the literature. Reciprocal translation summarises concepts by translating similar concepts into a single concept. This processes is continued until it is not possible to translate any more concepts into one another. For example, if the following terms are encountered: "service-based application adaptation", "serviceoriented system adaptation" or "dynamic modification of serviceoriented systems" they may all be translated into "service-based application adaptation". This technique then produces a normalised set of development processes covering the entire research area assuming that the included studies are representative. Other statistical data such as frequency distributions were reported in tables which illustrate how the research area changes over time.

In our meta-model we have made use of the following UML constructs some of which, for simplicity, have been tailored slightly from their official definitions:

- Class: A class (square box) in our mata-model is used to represent concepts or processes.
- Association: An association in our model is used to represent a relation between two concepts (processes, sub-processes). In our model we have used standard associations (arrow without arrowhead) and directional associations (arrow with open arrowhead). Directional associations imply precedence and sequence between classes (processes).
- Aggregation: An aggregation (arrow with diamond head) is used in the context of this model to represent "has a" relationships.
- *Generalization*: A generalization (arrow with closed arrowhead) relation in our model represents a relationship between two classes (processes) where one class is a sub-type of the other.

The narrative summary synthesis method is used twice during this study, firstly to summarise the key processes from the metamodel and secondly to summarise and critically discuss each of the studies that contain processes for adapting SBAs. Since there are to many studies to discuss individually, it seems appropriate to discuss the process themes emerging from the studies. SBA development processes that also facilitate adaptation are expected to be a much smaller subset of the total number of studies which permits a narrative summary at an individual study basis. Therefore the meta-model and process summaries address RQ-1, while the summaries of each study with adaptation processes address RQ-2.

4. Results

After an exhaustive literature search, our data contained 722 studies. The two largest set of results came from IEEE Xplore (185) and the Web of Science (333). Studies were then excluded if they focused on a single process or technique or were specific to a particular application domain. If the exclusion criteria were relaxed there would be many more interesting studies. However they are beyond the scope of this review. After the abstracts and titles of each study were read the number of studies was reduced to 77. However, on fully reading each of these papers a further 20 were excluded leaving a total of 57 valid studies. The categorical data that was extracted from the selected studies is shown in Table 5.

² http://www.uml.org.

Table 5	
---------	--

Categorical data extracted from selected studies.

	Study	Source	Lifecycle processes	Sub processes	End to end model	Supports adaptation
	[S1]	Conference paper	RED, CON	MDD	No	Yes
	[S2]	Conference paper	RED, CON	MDD	No	No
	[\$3]	Journal paper	RED, CON, DP, EM	Discovery, composition	Yes	No
	[S4]	Conference paper	RED, CON, DP	MDD	Yes	No
	[\$5]	Workshop paper	RED, CON	Agent-oriented development	No	No
	[S6]	Conference paper	RED, CON	MDD	No	Yes
	[S7]	Conference paper	RED, CON	MDD	No	No
	[S8]	Workshop paper	RED, CON	Service composition	Yes	No
	[S9]	Conference paper	RED, CON	MDD, security	No	No
	[S10]	Conference paper	RED, CON, DP	MDD	Yes	No
	[S11]	Conference paper	RED	Agent-oriented development	No	Yes
	[S12]	Journal paper	RED, CON, DP	Service composition	No	No
	[S13]	Conference paper	RED, CON	Service composition	Yes	Yes
	[S14]	Conference paper	RED, CON	Service specification	No	Yes
	[S15]	Conference paper	RED, CON, DP	Service specification	No	No
	[S16]	Conference paper	RED	Service identification, service specification, service realisation, service composition	Yes	Yes
	[S17]	Conference paper	RED	Service identification, service specification, service realisation	Yes	No
	[\$18]	Conference naper	RED CON DP	realisation, service composition	Ves	No
	[510]	Conference paper	RED, CON, DI, RED, CON, DP, EM	Service discovery service composition	No	No
	[515]	Conference paper	RED, CON, DP, EM	service discovery, service composition	No	No
	[520]	Workshon naner	RED, CON	МПЛ	No	No
	[527]	Conference naper	RED, CON	Collaborative modelling	Ves	No
	[522]	Thesis	RED, CON	MDD	No	No
	[523]	Journal namer	RED, CON DP		No	No
	[524]	Workshon naner	RED, CON, DI	Security	No	No
	[526]	Workshop paper	RED CON	Formal methods	No	No
	[\$27]	Conference naper	RED, CON	MDD for SOMA	No	No
	[\$28]	Book section	RED, CON	Service composition	No	No
	[\$29]	Journal naper	RED, CON	Formal methods	No	No
	[\$20]	Conference naper	RED CON	Context aware MDD	No	Ves
	[\$31]	lournal naper	RED, CON	MDD	No	No
	[\$32]	Conference naper	RED, CON	MDD	No	Ves
	[\$33]	Conference paper	RED	Formal methods	No	No
	[\$34]	Conference paper	RED CON	MDD	No	No
	[\$35]	Workshop paper	RED CON	MDD	No	No
	[\$36]	Conference naper	RED CON	Formal methods	No	No
	[\$37]	lournal naper	RED CON	Semantic modelling	No	No
	[S38]	Symposium	RED, CON, DP, EM	MDD, service specification, service realisation,	Yes	Yes
	[023]	Conforma papar	RED CON		No	No
	[539]		RED, CON	MDD	No	NO
	[540]	Conference paper	RED, CON DD FM		NO	NO
	[541]	Journal paper	RED, CON, DP, EIVI		No	No
	[542]	Conference paper	RED, CON	Somica identification convice enacification convice	No	NO
	[345]		KED, CON	realisation	NO	INO
	[S44]	Conference paper	RED, CON	Service identification	No	No
	[\$45]	Conference paper	RED, CON		No	No
	[S46]	Workshop paper	RED, CON	MDD	No	No
	[\$47]	Journal paper	RED, CON, DP	MDD, service identification, service specification, service realisation	Yes	No
	[S48]	Conference paper	RED, CON	Aspect-oriented development	No	No
	[S49]	Workshop paper	RED, CON, DP		Yes	No
	[S50]	Conference paper	RED, CON	Aspect-oriented development	No	No
	[S51]	Conference paper	RED, CON, DP	MDD	Yes	No
	[S52]	Conference paper	RED, CON, DP	Web based MDD	Yes	No
	[S53]	Conference paper	RED, CON, DP	MDD	Yes	No
	[S54]	Journal paper	RED, CON, DP		Yes	No
	[S55]	Journal paper	RED, CON	MDD	No	No
	[\$56]	Conference paper	KED, CON, DP	Service discovery	Yes	No
	[\$57]	Conference paper	RED, CON, DP, EM		Yes	No
_						

RED – requirement engineering and design.

CON – construction.

DP - deployment and provisioning.

EM - execution and management.

MDD – model driven development.

4.1. Summary of selected studies

From the 57 papers selected there was a wide variation in the approaches described, the quality of the research and the maturity of the studies. There were 45 secondary research and 12 primary

research studies which suggest that the research area is lacking in primary studies. Analysis shows that there were 40 academic studies compared to 17 industry based studies. Although secondary or academic studies may be as valid as their counterparts this trend suggests that the area is lacking mature industry-based

Table 6

Types of publication by year.

Publication type	Year						Total
	04	05	06	07	08	09	
Book section					1		1
Conference paper	3	3	3	9	14	4	36
Journal paper	1	1	1	2	2	3	10
Symposium					1		1
Thesis				1			1
Workshop paper	1		3	3		1	8
Total	5	4	7	15	18	8	57



Fig. 1. Publication trend.





Fig. 2. Gartner Hype Cycle.

research. A good indication of the maturity of an area is the type of publications in that area. Journal articles are often more mature than conference or workshop papers. Table 6 shows the number of each type of publication by year. Conference papers, workshop papers and journal articles make up the majority of publications. There are only 10 journal articles with a total of 44 workshop and conference papers.

As illustrated in Fig. 1, the first service-oriented development approaches appeared in 2004, with the number of publications rising rapidly from 2006 to 2008. Then interestingly the number of publications drop off again in 2009. One interpretation for the 2009 drop is that some publications were not yet indexed by the electronic databases when the searches were conducted in October 2009. Another interpretation for the drop in 2009 may be that the

able	1
tudy	tuno

т

Study type.	
Literature based	25
Discussion	12
Example application	9
Case study	8
Exploratory case study	2
Experience report	1

publication trend is following the Gartner Hype-Cycle³ in Fig. 2, in this case the trend would suggest that the research area has already reached its peak of inflated expectations with 2009 representing its trough of disillusionment. This corroborates the Hype-Cycle of Service-Oriented Software Engineering (SOSE) challenges reported in [13]. Its is advisable not to draw draw conclusions from the trend without knowing the true number of publications indexed in 2009.

Another interesting way to view the selected studies is by study type, this may also give us a view into the quality and maturity of the selected studies. As can be seen in Table 7, the most popular study type was literature based studies. In this context, "Literature based" means that existing approaches from the literature were used or derived from to create service-oriented development approaches. The next most popular study type were discussion based papers. The discussion based papers that were reviewed were mostly short papers based on the authors' experiences rather than being based on primary or secondary research. The remaining studies were either example applications or case studies. Example applications were used to illustrate tool supported approaches such as Model Drivel Architectures (MDA).

5. RQ-1: Mapping study

In this section a quantitative mapping analysis of the selected studies is provided, each study is categorised according to the processes they contain and mapped to a process meta-model. The publication and study type distributions are shown in tables, while the process category relationships are mapped using a subset of UML (Unified Modeling Language). This approach was chosen because it is an intuitive representation format with the ability to represent associations between processes.

5.1. Software process meta-model

The meta-model shown in Fig. 3 displays the processes identified in this study along with their inter-relations.

The meta-model construction was a multi-stage process that was completed after the data collection activity of the SLR. The first step of constructing the meta-model was to classify the processes extracted from the literature into groups of related processes using reciprocal translation. One of the decisions during reciprocal translation is which process description to use in order to describe a group of processes. In our earlier example we encountered the following processes: "service-based application adaptation", "service-oriented system adaptation" or "dynamic modification of service-oriented systems". Any of these individual terms can be used to describe the process in our meta-model. In order to keep the resultant processes consistent with one another the processes from Papazoglou and Van Den Heuvel's [5] Service Development Life-Cycle (SLDC) approach were used as a base point for the translation process. The four life-cycle processes from the SLDC which were used as the base point are highlighted in Fig. 3. This approach was chosen as a base point because it is a detailed approach that has high-level processes for the entire development life-cycle.

³ http://en.wikipedia.org/wiki/Hype_cycle.



Fig. 3. Service development process meta-model.

There are several other high-level life-cycle models that could have been chosen for the base point, however, the same meta-model should result regardless of which one is chosen if the reciprocal translation process is followed accurately. Processes encountered during the review were evaluated as to whether or not they could be translated into one of the base point processes. If they could be translated they were eliminated. If not, they were added to the meta-model as a separate process or sub process.

The next step was to construct these related processes into a meta-model where associations between processes were represented. The completed meta-model gives a comprehensive view of the state of the art in software processes for SBAs. In the following sections an overview will be given of each of the prominent software processes proposed in the studies selected for this review.

6. Systematic review

6.1. RQ-1: Process categories

6.1.1. End to end models

Many of the studies encountered in this review propose complete SBA development methodologies with processes for designing, implementing and operating them. These methodologies often borrow concepts from existing software development techniques or methodologies, for example, Kruger and Meisinger [S15] propose an extension of the V-Model for use with SBAs. Similarly Christou et al. [S24] propose the use of Agile methods along with the Rational Unified Process (RUP) development methodology for the development of SBAs. The use of Software Product Lines (SPL) development techniques are also suggested in combination with Agile methods by Karama *at al* [S12]. Deubler et al. [S51] have developed a tool to support the development of SBAs, the tool's primary functions are to formally verify service-based systems as well as the generation of system code.

Many of the end to end process models in the literature follow the classical waterfall development model with some modifications that are particular to SOC. Adamopoulos [S18] proposes an iterative development methodology which includes many of the waterfall model's processes, however they are adapted to suit Web Service Based Applications (WSBAs). Papazoglou and van den Heuvel [S41] suggest a process model that also bears a close resemblance to the waterfall model. However, it has processes such as "Execution and Monitoring" that are specific to SBAs. Processes in this model such as "Analysis and Design" bear resemblance to traditional processes not intended for SBA development. However, they are tailored specifically to meet the requirements of such applications.

Along with the fully defined process models, there are many "work in progress" models in the literature. Many of the "work in progress" models attempt to discover all of the service related concepts and aspects that make development of SBAs unique from the development of traditional software applications. One such work by Engels and Assmann [S42] set out the several development challenges of service-oriented enterprise architectures. Ivanyukovich et al. [S54] are also working towards a service-oriented process model They propose a model with the following three dimentions: managing the software change, specifying the development process and targeting stakeholders goals.

The final type of end to end process model encountered in this review are interdisciplinary development models. These development models involve expertise from several domains. For example, Lamparter and Sure [S19], propose a development methodology that includes aspects of web service engineering, market engineering and ontology engineering.

6.1.2. Analysis and design

There are many analysis and design process models reported for the development of services and SBAs. Most approaches fall in to the categories of top down, meet in the middle or ground up service realisation [S41]. The top down design approach involves working backwards from existing service interfaces. The components required to implement the service are constructed based on the functionality described by these interfaces. This design technique is considered best practice as business requirements shape the service interfaces which in turn drive the development process. The bottom up approach differs in that the functionality of existing applications are made available by wrapping existing system functions with services. Finally, the meet in the middle approach combines features of both top down and bottom approaches. This technique is useful when a service needs to be created to implement an existing service interface. However, rather than create a new software component to implement this service, which would be done with top down design, an existing software component that has all of the required functionality would be wrapped with the required service.

The top down approach is more business oriented than its counterparts because services are created to meet specific business requirements. The same can be said for the meet in the middle approach when software components exist to suit service requirements. The ground up service realisation approach is least business oriented in that existing software components are wrapped with services which limits the capability of resultant applications.

After reviewing the development models that address the analvsis and design processes, it appears that many of these models inherit characteristics from the popular Service-Oriented Modelling and Architecture (SOMA) methodology [S47,S43] developed by Arsanjani et al. at IBM. The SOMA methodology is one of the most mature design process models encountered in the review. Its developers claim that methodology was developed during involvement with hundreds of SOA projects. SOMA's design activities are divided into three processes: "Service Identification", "Service Specification" and "Service Realisation". These processes appear in many other models in some form or another. For example, Engels et al. [S4] propose a method for engineering SOAs which contain the design processes of: "Identify top-level business services", "Refine business services" and "Specify business services". The SOMA processes: Service Identification, Service Specification and Service Realisation and the other analysis and design processes encountered will be discussed in more detail in the following subsections

Service identification is a process that identifies the candidate services required for developing a SBA. It involves the identification of the required services as well as the software that will realise those services. SOMA identified three techniques for the identification of services: goal-service modeling (GSM), domain decomposition, and existing asset analysis. Kenzi *at al* [S6] propose a Model Driven Architecture (MDA) framework for the development of adaptable SBAs that uses service identification models for the identification of candidate services.

Service specification involves the development of service interfaces and the management of service dependencies, once the required services have been identified. Most service development methodologies either implicitly or explicitly contain a service specification process specifying services is one of the fundamental building blocks of SBAs.

Service realisation, like service specification, is a vital step in the development of SBAs. Realisation is the final process before the implementation of services. It involves the implementation of the service specification into usable design documents that can be used by the service implementers. Service design documents need to include artifacts such as component implementation details and SOA solution stack details. An extremely common approach being proposed for the realisation of services is the application of Model Driven Architecture (MDA). This is a model driven design approach with the added benefit of automatic code generation from design models. MDA will be discussed in more detail later in this review.

Another description given to the analysis and design processes for SBAs is service-oriented analysis and design (SOAD). This is an extension of the Object-Oriented Development (OOD) and Component-Based Development (CBD) paradigms. SOAD is often used as a general term in the literature to describe analysis design approaches for service oriented systems. Different SOAD methodologies take different approaches to service design. For example, Chang and Kim [S17] propose a 5 phase design process within the following phases: "Identifying Business Processes", "Defining Unit Services", "Discovering Services", "Developing Services", and "Composing Services". Comparatively Kambhampaty [S40] proposes a SOAD methodology with four processes which are quite different from Chang and Kim's methodology: "Activity Services Development", "Business Process Services Development", "Client Services Development" and "Data Services Development".

Service discovery for static SBAs occurs during application design, while the discovery of services for dynamically adapting SBAs may occur during system runtime. There are many ways that service discovery can be achieved, from manually searching service directories to automatically scanning directories for candidate services that match a given criteria. Howard and Kerschberg [S3] present a framework to support the preparation, publication, requisition, discovery, selection, configuration, deployment, and delivery of semantic web services. Their framework specifies that both published services and service clients contain semantic descriptions of their capabilities and various types of requirements.

Service composition is a service design/construction process that involves the combination of services into new complex services or SBAs. Service composition spans several processes of the software development life-cycle – how services are composed needs to be established during the design of SBAs, the composition gets enacted during the development of a SBA and finally service compositions need to be monitored and possibly adapted during the operation of SBAs. Ren et al. [S8] propose a framework that facilitates the visual design, validation and development of web service compositions.

Agent-orientation describes a computer architecture that is made up of a system of intelligent agents. The agents have characteristics such as autonomy, social ability, reactivity, and pro-activeness [18]. Lu and Chhabra [S5] propose a service-oriented design methodology that involved the wrapping of intelligent agents with web services which speeds up the development process and results in more competent web services.

6.1.3. Construction

Many of the process models encountered in the literature focus on the construction of services [S52], SBAs [S10] or both [S7]. The construction of services is a role undertaken by service providers whereas the construction of SBAs is a role undertaken by service consumers. Service providers and service consumers have differing roles but may be in the same organisation. The focus of this review is the construction of SBAs rather than services alone, however in the absence of suitable existing services, the construction of services is a sub-process of the SBA construction process.

The SBA development processes proposed vary in their approach. A common proposal is to adapt an existing traditional engineering approach or framework to suit the service-oriented paradigm. Meisinger and Kruger [S15] suggest an adapted version of the traditional V-Model for the development of service-oriented systems. Similarly there have been many Model Driven Development (MDD) techniques suggested for the construction of SBAs. MDD is a popular approach for the design and implementation of object-oriented software. During MDD analysis and design models can be used to automatically generate system code automatically constructing the system. MDD can greatly simplify the construction process of SBAs by abstracting complex implementation standards and service interface details from developers. This process will be discussed in more detail in the next section.

Model Driven Development (MDD) is a development paradigm that uses models to represent and reason about problem and solution domains [S7]. Modeling is a useful method of working towards a solution which enables the representation of dependencies between the components being modeled. Once a model has been created it is possible to view the effects of changing part of the model by examining dependencies within the models. A model can be used as a reference in order to construct a software system, alternatively using the appropriate technology a well defined model can be used to automatically generate a platform specific model and then application code which can form the skeleton of a system. This use of modeling to generate system code is referred to as Model-Driven Architecture (MDA).

Models are an intuitive method of developing solutions while at the same time they abstract the specific details of implementing the solution. This proposition is particularly appealing to SOC where there are many protocols and standards involved in implementing SBAs. In order to implement a basic WSBA the solution may involve the SOAP protocol for messaging, WSDL for describing service interfaces, and BEPL for implementing service compositions. Each of these are verbose XML based languages which are difficult to interpret by humans and are error prone when constructed manually. For this reason there have been many service development processes proposed that are based on MDD. In this systematic review 17 different MDD approaches were encountered.

A common approach proposed is to develop system models with UML [S35] diagrams. This is a logical choice as there are many MDA tools in existence that can handle UML. Johnson and Brown [S7] outline an MDD technique for developing SBAs using a UML profile suitable for service modelling. Many of the MDD approaches proposed conform to MDA principles where there are several models created during development and these different model types can be transformed into one another and finally into source code. The most common model types proposed in the literature were Computation Independent Models (CIM), Platform Independent Models (PIM) and Platform Specific Models (PSM) [S32,S37]. Each of these models are one order of abstraction higher than their successor, for example, a CIM models domain specific artifacts which would use vocabulary familiar to domain practitioners, while PIM models refer to the structure of a system without referring to platform specific details [19].

There are too many MDD approaches in the literature to review in detail, so a list is presented to briefly summarise some of the approaches suggested:

- Collaborative modeling is a collaborative MDD approach where stakeholders involved in system development may collaborate remotely using XML nets [S22].
 - Semantic modeling is a MDD approach that uses semantic technologies to automatically translate between the different abstraction levels in MDA [S37].
 - SOMA Bercovici et al. [S27] propose a MDD approach that complies with IBMs SOMA methodology. It complements all of the SOMA development processes with the added benefits of MDD.
 - Context aware Samyr and Slimane [S30] propose a MDD development approach that can be used to develop context aware SBAs. Contextaware applications have the added benefit of adapting to variable contextual parameters.
 - Web based Tavor et al. [S52] (2008) propose a web based model editor which is part of a model driven service engineering process. It has advantages such as ease of accessibility, zero client foot print and simple consumption.

MDD automates many of the proceses involved in software development such as analysis, design or implementation. However, the use of MDD does not negate the advantages of a software process model for the other processes in the development life-cycle. In conjunction with MDD it is often necessary to implement processes such as requirements engineering, operation and management, or adaptation all of which would benefit from the adherence to a suitable process model.

6.1.4. Deployment and Provisioning

Papazoglou and van den Heuvel [S41] describe service provisioning as a process that contains practices such as service metering, service rating and service billing. In other words it is an activity that allows service providers to monitor service usage and charge where appropriate. Provisioning may not be a concern when service producers and consumers are within a single organisation. Service deployment on the other hand involves making services available to service consumers which may include SBAs. There were 19 out of the 57 primary studies which contained details about the activities required for deployment or provisioning.

6.1.5. Execution and Monitoring

The execution and monitoring processes are concerned with the runtime activities of services and SBAs [S41]. Services need to be monitored during runtime in order to ensure that they are available and functioning correctly. Similarly SBAs need to monitor each of their component services to ensure that they are available and functioning correctly. Monitoring can be an automatic or manual process depending on the level of sophistication of the service or application. If the monitoring process for a SBA detects that a service becomes unavailable or its characteristics have changed to a state that are no longer suitable then it may be desirable to adapt the application. Adaptation in the context of (SOC), (covered in more detail in Section 6.1.7) is a process which either automatically or manually facilitates the re-configuration of services within SBAs to a more desirable level. Only four of the primary studies explicitly address Execution and Monitoring. However, nine primary studies address adaptation which is implicit in Execution and Monitoring. Curiously, only one of the papers that explicitly supports Execution and Monitoring also included processes supporting adaptation [S38].

The *Security* process is usually regarded as an ongoing concern for services and SBAs. As with traditional software systems, security has to be considered at each stage of the development life-cycle. Fernandez et al. [S25] propose the extension of a secure development methodology to the development of SBAs. Delessy and Fernandez [S9] propose a pattern-driven security process for SBAs.

6.1.6. Formal Methods

Formal Methods (FM) is a software engineering discipline that centres around mathematical techniques that can be applied to processes such as formal specification, development and verification of software systems [20]. These techniques are expensive to implement so they are usually reserved for safety or security critical applications. Due to the complexity of implementing these techniques they are applied sparingly. However, there have been many applications of FM proposed to aid in the analysis, design and construction of SBAs. FM techniques such as formal specification can be beneficial for the construction of SBAs as they will ensure that the complex applications are constructed correctly. The SENSORIA project [S38], amongst others, proposes a method for Augmenting Service Engineering with various types of formal methods for the analysis, transformation and dynamicity of service-oriented systems. Formal methods are often used in conjunction with other software engineering techniques for the development of SBAs. For example, Nguyen et al. [S29] propose an SBA engineering methodology developed using techniques from formal methods as well as Model Driven Architecture (MDA). One of the problems with using Formal Methods during the development of SBAs or using Formal Methods in general is the steep learning curve involved. Since Formal Methods uses mathematics in order to validate or specify software applications this can be off-putting for practitioners.

6.1.7. Adaption

One of the research questions set out at the beginning of this review was whether or not the software process models encountered would facilitate the adaptation of SBAs. Adaptation has varying definitions depending on the context in which it is used. In terms of SOC, adaptation is most often defined as the enactment of an adaptation strategy in order to satisfy adaptation requirements [21]. Adaptation requirements can range from failure recovery, to the consumption of services from providers who offer better business propositions. Prior to the emergence of SOC, software adaptation has been defined as a discipline that provides techniques which enable the reuse of existing pieces of software to create systems with new functionality [22]. This type of adaptation is facilitated through the use of software adaptors which enable software component reuse. Prior to the concept of adaptation, software engineers have been using Component-Based Software Engineering (CBSE) [23] or Component-Based Development (CBD) [24] to reuse software components. Traditionally, it has been very difficult for software engineers to find software components that exactly meet their requirements so implementing CBSE/CBD has proven difficult. Software adaptation, however, solved this problem by facilitating the adaptation of components to meet the necessary requirements. Adaptation of SBAs can be achieved with greater ease than traditional component based systems because services implementations such as Web Services are loosely coupled and expose standard interfaces. Standard interfaces provide greater interoperability between service providers and consumers, thus alleviating the requirement for components to be adapted. SBAs can easily be adapted by exchanging their services for alternative services that expose the required functionality.

Services can also be adapted; as we have seen SBAs can be adapted by re-arranging their component services. Chang and Kim [S14] outline a service-oriented analysis and design process for the development of adaptable services. In certain circumstances it may also be desirable to adapt services in order to meet the requirements of SBAs. In a scenario where a service has many service clients which all may differ slightly, it may be beneficial for the service to adapt to meet each of the individual client's requirements. This, however, may also be viewed as suboptimal design as there can be many versions of the same service or many similar versions in operation at the same time. This can lead to problems

Table a	8
---------	---

Adaptation specific process model summary.

with maintainability as well as problem resolution. A detailed narrative summary of each of the process models from the selected studies follows in Section 6.2.

6.2. RQ-2: Adaptation process models

In this section our second research question is addressed. Process models from the selected studies which contain adaptation related processes are critically reviewed with emphasis on the approaches taken as well as any validations that have been conducted. A summary of these process models is illustrated in Table 8 with detailed reviews in the following sub-sections.

6.2.1. ProDAOSS

Achbany et al. [S11] propose a Process for Developing Adaptable and Open Service Systems (ProDAOSS) as a plugin for the I-Tropos framework. I-Tropos is a comprehensive end to end agent-oriented development methodology which assigns a crucial role to requirements analysis and specification processes [25]. The ProDAOSS process model concentrates on three process areas. The first is Organizational Modeling and Requirements Engineering using a framework called FaMOs which models at the service level. The second contribution of ProDAOSS is an Architectural Design process which uses a Multi-Agent Software (MAS) architecture to facilitates the adaptation of SBAs. This aspect of ProDAOSS appears to focus on providing an architecture rather than specifically outlining the process details of how to implement it in an adaptable SBA. Finally, the last item described in ProDAOSS is a Detail Design process area that contains process details for a Reinforcement Learning Model which focuses on exploration and exploitation and a Probabilistic Reputation Model used to estimate the reputation of a service providers Quality of Service (QoS). The ProDAOSS process model has been applied to two case studies which contributes towards its validity.

6.2.2. PLASTIC

Autili et al. [S1] take a different approach to adapting SBAs where the services rather than the applications themselves adapt. When an application's context changes, adaptable SBAs should adapt by choosing a service more suited to the new context. However, if individual service adapt to meet context changes it negates the need to adapt the application itself. The advantage of service adaptability over SBA adaptation is that it is more straight forward to implement adaptable services, however adaptable SBAs are more flexible and allow the dynamic binding of services from alternative providers during run-time. The PLASTIC development process addresses adaptation in relation to context change only, and does not support other forms of adaptation. It consists of a design phase where all possible run-time contexts are models, where these model are used to automatically generate code skeletons that are manually coded to satisfy each context type. Then, at run-time, the code segment that best suits the current context is executed to achieve the desired functionality. It is a useful adaptation approach

Adaptation process model	References	Adaptation approach	Based on	Development approach	Evaluation
ProDAOSS	[S11]	SBA adaptation	I-Tropos	Agent-oriented development	Case study
PLASTIC	[S1]	Service adaptation	Bespoke	Other	None
Muliview SOAD	[S6]	Service adaptation	SOAD	MDD	None
BCDF	[S13]	SBA adaptation	Bespoke	Other	Worked example
Chang et al.	[S14, S16]	Service adaptation	SOAD	Other	None
CSOMA	[S32]	Service adaptation	SOMA	MDD	None
CSOA	[S30]	SBA adaptation	Bespoke	MDD	Worked example
Dino	[S38]	SBA adaptation	Bespoke	MDD	None

but is limited to context only adaptation and by the fact that the services adapt rather than the SBA itself.

6.2.3. Muliview SOAD

Kenzi et al. [S6] propose a service-oriented anaylsis and design (SOAD) development process with the same gaol as PLASTIC to adapt services rather than SBAs Therefore, their approach is subject to the same pros and cons as PLASTIC mentioned in the previous sub-section. Kenzi et al. take a different approach however, opting to for a Model Driven Development (MDD) process. The process focuses on providing a base or core set of functionalities for all service clients with specific functionality being provided by means of multiple views to clients with specific adaptation requirements. The services required to implement these base and multiview functionalities are modeled at design time. Then, these models are used to generate implementation code. Once the services are implemented at runtime the base and multiview services are combined in order to provide the necessary functionality to meet the contextual requirements of each client. This is a useful approach but it lacks the ability to adapt by binding to services which are provided by third parties.

6.2.4. BCDF

The Business Collaboration Design Framework (BCDF) [S13] is a rule based development approach for the development of adaptable SBAs which facilitate ad-hoc business collaborations based on predefined rules. They propose many types of rules such as business rules, operational rules and services rules, all of which are specified using an XML based standard called RuleML. Specifying the rules as RuleML allows them to be transmitted to third parties as well as for use in automated processes. This is a useful approach that can be employed for automatic adaptation as the rules that are encoded in RuleML may be parsed at runtime to guide SBA adaptation. One of the key challenges with this approach is the accurate representation of business rules. If the coded rules do not accurately reflect reality then the consequences of any subsequent adaptation may be undesireable. Orriens et al. [S13] illustrate the feasibility of BCDF with an example in the auto-insurance domain. Unfortunately the example focuses on how rules are encoded and interpreted, rather than how BCDF fits into the overall development process.

6.2.5. Chang et al.

Chang et al. [S14,S16] provide details of a SOAD approach which can be used to develop adaptable services. This is another of those approaches that achieves adaptability through the adaptation of services rather that the adaptation of SBAs. They motivate their work by hilighting the benefits of software reuse; pointing out that adaptable services are more suitable for reuse than static services. They explain that service variability is necessary when there are many clients each with slightly different functional requirements and contexts. The SOAD process that they present contains five phases: Defining Target Services, Defining Unit Services, Planning Service Components Acquisition, Acquiring Service Components and Acquiring Service Components. Throughout the development phases attention is paid to provision of adaptable services that can cope with variation in factors such as compositions, interfaces and logic. This is a useful service adaptation approach and has the added benefit of being very process focused with low level details of adaptation phases, processes, activities and artifacts.

6.2.6. CSOMA

The Contextual Service-Oriented Modeling and Analysis (CSO-MA) [S32] development methodology is a SOMA approach aimed at designing services that can adapt during runtime in order to meet contextual adaptation requirements. The approach, which in-

cludes a UML modeling profile for creating Platform Independent Models (PIMs), proposed an outline for a wider MDD approach. The authors highlight that their PIM modeling profile has advantages of being platform independent (not tied to Web Services) as well as specifically addressing adaptation requirements at design time. The CSOMA development approach consists of process details for three layers which are also common with many MDD approaches: a Computational Independent Model (CIM), their UML based PIM profile, and a Platform Specific Model (PSM).

The CSOMA PIM modeling profile contains modeling constructs which can represent variability in business logic and orchestration logic. When used as part of a wider MDD approach, the modeled variability can be transformed into application code which can facilitate the runtime adaptation of business services. This MDD approach is useful because it hides the complexities of low level implementation technologies from system developers.

6.2.7. CSOA

Like the previous approach (CSOMA), CSOA [S30] is an MDD approach for the context-aware development, however, in this case CSOA relates to the development of context-aware SBAs rather than services. The approach employs viewpoint principles from the Object Management Groups (OMG) Enterprise Collaboration Architecture (EDOC-ECA). Among the many viewpoints used in the modeling process is an adaptation view which facilitate the modeling of Business Components, Contextual Components and Connectors which handle bindings and middleware activities between business and context components. An application of the approach is illustrated using a mobile GPS application as an example, where the application adapts to meet a contextual environment where the user is a wheelchair user. In this case the application adapts by putting more emphasis on map items such as pedestrian crossings which are necessary for wheel chair users while also providing voice input functionality for hands free use. This is a useful approach with the added benefit of providing adaptation at the application level rather than just at the service level. Unfortunately the approach focuses on specifying the MDD architecture and does not give much detail on how the approach fits into the overall development process.

6.2.8. Dino

SENSORIA was a large European project aimed at developing a comprehensive service engineering approach, Wirsing et al. [S38] describe it as:

SENSORIA is developing a novel comprehensive approach to the engineering of service-oriented software systems where foundational theories, techniques and methods are fully integrated into pragmatic software engineering processes.

A key goal of this project was to facilitate the design of adaptable open ended service-oriented systems that can adapt at runtime to meet a changing environmental conditions or by optimizing an existing composition by selecting cheaper or better quality services. A deliverable from the project which attempts to meet these goals is a service broker engine called Dino [26] with an associated development methodology. The approach is based on MDD engineering create models in UML2 using a Mode profile. A Mode in terms of SOC abstract a set of services that contribute to a shared goal. They are particularly useful when modeling systems that self-heal, self-optimise and self-assemble.

This approach has all the advantage of the other MDD approaches such as the abstraction of low level interface details and service descriptions. However, the use of MDD has an associated learning curve which may be off putting to many developers and architects.

7. Discussion

The results of this review have given us a useful insight into the state of the art of the research in the area of service-oriented software process models. What is interesting to note is that out of all of the approaches reviewed, only a few of them [7],[5] appear mature enough for use in the development of real-life SBAs. Arsanjani et al.'s process model is the only one that is based on extensive empirical evidence. They claim that it was constructed based on the experiences of developing hundreds of service based projects. Unfortunately, in their publication they do not clearly show how the processes and activities from SOMA were elicited from the experiences on which it is based. Papazoglou et al.'s SLDC do not draw on empirical evidence but they have based their approach on RUP and CBSE which are both mature proven approaches. Even with the lack of empirical validation, the SLDC is a complete lifecycle model with enough detailed activities to guide the development of SBAs.

On reviewing the data collected in the review, specifically summaries, one of the strongest research themes in the complete dataset is Model Driven Development (MDD) of SBAs. Given the complex standards used in SBAs such as Business Process Execution Language (BPEL) and Web Service Definition Language (WSDL), many authors believe that the model driven approach is the only way to achieve a high level of quality and productivity while developing these systems.

7.1. Literature Gaps

There is a lot of opportunity for further research in this area. Most of the studies that were reviewed only address particular areas of the development life-cycle. Of the studies that attempt to address the entire life-cycle, few of them are validated with real-life scenarios. The process that are spread over many different process models and studies need to be consolidated in order to address the entire SBA development life-cycle. Then, once more complete process models have been developed, they need to be validated so that they can be adopted with confidence by SBA practitioners.

Other obvious areas of opportunity are the phases of the processes that are not addressed often in the literature. The most obvious of these processes are operation and management, and deployment and provisioning. Similarly, the type of research that has been completed in this area are primarily secondary studies. This lack of primary studies suggest that existing primary studies are being repeatedly cited producing similar secondary studies.

7.2. Limitations of the Review

As with SLRs, there are some limitations associated with this SLR. With SLRs in general there is often a lot of subjectivity involved in the study selection as well as the data extraction processes. In order to minimise the subjectivity involved in this study the reviewers strictly adhered to the instructions set out in the review protocol. The majority of the study selection and data extraction activities were conducted by the first author of this report. In order to test for reviewer bias a random selection of studies were reviewed and extracted by the second author and compared to the results of the first author. The difference between the results of the two authors were small enough not to warrant further investigation.

Another limitation of this study is how the search string was constructed and validated. This process often involves trial and error in order to strike the balance between the minimum set of representative results and getting thousands of results which would be impossible to process. During this study it was found that the initial set of terms used were too generic and returned far too many results. For example, using the term "process" alone generates thousands of results, however, using the term "development process" greatly reduces the number of results while keeping relevant studies. The search strings in this study were validated by ensuring that some well known relevant studies were returned with the results from the various electronic databases. The construction and validation of the search string was found to be one of the most critical factors in the design of the review protocol. A high quality search string provides a smaller set of relevant results which allows the reviewer to focus more effort on reviewing the relevant studies rather than spending a lot of time filtering irrelevant studies. The key is not to have an overly conservative search string which could risk the exclusion of relevant studies.

Two exclusion criteria used in this study which had a big impact on the number of results returned were the exclusion of technology-specific and application domain-specific process models. A decision was chosen to exclude these studies in an attempt to identify studies that contained more generic processes that can be used by anyone not just individuals from a specific domain or who employ a certain technology. The downside to this is that many useful approaches may have been excluded at the expense choosing only generic studies. A more positive observation from using these criteria is that many technology specific or domain specific approaches are specialisations of previously reported generic approaches which are already included in the review.

The final noteworthy limitation of this review is that grey literature such as technical reports or unpublished material are excluded from the study. From a search point of view, excluding this material make the SLR more straight-forward and repeatable but at the cost of potentially excluding valuable studies. It is common for process models to be reported in standards documents which are often excluded from research databases so, unfortunately, they are also excluded from this report.

8. Conclusions and future work

There are clearly a lot of studies in the public domain with a many process models that can be used to guide the development of SBAs. This review has identified and categorised the processes within those process models to give an overview of the entire research area. It has also identified studies that contain processes that can facilitate the adaptation of SBAs. This review is a useful resource for practitioners and researchers who want to find process models for specific SBA development processes or the entire SBA development life-cycle. It is a particularly useful resource for those interested in SBA adaptation.

Within the results it was found that there were some strong process themes emerging, such as MDD is a particularly popular theme. Another common approach was the modification of existing development methodologies to meet the needs of SOC. A promising indication from the results is that 8 process models include adaptation processes. This is a welcome result since adaptability is one of the key benefits of using a SOA.

One of they key findings within the study is that the published process models lack empirical validation. Existing approached while promising need to be strengthened with data from empirical research to prove their applicability in real-life scenarios. Future work planned for this research area is to codify the results of this review using content analysis as well as enhancing existing research with empirical data gathered through industry interaction.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube) and has been supported, in part, by Lero – the Irish Software Engineering Research Centre, Science Foundation Ireland Grant No. 03/CE2/1303.1.

We would like to thank the reviewers and Dr. Sarah Beecham for their valuable comments.

Appendix A. Selected studies

- [S1] M. Autili, L. Berardinelli, V. Cortellessa, A. Di Marco, D. Di Ruscio, P. Inverardi, et al., A development process for selfadapting service oriented applications, in: International Conference on Service-Oriented Computing (ICSOC 2007), 2007, pp. 442–448.
- [S2] H. Wada, J. Suzuki, K. Oba, A feature modeling support for non-functional constraints in service oriented architecture, in: IEEE International Conference on Services Computing (SCC 2007), IEEE Computer Society, 2007, pp. 187–195.
- [S3] R. Howard, L. Kerschberg, A framework for dynamic semantic web services management, International Journal of Cooperative Information Systems, 13 (2004) 441–485.
- [S4] G. Engels, A. Hess, B. Humm, O. Juwig, M. Lohmann, J.P. Richter, et al., A method for engineering a true service-oriented architecture, in: International Conference on Enterprise Information Systems (ICEIS 2008), Barcelona, Spain, 2008, pp. 272–281.
- [S5] H. Lu, M. Chhabra, A methodology for agent oriented web service engineering, in: Pacific Rim International Workshop on Multi-Agents (PRIMA 2006), Springer, 2006, pp. 650–655.
- [S6] A. Kenzi, B. El Asri, M. Nassar, A. Kriouile, A model driven framework for multiview service oriented system development, in: International Conference on Computer Systems and Applications, IEEE Computer Society, 2009, pp. 404–411.
- [S7] S.K. Johnson, A.W. Brown, A model-driven development approach to creating service-oriented solutions, in: International Conference on Service-Oriented Computing (ICSOC 2006), Springer, 2006, pp. 624–636.
- [S8] Z.H. Ren, B.H. Jin, J. Li, A new web application development methodology: web service composition, in: International Workshop on Web Services, E-Business, and the Semantic Web (WES 2003), 2004, pp. 134–145.
- [S9] N.A. Delessy, E.B. Fernandez, A pattern-driven security process for SOA applications, in: Proceedings of the 2008 ACM Symposium on Applied Computing, ACM, Fortaleza, Ceara, Brazil, 2008, pp. 2226–2227.
- [S10] L. Bahler, F. Caruso, J. Micallef, A practical method and tool for systems engineering of service-oriented applications, in: The 8th International Conference on Web Information Systems Engineering, Springer, 2007, pp. 472–483.
- [S11] Y. Wautelet, Y. Achbany, J. Lange, M. Kolp, A process for developing adaptable and open service systems: application in supply chain management, in: International Conference on Enterprise Information Systems (ICEIS 2009), Springer, Milan, Italy, 2009, pp. 564–576.
- [S12] M. Karam, S. Dascalu, H. Safa, R. Santina, Z. Koteich, A product-line architecture for web service-based visual composition of web applications, Journal of Systems and Software, 81 (2008) 855–867.
- [S13] B. Orriens, J. Yang, M. Papazoglou, A rule driven approach for developing adaptive service oriented business collaboration, in: International Conference on Service-Oriented Computing (ICSOC 2005), 2005, pp. 182–189.
- [S14] S.H. Chang, S.D. Kim, A service-oriented analysis and design approach to developing adaptable services, in: IEEE Interna-

tional Conference on Services Computing, 2007, pp. 204–211.

- [S15] M. Meisinger, I.H. Kruger, A service-oriented extension of the V-Modell XT, in: 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, IEEE, Tucson, Arizona, 2007, pp. 256–268.
- [S16] C. Soo Ho, A systematic analysis and design approach to develop adaptable services in service oriented computing, in: Congress on Services (Services 2007), 2007, pp. 375–378.
- [S17] S. Chang, S. Kim, A systematic approach to service-oriented analysis and design, in: International Conference on Product-Focused Software Process Improvement (PROFESS 2007), Springer, 2007, pp. 374–388.
- [S18] D. Adamopoulos, An integrated development environment for synthesising web services, in: International Conference on E-Business Engineering (ICEBE 2008), 2008, pp. 776–779.
- [S19] S. Lamparter, Y. Sure, An Interdisciplinary Methodology for building service-oriented systems on the web, in: IEEE International Conference on Services Computing, IEEE Computer Society, 2008, pp. 475–478.
- [S20] W. Chengjun, Applying pattern oriented software engineering to web service development, in: International Seminar on Future Information Technology and Management Engineering (FITME 2008), 2008, pp. 214–217.
- [S21] I. Krüger, M. Meisinger, M. Menarini, Applying service-oriented development to complex systems: BART case study, in: Monterey Conference on Reliable Systems on Unreliable Networked Platforms, Springer, 2007, pp. 26–46.
- [S22] T. Karle, A. Oberweis, Collaborative model driven software development for SOA-based systems, in: International United Information Systems Conference (UNISCON 2008), Klagenfurt, Austria, 2008, pp. 189–200.
- [S23] S. Mittal, Devs unified process for integrated development and testing of service oriented architectures, University of Arizona, 2007.
- [S24] I. Christou, S. Ponis, E. Palaiologou, Experiences of Using the Agile Unified Process in the Banking Sector, Software, IEEE. PP (2009) 1–1.
- [S25] E.B. Fernandez, P. Cholmondeley, O. Zimmermann, Extending a secure system development methodology to SOA, in: Database and Expert Systems Applications (DEXA'07), 2007, pp. 749–754.
- [S26] L. Bocchi, J.L. Fiadeiro, A. Lapadula, R. Pugliese, F. Tiezzi, From architectural to behavioural specification of services, in: Proceedings of the Sixth International Workshop on Formal Engineering Approches to Software Components and Architectures (FESCA 2009), 2009, pp. 3–21.
- [S27] A. Bercovici, F. Fournier, A. Wecker, From business architecture to SOA realization using MDD, in: Proceedings of the 4th European Conference on Model Driven Architecture: Foundations and Applications, 2008, pp. 381–392.
- [S28] N. Protogeros, D. Tektonidis, A. Mavridis, C. Wills, A. Koumpis, FUSE: A framework to support services unified process, in: Enterprise Interoperability III: New Challenges and Industrial Approaches, Springer, 2008, pp. 209–220.
- [S29] D. Nguyen, W. van den Heuvel, M. Papazoglou, V. de Castro, E. Marcos, GAMBUSE: a Gap Analysis Methodology for Engineering SOA-Based Applications, Conceptual Modeling: Foundations and Applications, 2009, pp. 293–318.
- [S30] S. Vale, S. Hammoudi, Model driven development of context-aware service oriented architecture, in: The 11th IEEE International Conference on Computational Science and Engineering – Workshops, 2008, pp. 412–418.
- [S31] J.K. Strosnider, P. Nandi, S. Kumaran, S. Ghosh, A. Arsanjani, Model-driven synthesis of SOA solutions, IBM Systems Journal, 47 (2008) 415–432.

- [S32] K. Boukadi, L. Vincent, P. Burlat, Modeling adaptable business service for enterprise collaboration, in: IFIP WG 5.5 Working Conference on Virtual Enterprises (PRO-VE 2009), Springer, Thessaloniki, Greece, 2009, pp. 51–60.
- [S33] E. Colombo, J. Mylopoulos, P. Spoletini, Modeling and analyzing context-aware composition of services, in: International Conference on Service Oriented Computing (ICSOC 2005), Springer, 2005, pp. 198–213.
- [S34] M. Rychly, P. Weiss, Modeling of Service Oriented architecture – from business process to service realisation, in: International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2008), 2008, pp. 140–146.
- [S35] M. López-Sanz, C.J. Acuóa, C.E. Cuesta, E. Marcos, Modelling of service-oriented architectures with UML, in: Proceedings of the 6th International Workshop on the Foundations of Coordination Languages and Software Architectures (FOCLA-SA 2007), Elsevier, 2008, pp. 23–37.
- [S36] R. Huigui, Z. Ning, J. Min, W. Jiaxin, Research on service-oriented framework of interface prototype driven development, in: International Conference on Computer Science and Software Engineering, IEEE Computer Society, 2008, pp. 552–557.
- [S37] C. Pahl, Semantic model-driven architecting of service-based software systems, Information and Software Technology, 49 (2007) 838–850.
- [S38] M. Wirsing, M. Hoelzl, L. Acciai, F. Banti, A. Clark, A. Fantechi, et al., SENSORIA patterns: augmenting service engineering with formal analysis, transformation and dynamicity, in: Leveraging Applications of Formal Methods, Verification and Validation, Third International Symposium, 2008, pp. 170–190.
- [S39] M. Wirsing, M. Hoelzl, N. Koch, P. Mayer, A. Schroeder, Service engineering: the sensoria model driven approach, in: Proceedings of Software Engineering Research, Management and Applications (SERA 2008), Prague, Czech Republic, 2008, pp. XIV–XVI.
- [S40] S. Kambhampaty, Service oriented analysis and design process for the enterprise, in: Proceedings of the 7th Conference on 7th WSEAS International Conference on Applied Computer Science, World Scientific and Engineering Academy and Society (WSEAS), Venice, Italy, 2007: pp. 366-371.
- [S41] M.P. Papazoglou, W.V.D. Heuvel, Service-oriented design and development methodology, International Journal of Web Engineering and Technology, 2 (2006) 412–442.
- [S42] G. Engels, M. Assmann, Service-Oriented Enterprise architectures: evolution of concepts and methods, in: 12th International IEEE Enterprise Distributed Object Computing Conference, IEEE Computer Society, 2008, pp. xxxiv–xliii.
- [S43] A. Arsanjani, A. Allam, Service-Oriented Modeling and architecture for realization of an SOA, in: IEEE International Conference on Services Computing, 2006, pp. 521–521.
- [S44] H. Karhunen, M. Jantti, A. Eerola, Service-oriented software engineering (SOSE) framework, in: Proceedings of the International Conference on Services Systems and Services Management, 2005, pp. 1199–1204 Vol. 2-1199–1204 Vol. 2.
- [S45] A. Erradi, S. Anand, N. Kulkarni, SOAF: An architectural framework for service definition and realization, in: International Conference on Services Computing (SCC 2006), 2006, pp. 151–158.
- [S46] D. Dori, SODA: Not just a drink! From an object-centered to a balanced object-process model-based enterprise systems development, in: Fourth Workshop on Model-Based Development of Computer-Based Systems and Third International Workshop on Model-Based Methodologies for Pervasive and Embedded Software, 2006, pp. 3–14.

- [S47] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, K. Holley, SOMA: a method for developing service-oriented solutions, IBM Systems Journal, 47 (2008) 377–396.
- [S48] I.H. Kruger, R. Mathew, Systematic development and exploration of service-oriented software architectures, in: Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture, 2004: pp. 177–187.
- [S49] J. Koehler, R. Hauser, J. K++ster, K. Ryndina, J. Vanhatalo, M. Wahler, The Role of Visual Modeling and Model Transformations in Business-driven Development, in: Proceedings of the Fifth International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT 2006), Elsevier, Budapest, Hungary, 2008: pp. 5-15.
- [S50] B. Demchak, C. Farcas, I.H. Kruger, The Treasure Map for Rich Services, in: IEEE International Conference on Information Reuse and Integration, 2007: pp. 400-405.
- [S51] M. Deubler, J. Grunbauer, G. Popp, G. Wimmel, C. Salzmann, Tool supported development of service-based systems, in: 11th Asia-Pacific Software Engineering Conference, IEEE Computer Society, Busan, Korea, 2004: pp. 99-108.
- [S52] A. Anaby-Tavor, D. Amid, A. Sela, A. Fisher, K. Zhang, O.T. Jun, Towards a Model Driven Service Engineering Process, in: IEEE Congress on Services – Part I, 2008: pp. 503–510.
- [S53] M. Deubler, J. Grunbauer, G. Popp, G. Wimmel, C. Salzmann, Towards a model-based and incremental development process for service-based systems, in: Proceedings of the IASTED International Conference on Software Engineering, 2004: pp. 183–188.
- [S54] A. Ivanyukovich, G.R. Gangadharan, V. D'Andrea, M. Marchese, Towards a service-oriented development methodology, Journal of Integrated Design and Process Science, 9 (2005) 53–62.
- [S55] V. De Castro, E. Marcos, R. Wieringa, Towards a service-oriented MDA-based approach to the alignment of business processes with IT systems: from the business model to a web service composition Model, International Journal of Cooperative Information Systems. 18 (2009) 225–260.
- [S56] V. Bicer, S. Lamparter, Y. Sure, A.H. Dogru, Towards an interdisciplinary methodology for service-oriented system engineering, in: 24th International Symposium on Computer and Information Sciences, 2009: pp. 486–491.
- [S57] C. Hong-Mei, Towards Service Engineering: Service Orientation and Business-IT Alignment, in: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), 2008: pp. 114–114.

References

- W.S. Humphrey, Managing the Software Process, Addison-Wesley Professional, 1989.
- [2] W.W. Royce, Managing the development of large software systems, in: Proceedings of IEEE Wescon, vol. 26, 1970, p. 9.
- [3] C.P. Team, CMMI for Development, Version 1.2. Tech. Rep. CMU/SEI-2006-TR-008, Software Engineering Institute, Carnegie Mellon University, 2006.
- [4] I. 1/SC, ISO/IEC 15504-5:2006, Information Technology Process Assessment Part 5: An Exemplar Process Assessment Model, International Organization for Standardization, 2007.
- [5] M.P. Papazoglou, W.V.D. Heuvel, Service-oriented design and development methodology, International Journal of Web Engineering Technology 2 (4) (2006) 412–442.
- [6] P. Kruchten, The Rational Unified Process: An Introduction, third ed., Addison Wesley, 2003.
- [7] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, K. Holley, SOMA: a method for developing service-oriented solutions, IBM Systems Journal 47 (3) (2008) 377–396.
- [8] S. Durvasula, M. Guttmann, A. Kumar, J. Lamb, T. Mitchel, B. Oral, Y. Pai, T. Sedlack, H. Sharma, S. Sundaresan, Introduction to Service lifecycle. SOA Practitioner's Guide. Part 3, 2007.
- [9] S. Lane, Q. Gu, P. Lago, I. Richardson, Adaptation of service based applications: a maintenance process? Tech. Rep. Lero-TR-2010-08, Lero, the Irish Software Engineering Research Centre, University of Limerick, 2010

- [10] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering. Tech. Rep., Keele University, 2007.
- [11] J.C. Derniame, B.A. Kaba, D. Wastell, Software Process: Principles, Methodology, and Technology, Springer Verlag, 1999.
- [12] S-Cube Knowledge Model, 2010. <http://www.s-cube-network.eu/km.
- [13] Q. Gu, P. Lago, Exploring service-oriented system engineering challenges: a systematic literature review, Service Oriented Computing and Applications (2009) 1–18.
- [14] T. Dybå, T. Dingsøyr, Empirical studies of agile software development: a systematic review, Information and Software Technology 50 (9–10) (2008) 833–859.
- [15] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: 12th International Conference on Evaluation and Assessment in Software Engineering, 2008.
- [16] D. Budgen, M. Turner, P. Brereton, B. Kitchenham, Using mapping studies in software engineering, in: Proceedings of PPIG, 2008, pp. 195–204.
- [17] G.W. Noblit, R.D. Hare, Meta-ethnography: Synthesizing Qualitative Studies, SAGE, 1988.
- [18] M. Wooldridge, N.R. Jennings, Intelligent agents: theory and practice, Knowledge Engineering Review 10 (2) (1995) 115–152.

- [19] X. Cao, H. Miao, Y. Chen, Transformation from computation independent model to platform independent model with pattern, Journal of Shanghai University (English Edition) 12 (6) (2008) 515–523.
- [20] NASA LaRC Formal Methods Program: What is Formal Methods?. http://shemesh.larc.nasa.gov/fm/fm-what.html.
- [21] S. Benbernou, State of the Art Report, Gap Analysis of Knowledge on Principles, Techniques and Methodologies for Monitoring and Adaptation of SBAs, S-Cube Consortium, 2008.
- [22] C. Canal, J.M. Murillo, P. Poizat, Software adaptation, L'object 12 (1) (2006) 9–13.
- [23] W. Kozaczynski, G. Booch, Component-based software engineering, IEEE Software 15 (5) (1998) 34–36.
- [24] P. Herzum, O. Sims, Business Components Factory: A Comprehensive Overview of Component-Based Development for the Enterprise, John Wiley & Sons, Inc., 2000.
- [25] F. Giunchiglia, J. Mylopoulos, A. Perini, The Tropos software development methodology: processes, models and diagrams, in: Agent-Oriented Software Engineering, LNCS 2585, Springer Verlag, 2002, pp. 162–173.
- [26] H. Foster, A. Mukhija, D. Rosenblum, S. Uchitel, A model-driven approach to dynamic and adaptive service brokering using modes, in: Service-Oriented Computing – ICSOC 2008, 2008. pp. 558–564.