HAL
open science

Web Services Choreographies Adaptation: A Systematic Review

Gustavo Oliva, Fernando Hatori, Leonardo Leite, Marco Gerosa

To cite this version:

Gustavo Oliva, Fernando Hatori, Leonardo Leite, Marco Gerosa. Web Services Choreographies Adaptation: A Systematic Review. [Research Report] 2011. inria-00585829

HAL Id: inria-00585829

https://inria.hal.science/inria-00585829v1

Submitted on 14 Apr 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Web Services Choreographies Adaptation: A Systematic Review

Gustavo Ansaldi Oliva
Fernando Hattori
Leonardo Alexandre Ferreira Leite
Marco Aurélio Gerosa

Technical Report No: RT-MAC-2011-02

Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo (USP)

January 2011

Web Services Choreographies Adaptation: A Systematic Review[1]

Gustavo Ansaldi Oliva
Fernando Hattori
Leonardo Alexandre Ferreira Leite
Marco AurélioGerosa

Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo (USP)

{goliva, hattori, leofl, gerosa}@ime.usp.br

Abstract-*A service choreography describes the observable behavior of a multi-party collaboration from a global perspective. Unlike service orchestrations, each party involved in the collaboration knows exactly when to execute its operations and with whom to interact in order to achieve the established common goal. In this context, especially when a large number of parties is involved, choreographies frequently need to adapt to changes in their definition model or to disturbances in the infrastructure environment. Despite recent research on Service Oriented Computing (SOC), little is known about services choreographies adaptation strategies and their characteristics. In this systematic literature review, we aimed to identify, understand and synthesize the state-of-the-art of this research topic. We systematically and exhaustively examined 5 scientific papers repositories and found 18 relevant papers that matched our inclusion criteria. We classified the adaptation strategies into five categories (model-based, monitor-based, multi-agent-based, formal methods, and miscellaneous) and discussed their influencing factors, limitations, and drawbacks. Furthermore, we also grouped such strategies according to their dynamicity (design-time or runtime)*

Keywords: Service choreography, choreography adaptation, choreography reconfiguration, choreography customization.

[1]The research leading to these results has received funding from HP Brasil under the Baile Project and from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement number 257178 (project CHOReOS - Large Scale Choreographies for the Future Internet).

Table of Contents

1 Introduction 1
2 Background 1
2.1 Web Service Orchestration vs. Web Service Choreography 2
2.2 Systematic Literature Review 2
3 Review Protocol 3
3.1 Research Questions 4
3.2 Inclusion and Exclusion Criteria 4
3.3 Data Sources, Search String and Study Selection Procedures 4
3.4 Citation management and studies retrieval 5
3.5 Study Assessment 6
3.6 Data Extraction 6
3.7 Synthesis of Findings 7
3.8 Protocol Evaluation 7
4 Results 7
4.1 Study Selection Analysis 7
4.2 Overview of Primary Studies 9
4.3 Choreography Adaptation Strategies 13
4.3.1 Model-based choreography adaptation 14
4.3.1.1 [Mahfouz *et al.*, 2009a], [Mahfouz *et al.*, 2009b] 15
4.3.1.2 [Moo-Mena and Drira, 2007], [Moo-Mena *et al.*, 2007] 16
4.3.1.3 [Fabra *et al.*, 2008] 17
4.3.2 Monitor-based choreography adaptation 18
4.3.2.1 [Paspallis and Papadopoulos, 2005] 18
4.3.2.2 [Ezenwoye *et al.*, 2010], [Ezenwoye and Tang, 2010] 19
4.3.2.3 [Yang *et al.*, 2009] 20
4.3.3 Multi-agent based choreography adaptation 21
4.3.3.1 [Xu *et al.*, 2007] 22
4.3.3.2 [Zhao *et al.*, 2009] 22
4.3.3.3 [Morreale *et al.*, 2007] 23
4.3.4 Process calculus and FSM based choreography adaptation 24
4.3.4.1 [Brogi *et al.*, 2004] 24
4.3.4.2 [Salaün and Roohi, 2009] 25
4.3.4.3 [Rinderle *et al.*, 2006a], [Rinderle *et al.*, 2006b] 26
4.3.5 Miscellaneous 26
4.3.5.1 [Nabuco *et al.*, 2008] 27
4.3.5.2 [Decker and Riegen, 2007] 27
4.4 Study evaluation criteria analysis 28
5 Discussion and Lessons Learned 30
5.1 Choreography adaptation strategies 30

5.2 The systematic review process 31
6 Threats to Validity 32
6.1 Search, Selection, and Extraction Biases 32
6.2 Sensitivity Analysis 32
6.3 Limitations of this review 33
7 Conclusion and Future Work 33
Acknowledgements 34
Appendix A – Studies included in the review 35
Appendix B – [Mahfouz *et al.*, 2008a]: Choreography derivation algorithm 36
Appendix C – Notes on WSCI 39
Appendix D – [Brogi et al., 2004]: Web service compatibility analysis 40
Appendix E – Search Engine Forms 41
References 46

1 Introduction

Service oriented architecture (SOA) is an architectural style for the implementation and delivery of loosely-coupled distributed services [Ross-Talbot, 2005]. Common SOA implementations rely on web services and a set of basic core standards (XML, WSDL, UDDI, SOAP, etc.) to enable service interoperability. However, these standards do not provide the rich behavioral detail that describes the role an individual service plays as part of a larger and more complex collaboration [Barker *et al.*, 2009]. The collaborations can be described from the viewpoint of a single participant using orchestration languages (such as WS-BPEL) or from a global perspective using choreography languages (such as WSCI, WS-CDL or BPMN2) [Cambronero *et al.*, 2009].

Service choreographies describe the externally observable peer-to-peer interactions that occur between services that belong to a composition. Therefore, differently from orchestration technologies, a choreography does not rely on a central coordinator. A choreography model describes multi-party collaboration and focuses on the message exchange; each web service involved on it knows when to execute its operations and with whom to interact. A choreography definition can be used at design-time to ensure interoperability among a set of peer services from a global perspective, in a peer-to-peer fashion [Barker *et al.*, 2009], [Peltz, 2003].

There are several challenges to perform adaptations on service choreographies [Srivastava and Koehler, 2003][Papazoglou *et al.*, 2007]. For instance, choreography languages are not usually able to acquire the business-domain knowledge behind the services interaction [Mahfouz *et al.*, 2009a], which raises difficulties to reason if a particular customization satisfies the goals of participants. Assessing the impact of changing one participant's process (local view) on the choreography (global view) as well as other participants' processes is equally challenging. Furthermore, one may want to modify the choreography specification while it is enacted and have the system dynamically reconfigured. For example, interactions between service may be added (choreography extension) or removed (choreography simplification) due to the addition of functionalities to the system at hand or loss of a service. Also, one may want to modify the choreography as a result of services violating predefined QoS threshold values. This reconfiguration usually involves defining a new choreography specification that replaces the previous one and that may lead to the generation of new peers.

This systematic review [Kitchenham and Charters, 2007], [Biolchini *et al.*, 2005] aims to identify and synthesize the existing approaches to service choreography adaptation and their influencing factors. We understand influencing factors as factors that have the ability to interfere/disturb/affect the behavior/performance/effectiveness of a particular choreography adaptation strategy. This work provides an overview of investigated topics, their findings and limitations, as well as implications for research and practice. This kind of study aids practitioners who want to stay up to date and researchers who want to identify topic areas that have been researched or where there is lack of research [Dyba and Dingsøyr, 2008].

This report is organized as follows. In Section 2, we differentiate service orchestrations from service choreographies and present the systematic review methodology. Section 3 describes the systematic review protocol employed in this study. Section 4 reports the findings by presenting an overview of the studies, their characteristics and categorization in groups. Section 5 presents the discussion around the results and the lessons learned; Section 6 presents the possible biases, a sensitivity analysis, and limitations of this study. Section 7 concludes and presents future work directions.

2 Background

In this section we differentiate web service orchestration from web service choreography, as these terms are frequently misused and confusing. Then, we present a brief introduction to systematic literature reviews to clarify the methodology used in this study.

2.1 Web Service Orchestration vs. Web Service Choreography

Although the terms service orchestration and service choreography refer to two distinct concepts, they are often misused and interchangeably employed by some authors. For instance, the IBM WebSphere Process Server[2] product contains a component called "Business Process Choreographer" that supports the modeling of business processes using the Web Services Business Process Execution Language (WS-BPEL). Hence, although the component has "choreographer" in its name, it was conceived only to support orchestrations.

The main difference between the terms regards executability and control. Peltz [2003] defines orchestration as an executable business process that interacts with both internal and external web services, in such a way that the message exchange sequences are controlled and managed by an orchestration engine. The same author states that a choreography tracks the message sequences among multiple parties and sources (typically the public message exchanges that occur between web services) rather than a specific business process that a single party executes. A service choreography is often realized as communicating orchestrations (Figure 1).

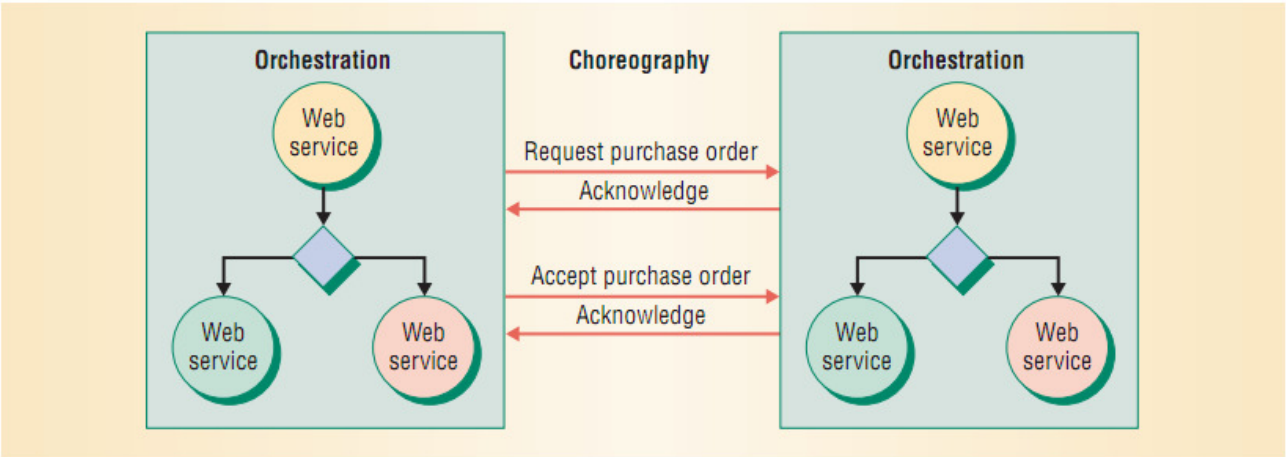

Figure 1: *Orchestration refers to an executable process, while choreography tracks the message sequences between parties and sources [Peltz, 2003].*

W3C defines orchestration as the sequence and conditions in which a web service invokes other web services to realize some useful function. This means that an orchestration is the pattern of interactions that a web service agent must follow to achieve its goal[3]. The W3C's Web Services Choreography Working Group, which was responsible for the specification of the WS-CDL choreography description language, defined choreography as "the sequences and conditions under which multiple cooperating independent agents exchange messages in order to perform a task and achieve a goal state"[4]. The group also states that a choreography description should make it as easy as possible to: promote a common understanding between WS participants; automatically validate conformance; ensure interoperability; increase robustness; and generate code skeletons.

2.2 Systematic Literature Review

The term "Systematic Review" refers to a specific methodology of research, developed to gather and evaluate the available evidence concerning a specific topic [Biolchini *et al.*, 2005]. In contrast to the usual process of literature, a systematic review follows a very well defined and strict sequence of methodological steps that relies on a well-defined protocol. This protocol has a central issue or topic that represents the core of the research, and uses specific concepts and terms. The

[2] http://www.ibm.com/software/integration/wps
[3] http://www.w3.org/TR/ws-gloss/
[4] http://www.w3.org/TR/ws-chor-reqs/

concepts and terms are defined to address information related to specific, predefined, focused, and structured questions. The methodological steps, the strategies to retrieve the evidence and the focus of the question are explicitly defined, so that other professionals may reproduce the same protocol and judge the adequacy of the chosen standards.

The type of acceptable evidence to be gathered in a systematic review is stated beforehand. Besides comparing results of individual studies, different kinds of syntheses can be done. In particular, meta-analysis is a type of research synthesis where the original individual studies are treated as if they were part of a larger study, having their data gathered in a single and final result that summarizes the whole evidence. By selecting studies that are compatible in their quality level, and by taking strict care of their specific details, this methodological procedure produces evidence as well as reveals aspects that the original studies are not individually able to elucidate. For instance, meta-analysis may prove that the results are statistically significant when small studies give inconclusive results with large confidence intervals. Besides, when conflicting results arise from different individual studies, meta-analysis may reconcile the data in a synthetic result.

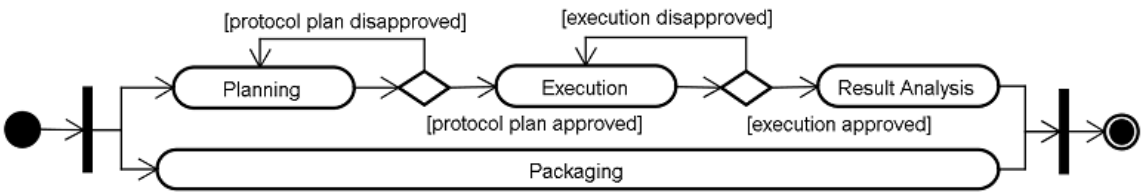

Figure 2: *The systematic review process [Biolchini et al., 2005]*

More information regarding core concepts of systematic review methodology can be found on [Kitchenham and Charters, 2007], [Biolchini*et al.*, 2005], [Petticrew and Helen, 2005], [Higgins and Green, 2008]. Specifically, the first two references present concepts from a Software Engineering viewpoint. According to Biolchini*et al.*(2005), applying systematic reviews in software engineering is much more difficult than in other areas, greatly due to the lack of rigor and conscience when reporting the results of the primary studies. It is also hard to make comparisons without quantitative data and without standardization in the form of presenting the results. According to Richard Hamming [Hamming, 1968], "perhaps the central problem we face in all of computer science is how we are to get to the situation where we build on top of the work of others rather than redoing so much of it in a trivially different way. Science is supposed to be cumulative, not almost endless duplication of the same kind of things".

3 Review Protocol

A systematic literature review evaluates and interprets all available research relevant to a particular question or topic area through a rigorous and auditable methodology. In this section, we briefly describe the review protocol elaborated as part of the planning phase of this study. The review protocol was developed and executed according to the guidelines and hints provided by [Kitchenham and Charters, 2007], [Biolchini*et al.*, 2005] and previous experiences at IME-USP[5] [Steinmacher et al., 2010]. The structure of the protocol is adapted from [Dyba and Dingsøyr, 2008].

[5] The complete version of the protocol is available on http://ccsl.ime.usp.br/baile

3.1 Research Questions

Formulating the research questions is the most important activity during the protocol definition [Kitchenham, 2004]. These search questions guide the systematic review. In our systematic review the research questions were:

Q1) What are the strategies (and respective implementing tools) for the adaptation of services choreographies?

Q2) Which are the influencing factors on the application of each strategy?

3.2 Inclusion and Exclusion Criteria

In this study, we are interested in adaptations that affect an existing choreography. Therefore, we discarded studies focused on topics such as choreography synthesis (or any kind of service composition), orchestration adaptations (outside the choreography context) and pure service adaptation. We defined the following criteria based on the research questions:

i. Papers must be available for download.
ii. Papers must propose or discuss a strategy for the choreography adaptation problem.
iii. Choreography adaptation must be motivated by:
    a. Modifications in a existing choreography definition;
    b. Modifications in the internal processes (e.g. orchestrations) that affect/disturb the choreography description.

3.3 Data Sources, Search String and Study Selection Procedures

We performed searches on the following sources: IEEE Xplore (http://ieeexplore.ieee.org), ACM Digital Library (http://portal.acm.org/dl.cfm?coll=portal), CiteSeerX (http://citeseerx.ist.psu.edu), SciVerse Scopus (http://www.scopus.com/home.url), and Web of Science/ISI Web of Knowledge (http://isiknowledge.com/?DestApp=WOS). A detailed explanation on sources selection and evaluation criteria are given in Sections 2.4 and 2.5 of the complete review protocol.

Medical guidelines recommend considering a question about the effectiveness of a treatment from three perspectives: population, interventions and outcomes. More recently, Petticrew and Roberts [2005] suggested using the PICOC criteria (Population, Intervention, Context, Outcomes, and Comparison) to frame research questions. In this review, the approach to construct the search string was based on [Kitchenham and Charters, 2007] and relies on the PICOC criteria:

1. A structured question SQ in the form of a PICOC statement was derived from the research questions Q1 and Q2 (Section 3.1).
2. The identified synonyms and alternative spellings for each of the SQ keywords were linked using the boolean operator OR.
3. The two OR lists obtained from the previous step were linked using the boolean operator AND.

(

(choreography OR "decentralized composition" OR "decentralized service composition" OR "distributed composition" OR "distributed service composition" OR "decentralized interacting services"**)**

AND

```
        (
        (custom*)
        OR
        ("self-configurable" OR "self-configuring" OR "self-configuration" OR
        "auto-configurable" OR "auto-configuring" OR "auto-configuration" OR
        "self-healing")
        OR
        (adapt* OR reconfig*)
        )
)
```

Table 1: *Search String (* stands for truncation)*

The procedures for study selection were mainly derived from [Dyba and Dingsøyr, 2008]:

1. Query strings were built according to the specific syntax of each selected source (see Appendix E). Results from all sources were then grouped in a single spreadsheet.
2. Duplicated and invalid results were excluded.
3. All clearly irrelevant results were discarded (e.g., non Computer-Science papers).
4. The title of each remaining study was read and a new selection was done based on the inclusion/exclusion criteria.
5. The abstract of every preselected work from the previous stage was read and another selection was conducted based on the inclusion/exclusion criteria. As suggested in [Kitchenham and Charters, 2007], the researchers also downloaded the full article and checked the study conclusions when only reading the abstract was not sufficient to fully understand the objectives or the problems being addressed.
6. If there were multiple versions of the same study, only the most complete version was included.
7. The selected studies were fully read by at least one of the researchers, who became the responsible for writing a structured abstract (executive summary) of the study based on a template already defined (see Section 3.6).

Although steps 2 and 3 are not directly connected to the inclusion/exclusion criteria, their execution contributes to the elimination of completely irrelevant papers. Furthermore, as also suggested in [Dyba and Dingsoyr 2008], this initial "cleaning" supports the elaboration of a more concise and cohesive grouping of excluded studies in rejection categories. An inclusive approach was carried out for the initial selection (stages 3 to 5), i.e. for one article to be preselected, it was sufficient that one of the researchers decided in favor of the inclusion. In adition, all stages of the whole selection process were supervised by a more experienced researcher and all rejected studies were adequately grouped in rejection categories (with exception of irrelevant studies).

3.4 Citation management and studies retrieval

In order to assist the process of gathering and managing all the results from search engines, we used two citation management tools, namely JabRef[6] and Mendeley[7].JabRef is a reference management software that uses BibTeX as its native format. JabRef provides an easy-to-use interface for editing BibTeX files, importing data from online scientific databases, and managing and searching BibTeX files. We used JabRef to store and manage all results (citations) obtained from the query submission to the data sources.

Mendeley is a desktop and web software for managing and sharing research papers, discovering research data and collaborating online. It combines Mendeley Desktop, a PDF and citation management application with Mendeley Web, an online social network for researchers. Mendeley Web Importer, a small “bookmarklet” provided by Mendeley Desktop, was particularly helpful as it helped to

[6] http://jabref.sourceforge.net/
[7] http://www.mendeley.com/

import the results from ACM Digital Library to JabRef (this source did not provide flexible exporting facilities).

3.5 Study Assessment

For each selected study, we conducted an assessment based on questions to be answered with Yes/Partially/No (corresponding to the scores 1.0, 0.5, and 0.0 respectively). This assessment was included in each structured study abstract (see Section 3.6). It is important to highlight that although the total score of a single paper has little meaning, the set of all scores for each question represents a valuable axis of analysis (see Section 4.4).

	Study Assessment	**Score**
Q1	Is there a clear description of the strategy objectives?	
Q2	Is there a description of the applicability context and pre-conditions of the employed strategy? - Applicability context and pre-conditions cannot be identified(score 0.0) - Applicability context or pre-conditions are informally discussed or can be implicitly inferred somehow (score 0.5) - Applicability context and pre-conditions are explicitly described (score 1.0)	
Q3	Is there a description of the limitations and drawbacks of the employed strategy? - Limitations and drawbacks are not explicitly described (score 0.0) - Limitations are drawbacks are explicitly described (score 1.0)	
Q4	How is the strategy evaluated? - No evaluation (score 0.0) - Example of usage/Proof of Concept (score 0.5) - Experiment/study case/formal proof (score 1.0)	
Q5	Is there a discussion on the strategy scalability? - No discussion (score 0.0) - Informal discussion (score 0.5) - Discussion through formal proof or experiments (score 1.0)	
Q6	Does the strategy support dynamic service reconfiguration (including/removing services at runtime for the sake of adaptability)? - No support (score 0.0) - Partial support (score 0.5) - Support (score 1.0)	
Q7	Is the strategy implemented on a tool? - No tool (score 0.0) - A tool (score 1.0)	
	TOTAL	

3.6 Data Extraction

A structured abstract with the following fields was prepared for each primary study:

• Source • Paper Title • Paper Type {Journal article, Conference Paper, Short Conference Paper, Technical Report, PhD Thesis} • Authors • Year • Vehicle • Paper abstract	• Strategy dynamism {design-time, runtime, hybrid} • Description of the tool implementing the strategy • Strategy influencing factors • Strategy applicability context and pre-conditions • Strategy limitations and drawbacks • Study results/conclusion

• Research question/issue • Choreography adaption strategy	• Study assessment • Notes

3.7 Synthesis of Findings

The data extracted from studies was tabulated and plotted in order to present basic information about each work. The studies were also cohesively grouped into categories according to the strategy employed, and each category was discussed. A meta-analysis (statistical analysis of a large collection of analysis results of individual studies to integrate the findings) was conducted to help identify any interesting trends or limitations in current researches. The whole synthesis process was supported by frequent meetings among review authors. The synthesis of findings is presented in Section 4.

3.8 Protocol Evaluation

Before executing the actual systematic review, we have evaluated the protocol as follows:

i. Asking five experts (in systematic reviews and/or service choreographies) to review the protocol

All observations made by the experts were taken into account and most of them were incorporated into the protocol. Some of their contributions were:

- Additional synonyms for the keywords.
- Advices on how to perform the synthesis of findings (e.g. creating rejection categories).
- Advices on the systematic review methodology in general.

ii. Doing a pilot study

We did a pilot study of the systematic review in a single search engine, namely IEEE Xplore. By executing the protocol on this source iteratively, we were able to:

- Expand the review scope.
    - Design-time reconfiguration was incorporated (in the early days of the protocol we intended to identify only runtime choreography adaptation strategies).
- Improve query definition.
    - Several versions of the query were tested.
- Improve inclusion/exclusion criteria.

We have chosen IEEE Xplore because of its credibility and facilities to export the results. Such characteristics enabled rapid feedback and iterative protocol tailoring.

4 Results

After the full protocol execution, the results were summarized and analyzed (see Figure **2**) using the methods and plans defined previously (in the planning phase). In the next sections, we present the study selection analysis, an overview of the primary studies, the description and classification of each study according to its choreography adaptation strategy, and a meta-analysis regarding the study evaluation criteria.

4.1 Study Selection Analysis

We identified 17 primary studies on service choreography adaptation by performing the searches on the selected sources (refer to Section 3.3). Table 2 describes the stages of refinement of the study selection:

Stage 1 Results	Total results obtained from the query submission to the source
Stage 2 Results	Results excluding irrelevant papers
Stage 3 Results	Results excluding duplicated and invalid papers
Stage 4 Results	Results applying inclusion/exclusion criteria to the papers titles
Stage 5 Results	Results applying inclusion/exclusion criteria to the papers abstracts and conclusions

Table 2: *Results of each stage of the study selection process*

Table 3 shows the number of papers returned by each source at the end of each selection stage. The information obtained shows differences between the academic references sources.

Source	Stage 1 Results	Stage 2 Results	Stage 3 Results	Stage 4 Results	Stage 5 Results	% of Selected Results
ACM Digital Library	32	32	32	31	9	53%
CiteSeerX	19	19	18	18	4	24%
IEEE Xplore	37	36	33	32	9	53%
SciVerse Scopus	95	78	65	64	17	100%
Web of Science	45	45	40	37	9	53%

Table 3: *Number of papers in each source for each selection stage*

We discuss some lessons learned about these results (Table 3) on section 5.2.

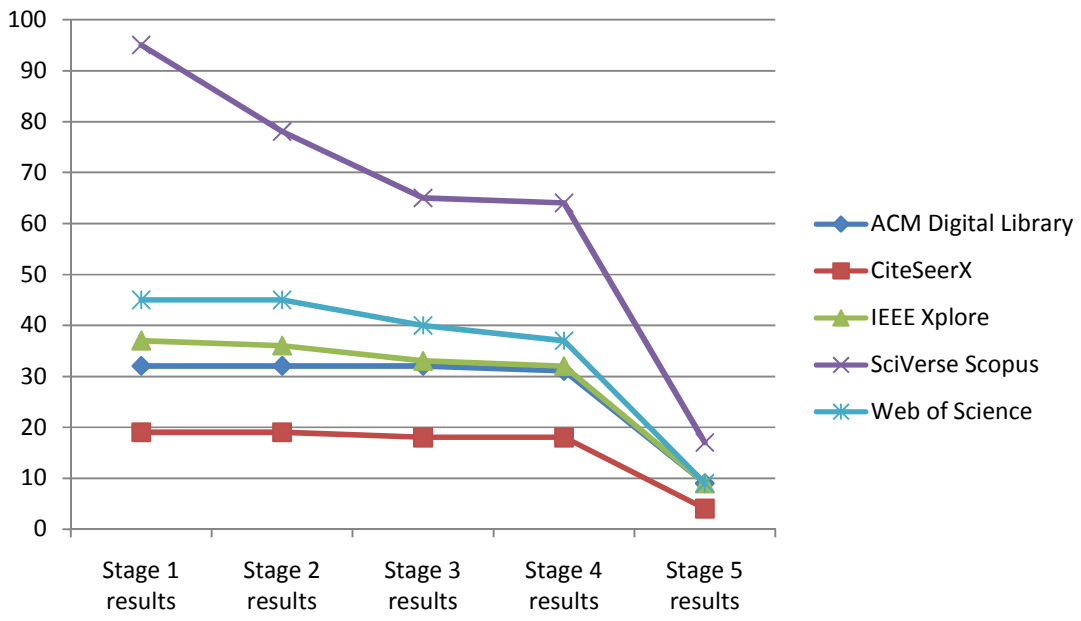

Figure 3: *Quantity of paper versus selection stages*

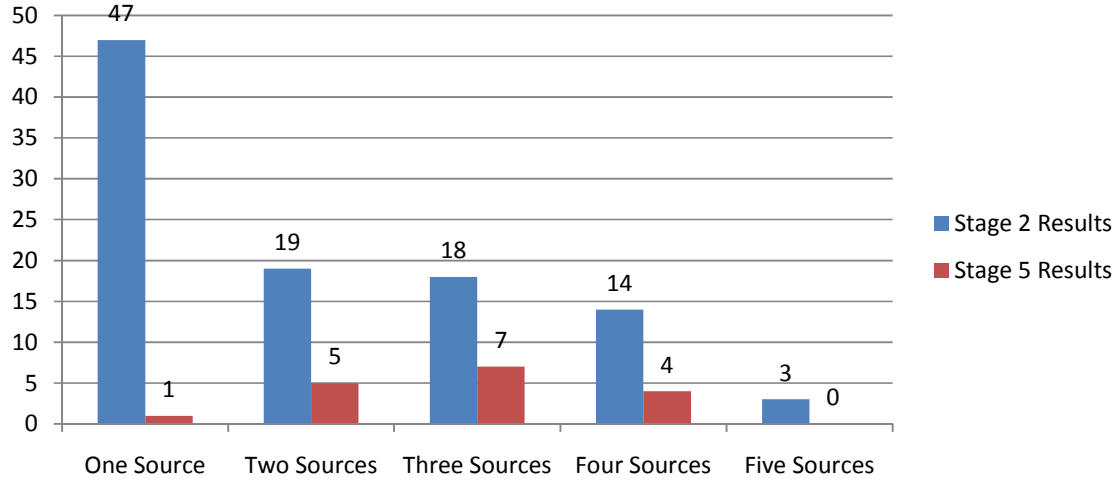

Figure 4:*Number of papers found in stages 2 and 5.*

Figure 4 shows the number of selected papers at the end of stages 2 and 5. While initially 47 studies appeared in only one source, only a single study was exclusive to a source at the end of stage 5. Even though Scopus has a large coverage, this analysis shows that no single source accounts for a large percentage of the selected studies. In particular, stage 5 results shown in Figure 4 fit a normal distribution.

4.2 Overview of Primary Studies

In this section, we provide a brief overview of the primary studies of choreography adaptation that we have selected. We address the questions regarding what kind of papers (conference papers, journal articles, etc.) was published in the area, how many studies were published per year and what were the publication channels. The citation references of the selected studies can be found on Appendix A.

Table 4 shows basic information about the primary studies. In addition to the 17 selected studies, we considered [Salaün and Roohi, 2009] since it was known by the authors, considering previous experiences/studies. Papers were classified as short conference papers (1-4 pages), conference papers (5+ pages), journal articles and technical reports.

#Id	Title	Year	Paper Type
[Xu *et al.*, 2007]	A formal model for dynamic web services composition MAS-Based and Simple Security Analysis Using Spi-Calculus	2007	Short Conference Paper
[Zhao *et al.*, 2009]	A multi-agent collaborative model for Web Services composition	2009	Short Conference Paper
[Paspallis and Papadopoulos, 2005]	Distributed Adaptation Reasoning for a Mobility and Adaptation Enabling Middleware	2005	Short Conference Paper
[Morreale *et al.*, 2007]	Dynamic conversations between agents with the PRACTIONIST Framework	2007	Conference Paper
[Ezenwoye *et al.*, 2010]	Dynamically reconfigurable data-intensive service composition	2010	Conference Paper
[Fabra *et al.*, 2008]	Enabling the Evolution of Service-Oriented Solutions Using an UML2 Profile and a Reference Petri Nets Execution Platform	2008	Conference Paper
[Rinderle *et al.*, 2006a]	Evolution of process choreographies in DYCHOR	2006	Conference Paper
[Brogi *et al.*, 2004]	Formalizing web service choreographies	2004	Journal Article
[Mahfouz *et al.*, 2009a]	From organizational requirements to service choreography	2009	Conference Paper
[Nabuco *et al.*, 2008]	Model-based QoS-enabled self-healing web	2008	Conference Paper

	services		
[Moo-Mena *et al.*, 2007]	Modeling architectural level repair in web services	2007	Conference Paper
[Ezenwoye and Tang, 2010]	Monitoring Decentralized Interacting Services with a Global State Choreography Model	2010	Short Conference Paper
[Salaün and Roohi, 2009]	On realizability and dynamic reconfiguration of choreographies	2009	Conference Paper
[Rinderle *et al.*, 2006b]	On the controlled evolution of process choreographies	2006	Technical Report
[Yang *et al.*, 2009]	Performance prediction based EX-QoS driven approach for adaptive service composition	2009	Conference Paper
[Moo-Mena and Drira, 2007]	Reconfiguration of Web Services Architectures: A model-based approach	2007	Conference Paper
[Mahfouz *et al.*, 2009b]	Requirements-driven collaborative choreography customization	2009	Conference Paper
[Decker and Riegen, 2007]	Scenarios and techniques for choreography design	2005	Conference Paper

Table 4:*Overview of selected primary studies*

Regarding the paper type (see Table 5), 84% of the results consisted of conference papers (both short and complete types). In particular, short papers accounted for approximately 20% of the total results. Only one journal article and a single technical report were found. In particular, the technical report from Yang *et al.* [2009] is a more complete version of the conference paper that had been found in ACM, IEEE, Scopus, and CiteSeerX sources.

Paper Type	Quantity	Percent
Short Conference Paper	4	22%
Conference Paper	12	67%
Journal Article	1	6%
Technical Report	1	6%
PhD Thesis	0	0%
Total	18	100%

Table 5:*Paper type distribution*

With respect to the distribution of studies per year (see Table 6and Figure 5), we notice that there was an increase in the number of papers from 2004 until 2007. While more than a quarter of the studies were published in 2009, only two studies were published in 2008 and 2010. More studies may come up in 2010, then, as the searches in all sources were performed on November 18th, 2010, and sources usually take some time to index new publications.

Publication Year	Quantity	Percent
2004	1	6%
2005	2	11%
2006	2	11%
2007	4	22%
2008	2	11%
2009	5	28%
2010	2	11%
Total	18	100%

Table 6:*Distribution of studies per year*

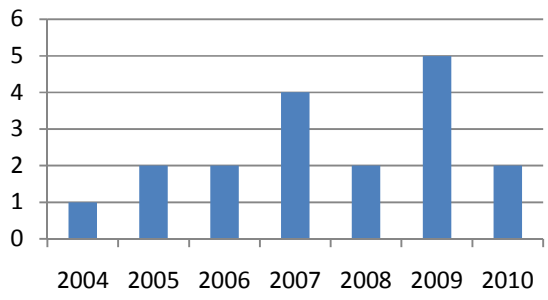

Figure 5: *Number of studies per year*

In relation to publication channels, we notice that apart from the conferences “International Conference on Web Information Systems and Technologies (WEBIST)” and the “On the move to meaningful internet systems (OTM)”, all channels had exactly one paper published (see Table 7). This shows that studies on service choreography adaptation are spread among many different conferences and journals, which makes it harder to gather literature about the topic.

Paper Type	Type	Quantity	Percent
IEEE International Conference on Industrial Informatics (INDIN)	Conference	1	6%
IEEE International Conference on Web Services (ICWS)	Conference	1	6%
IEEE International Symposium on Computers and Communications (ISCC)	Conference	1	6%
International Colloquium on Computing, Communication, Control, and Management (CCCM)	Conference	1	6%
International Conference on Business Information Systems (BIS)	Conference	1	6%
International Conference on Data Engineering (ICDE)	Conference	1	6%
International Conference on Data Engineering (ICDE)	Conference	1	6%
International Conference on Database and Expert Systems Applications (DEXA)	Conference	1	6%
International Conference on Internet and Web Applications and Services (ICIW)	Conference	1	6%
International Conference on Next generation Web Services Practices (NweSP)	Conference	1	6%
International Conference on Web Information Systems and Technologies (WEBIST)	Conference	2	11%
International Joint Conference on Service-Oriented Computing (ICSOC)	Conference	1	6%
On the move to meaningful internet systems (OTM) Confederated International Conferences	Conference	2	11%
Workshop on Autonomic and SELF-adaptive Systems (WASELF)	Conference	1	6%
Electronic Notes in Theoretical Computer Science	Journal	1	6%
Journal of Information Science and Engineering	Journal	1	6%
Total		18	100%

Table 7:*Publication channels distribution*

In the next section, we discuss the identified choreography adaptation strategies and their categorization into groups.

4.3 Choreography Adaptation Strategies

Choreography adaptation strategies encompass approaches and methods to adapt existing service choreographies, either at design-time, runtime or both. Adaptations that happen at design-time are often called “customizations,” while adaptations that occur at runtime are often referred as to “dynamic adaptation” strategies. We employ the term “hybrid” to refer to strategies that support adaptations at both design-time and runtime. As stated in Section 1, we define adaptation influencing factors as factors that have the ability to interfere/disturb/affect the behavior/performance/effectiveness of a particular choreography adaptation strategy.

Design-time adaptation strategies usually provide means (such as graphical models) to facilitate the customization of a choreography specification. This kind of adaptation usually involves generating a new choreography description that replaces the previous one. Runtime adaptation strategies tackle the issues of performing reconfiguration in a way that system functionality is less harmed as possible. In this case, the supporting tools and frameworks are much more complex.

Table 8 shows the distribution of dynamism across selected studies. Approximately little more

than a half of the selected studies deal with runtime adaptation, while only a quarter provide design-time adaptation. Only 2 out of 18 studies deal with hybrid strategies.

Dynamism	Quantity	Percent
Design-time	5	28%
Runtime	11	61%
Hybrid	2	11%

Table 8: *Dynamism distribution*

The strategies found were separated into four distinct and meaningful categories: Model-level based, Monitors based, Multi-agent based (MAS), and Process Calculus& Finite State Models based (PC & FSM).Table 9 and Figure 6 show the distribution of papers in each category.

Choreography Adaptation Category	Quantity	Percent
Multi-agent	3	17%
Model	5	28%
Monitor	4	22%
PC & FSM	4	22%
Miscellaneous	2	11%
Total	18	100%

Table 9: *Choreography adaptation categories distribution*

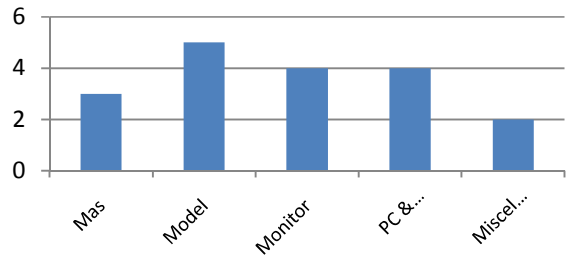

Figure 6: *Choreography adaptation categories distribution*

Model-based choreography adaptation strategies involve rationalizing about a choreography adaptation at a higher level of abstraction (Section 4.3.1). Monitor-based strategies involve delegating the monitoring and dynamic adaptation of the choreography to an entity referred as to “monitor”. Monitors are usually external to the choreography and their failure does not hinder the choreography execution (Section 4.3.2). Multi-agent based strategies involve the use of agents as autonomous, flexible, and adaptable software. The use of agents on web services compositions allows run-time changes, adding, removing, or replacing services according to changes with the environment or changes with the users’ requirements (Section 4.3.3). In the process calculus and finite state models strategies, the proposed choreography adaptation strategies are focused on the notions of process calculus and/or finite state models (FSM). Adaptation strategies are usually developed through a formalization of the choreography specification (Section 4.3.4). Moreover, some uncategorized strategies were found and described in Section 4.3.5.

In the following subsections, we aim to answer the systematic review questions (Section 3.1) by describing the choreography adaptation categories and respective studies. For each study, we present the question/issue being addressed by the authors and the characteristics of the employed adaptation strategy, including applicability context and preconditions, limitations and drawbacks, influencing factors and implementing tools.

4.3.1 Model-based choreography adaptation

According to Bran Selic [2003], modeling technology has matured to the point to offer significant leverage in all aspects of software development. He also argues that, in an increasing number of application areas, one can generate much of the application code directly from models by employing Model-driven Development (MDD) techniques. More precisely, MDD offers the potential for automatic transformation of high-level abstract application-subject models into running systems [Mellor *et al.*, 2003].

Model-based choreography adaptation strategies are directly connected to MDD and involve rationalizing about a choreography adaptation at the model level instead of the specification/code level. There are five studies in this category [Mahfouz *et al.*, 2009a], [Mahfouz *et al.*, 2009b], [Moo-

Mena and Drira, 2007], [Moo-Mena *et al.*, 2007], [Fabra *et al.*, 2008]. The study [Fabra *et al.*, 2008] was classified in the runtime adaptation group, but all the other model-based category studies were considered as performing design-time adaptation, which may reveal that model-based adaptation is a proper technique to operate at design time.

Studies [Mahfouz *et al.*, 2009a], [Mahfouz *et al.*, 2009b] will be presented together, as these are complementing papers from the same authors. Studies [Moo-Mena and Drira, 2007], [Moo-Mena *et al.*, 2007] will also be presented together.

4.3.1.1 [Mahfouz *et al.*, 2009a], [Mahfouz *et al.*, 2009b]

Paper Question/Issue: Ever-changing business needs call for adaptable choreographies. Conventional choreography description languages (CDLs) are not well-suited for adaptation as they embody little domain knowledge required to reason about participants' needs. Specially, the business goals of participants and strategic dependencies motivating the interaction are not explicitly represented. Therefore, it is not clear how to assess the impact of any suggested change to the choreography (global view) on the process of each participant (local view).

The authors developed a choreography customization framework that can be summarized by the following steps (see Figure 7):

1. Representing choreographed interactions at the level of organizational dependencies.
2. Performing required customizations to models of organizational requirements, to benefit from the acquired domain knowledge.
3. Inferring constraints on messaging from requirements models.
4. Deriving the resulting choreography description in an automated manner from the messaging constraints.

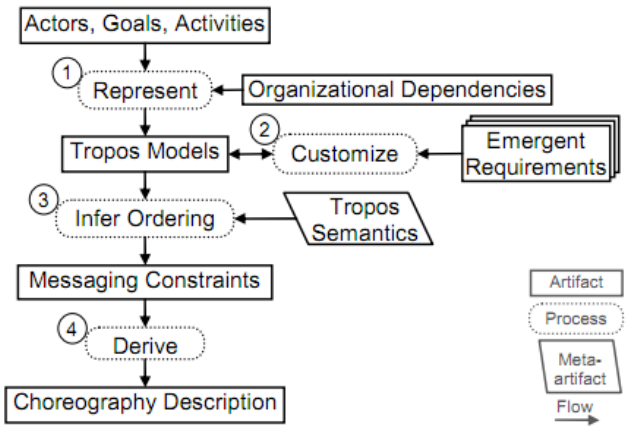

Figure 7: *Overview of the choreography customization framework [Mahfouz et al., 2009a].*

The framework adopts *Tropos* for modeling interaction requirements. *Tropos* is an agent-oriented methodology of software development focused on organizational requirements at various levels of abstraction. *Tropos* models capture goals of participants (actors) in the interaction, mutual dependencies that motivate them to interact, and activities they undertake to achieve their goals, including physical activities not represented in conventional CDLs. The global view of a choreographed interaction is modeled using Actor-Dependency (AD) models; the local view is modeled using Goal-Activity (GA) models; and behavioral obligations of participants are described using Formal Tropos (FT). The formality of FT enables the preservation of consistency between the two representations: organizational requirements and the choreographed-messaging specification. Details about Formal *Tropos* syntax and semantics are given in the paper.

Bridging requirements to choreography allows stakeholders to perform requested customizations to requirements models and to derive the customized messaging. The authors developed a choreography derivation tool that interprets an FT model and outputs the corresponding Abstract CDL (ACDL), in which constructs are directly drawn from WS-CDL[8]. On the other hand, bridging the local and global views helps to ensure that customizations of a choreography description do not violate the goals of any participant.

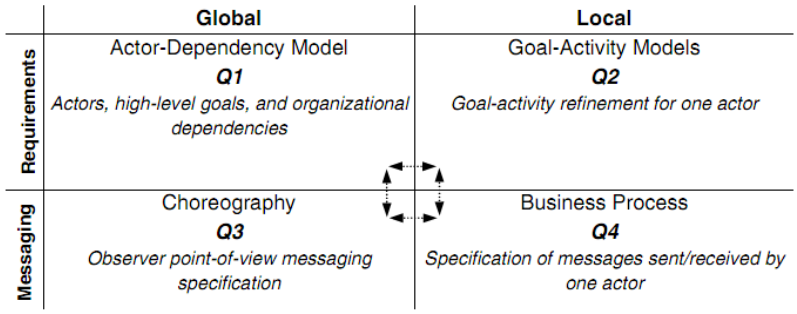

Figure 8: *The four views of the choreography customization framework*

In summary, the proposed choreography customization framework entails: representing choreographed interactions at the level of organizational requirements models; performing required customizations to the models in a collaborative manner that benefits from the acquired domain knowledge; and deriving the resultant choreography description automatically.

With respect to drawbacks, the framework supports only additive customizations (incremental changes to requirements). Moreover, the generated ACDL and the respective derivation algorithm have several limitations: necessity of being refined at the design phase (e.g. by specifying message types); no support for complex messaging patterns (only request-response pattern is supported) and no support for repetition or branching.

Regarding influencing factors, we believe that the performance and effectiveness of the choreography adaptation strategy depend directly on the characteristics of the derivation algorithm implemented by the tool. In addition, the proposed ACDL depends on the expressiveness of *Formal Tropos*. Finally, framework adopters should be able to elaborate and to interpret *Tropos* models.

4.3.1.2 [Moo-Mena and Drira, 2007], [Moo-Mena *et al.*, 2007]

Paper Question/Issue: Web services failures can be caused by a variety of factors, such as network faults, server crashes, web service unavailability, missing data and low QoS. Adequate solutions to these problems require the handling of adaptability at the system architectural level.

These papers present a model-based approach to reconfigure web services compositions, taking actions to repair faults at the architectural level. These repair actions consist of the addition/removal of services or connections between them. In particular, these papers focus on service duplication and substitution. The representation of services compositions is based on an extended UML notation. A graph is used to represent the architecture, while nodes and edges represent web services and their connections respectively. The proposed reconfiguration strategy is achieved through graph transformation actions.

The proposed architecture has three basic elements: the cooperative role, service category, and service class. The cooperative role represents a unit associated to a site wherein actors are connected. The category represents a global and common functionality offered by a set of web services. Service class refers to the functional behavior of a particular type of web service, and it is

[8] Appendix B provides a detailed description of the choreography derivation algorithm.

associated to a service category. Each service is represented by an UML component with tags related to architectural elements.

Some basic rules required to build reconfiguration actions are defined as graph transformations. The first rule (R1) defines the basic actions needed to add a web service to the system. The second rule (R2) deals with the addition of connections between web services. The third rule (R3) defines the necessary actions to remove elements and any associated connections. Therefore, service duplication can be defined as the application of rules R1 and R2 in sequence, while a service substitution can be defined as the application of rules R1, R2, and R3.

The main difference between [Moo-Mena et al., 2007] and [Moo-Mena and Drira, 2007] is that the first one is focused on repairing actions for web service faults, and the second one is focused on repairing actions for QoS mismatches. Moreover, the example scenario of the first study is a food ordering workflow of a food shop company, and the example in the second study concerns the review process in scientific publishing activities. However, the architecture – the graph transformation rules – and the reconfiguration actions proposed in both papers are essentially the same. We were not able to recognize any influencing factors in this study.

4.3.1.3 [Fabra *et al.*, 2008]

Paper question/issue: In highly dynamic scenarios, the companies have to change often to adapt to the environment. The changes in business processes should not require changes in IT infrastructure. Therefore, for enabling flexibility, the business process must be separated from IT infrastructures.

This paper defines orchestration as the description of private business processes followed by each party, and choreography as the description of the interactions that occur in IT infrastructure. Therefore, this paper proposes a set of UML 2 models for separately modeling orchestrations and choreographies, allowing the changes in business processes to occur independently from the interaction logic.

The UML models are developed according to a specific methodology for analyzing complex multi-agent systems called MacMAS, which relies on Agent-Oriented Software Engineering (AOSE) methodology. The methodology is supported by two diagrams: a dynamic one represents interactions or business process behavior throughout time; and the static one describes the relationships between services in a stationary way.

This paper uses extended role models in order to represent services relationships statically. These roles are used at the orchestration level to describe the business process without dealing with needed messages, and also to represent concrete interactions at the choreography level. In addition, UML 2 State Machines are employed to represent the dynamic part.

The authors implemented a platform called DENEB (Development and Execution of iNteroperable dynamic wEB processes) to the execution of the proposed models. Such platform, which is based on Petri-nets, allows changes at run-time and keeps the correct separation between the orchestration and choreography aspects. According to the authors, the main advantage of this approach is the rapid deployment of changes in the business process in a SOA-based platform, enabling the execution of adaptable and agile business process.

The DENEB platform has an enterprise service bus (ESB) responsible for supporting interactions between different components independently from their technology or communication protocols, and separating the logic of the message exchange from the concrete message routing and delivery. Moreover, the platform uses two classes of services engines: the workflows engines, which represent the business logic and processes, and is responsible for orchestration processes execution; and the protocol engine, which represents interaction logic and is responsible for the execution of choreography processes. The presence of these two engine classes allows DENEB to correctly handle and execute choreographies and orchestrations separately.

The translation from models to platform implementations is not direct and is treated as future work. Nevertheless, this integrated platform enables a flexible way to deal with business processes in highly dynamic scenarios. We were not able to recognize any influencing factors in this strategy.

4.3.2 Monitor-based choreography adaptation

Monitor-based strategies involve delegating the monitoring and dynamic adaptation of the choreography to an entity referred to as "monitor." Monitors are usually external to the choreography and their failure does not hinder the choreography execution. Monitors can be implemented in many forms, such as a web service or a middleware.

There are four studies in this category: [Paspallis and Papadopoulos, 2005], [Ezenwoye *et al.*, 2010], [Ezenwoye and Tang, 2010] and finally [Yang *et al.*, 2009]. In this category, the studies [Paspallis and Papadopoulos, 2005] and [Yang *et al.*, 2009] were classified as dealing with runtime adaptations, while studies [Ezenwoye *et al.*, 2010] and [Ezenwoye and Tang, 2010] were classified as presenting a hybrid approach.

Given that three studies are from 2009 or 2010 and two studies are conference short papers, we believe that monitor-based choreography adaptation may be an emerging topic of research. The papers [Ezenwoye *et al.*, 2010] and [Ezenwoye and Tang, 2010] will be presented together, as they are complementing studies of the same leading author.

4.3.2.1 [Paspallis and Papadopoulos, 2005]

Paper question/issue: Adaptive, mobile applications are designed to constantly adapt to the contextual conditions aiming at optimizing the quality of the service offered. Existing approaches depend on rigid composition plans that are defined at design time.

The authors propose a distributed adaptation reasoning strategy based on *utility functions*. Utility functions can be used to compute quantifiable measure of the service quality (utility) as it is experienced by the application users. In this respect, the overall objective of an implementing middleware can be defined as the continuous evaluation, and selection of a composition that maximizes the utility (how this selection is performed is not addressed in the paper). In distributed environments, this implies that the contextual information of all participating hosts must be communicated to the host that performs the adaptation reasoning. The authors suggest two kinds of reasoning: proactive and reactive. Proactive reasoning is more likely to achieve faster and more accurate results, and requires all context data to be communicated as soon as it becomes available. In contrast, reactive adaptation reasoning is better in terms of resource consumption, as it defers the communication of such context data until they are actually needed. An implementation middleware can also benefit from the usage of hybrid approaches or from switching dynamically one approach to the other on demand. As a final point, the authors argue that the proposed strategy provides additional benefits such as robustness, agility, and scalability.

The text does not present an explicit discussion about limitations or drawbacks. With relation to influencing factors, we believe that the amount of context data, which is transmitted from all the participant hosts to the adaptation reasoning host (ARH), may affect the strategy performance (due to network data transmission). Also, the strategy effectiveness is directly connected to the availability of the ARH and the participant hosts ability in sending information to it. No explicit discussion about applicability context and preconditions is presented in the text either. However, the strategy assumes the existence of an ARH (middleware) that is able to compute utility values and to perform distributed reconfiguration based on contextual information received from participating hosts. The discussed adaptation strategy is being implemented in the MADAM middleware as part of the MADAM project[9]. As this is a short paper that presents an ongoing project, no concrete results are provided.

[9] http://www.intermedia.uio.no/display/madam/Home

4.3.2.2 [Ezenwoye *et al.*, 2010], [Ezenwoye and Tang, 2010]

Paper question/issue: Both centralized and decentralized service composition models pose difficulties when promoting ease of dynamic reconfiguration in response to internal and external stimuli. In particular, the peer-to-peer interaction style offered by the choreography model is difficult to manage since no service has a global view of the overall integration**.**

The proposed choreography adaptation strategy is summarized in Figure 9.To propitiate adaptation, the choreography model written in WS-CDL is first transformed before execution. This transformation involves introducing redundancy and other constructions (fault-tolerance pattern) that enable runtime modification of the interaction.

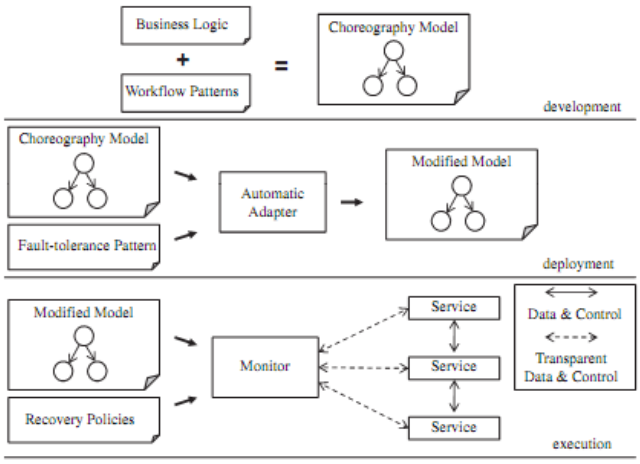

Figure 9: *The architecture of the adaptive service composition framework* [Ezenwoye and Tang, 2010]

The strategy includes a monitoring component that uses the transformed choreography model to have a global view of the service interaction, and introduce adaptive behavior. Such component is a composite service called "Monitor", which models the logic of interaction between the services participating in the choreography, observes the data and control-flow of the choreography, and has the global view of the interaction between the participating services. The monitor observes the interactions and intervenes only if it is necessary to provide the required adaptation to the system. According to the authors, the monitor encapsulates the behavioral and recovery policies that allow for the choreography to be reconfigured. In addition, the authors state that in case of a service failure, the monitor could assign the load to another service and provide it with the information to resume the task to be performed, for example. The use of a global model not only helps with the detection of failed executions, but can also be used to check for violations of desired invariants, producing backup versions of the application, detecting deadlock, and computing global measures such as cost. The monitor does not constitute a single point of failure since multiple monitors could be used. The advantage of this technique is that the logic for adaptive behavior is external to the interacting services.

The authors treat the proposed strategy as a hybrid service composition model, i.e. a combination of choreography and orchestration (Figure 10). The strategy also employs the Observer design pattern, with services playing the role of "observables" by providing notification interfaces through intercession techniques that allow the "observer" (Monitor) to be updated about service state change.

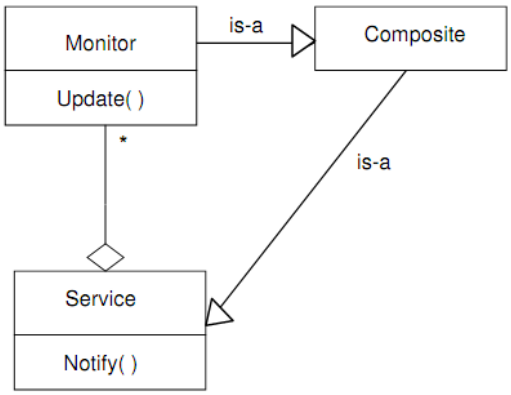

Figure 10:*Class diagram for the hybrid composition model*

No limitations or drawbacks are discussed in the paper. There is no explanation on how the monitor is generated from the choreography model either. Although not explicitly discussed in the paper, we believe that the strategy influencing factors are: availability and quantity of monitors (although the absence of monitors does not adversely affect the choreography); adaptor algorithm and fault-tolerance pattern (which generates adaptation-enhanced choreography model), and recovery policies (adaptive actions, such as changing the endpoint of a service in the choreography by means of interface intercession). Choreographies must be specified using WS-CDL and recovery policies must be defined. It is not clear whether the fault tolerance pattern must be specified or if it is already included in the adaptor algorithm. Since the two studies clearly reveal an ongoing work, no concrete results or prototype tools are given.

4.3.2.3 [Yang *et al.*, 2009]

Paper question/issue: The QoS of a web service usually evolves either because of internal changes or because of workload fluctuations. This paper addresses the issue of how to select services with global optimization of QoS in adaptive distributed service composition (choreographies).

In order to achieve global optimization in decentralized service composition, this paper introduces the “EX-QoS model”, which extends the traditional QoS model by considering the relation of data transmission between services. The proposed strategy works as follows. Firstly, a monitor of QoS-related context collects QoS-related contexts for prediction. Secondly, QoS from each service in the composite are predicted periodically. If the QoS of a certain service is predicted to be a large deviation at invocation time, and the pre-backup replacement (defined prior to the execution of the choreography) is predicted to be unavailable, the process of re-selection is triggered. Then, by taking the original replacement composite service as a reselected slice, a QoS-driven re-selection process based on a heuristic algorithm is done to obtain the new available replacement. Such re-selection process, which preserves the availability of a replacement, is more likely to finish before the invocation of the failed service. Therefore, when the failed service is to be invoked, the composite service will automatically switch to the replacement to heal itself.

The key to the proposed strategy is the performance prediction, which is based on a Semi-Markov probabilistic model. A Semi-Markov process is similar to the Markov process, with the exception of its transition probabilities depending on the amount of time that elapses since the last transition. The basic idea is to predict the data transmission speed of each service in the composite, and when the predicted speed is below the threshold, the confident degree is also low, which means that a failure is likely to happen at invocation time (implying in QoS).

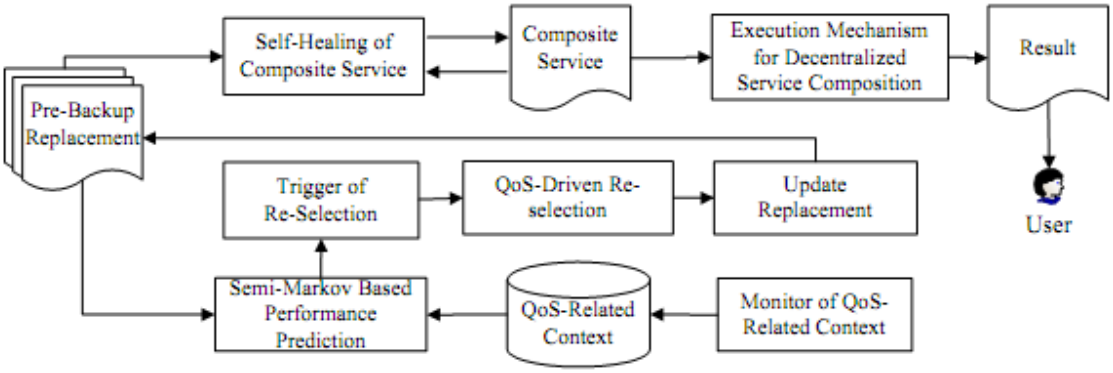

Figure 11: *Ex-QoSstrategy for adaptive service compositions [Yang et al., 2009].*

The limitations of this study are given by the following assumptions: (a) the speed of processing the request is constant; (b) the price of requesting for a service is constant; (c) the execution engine is never failed; (d) the failures by different service and communication links are separate and do not interfere with each other; and (e) during the data transmission process, data transmission speed does not change. Although not explicit in the paper, we believe that the strategy influencing factors include the availability of the QoS monitor and the availability of the pre-backup replacement service. Finally, the authors make clear that the re-selection process is suitable only for "simple" choreographies, as not all choreography constructs are supported.

In summary, the authors propose an adaptive approach for web service composition to adapt to dynamic property of services. The contributions of the paper include: (1) a proposal of an adaptive approach for web service composition; (2) presentation of an EX-QoS model for achieving global optimization in decentralized service composition (and corresponding heuristic algorithm); (3) the introduction of a method of performance prediction to make the re-selection (for updating the replacement) complete as early before the invocation of the failed service as possible.

4.3.3 Multi-agent based choreography adaptation

Multi-agent systems (MAS) are systems where programs called "agents" interact with each other, achieving a set of goals or accomplishment of tasks [Lesser, 2000]. The use of agents offers modularity, which enables the handling of complex, large or unpredictable problems. Each agent has autonomy to use the most appropriate paradigm to solve its particular problem and the coordination is necessary only when interdependency arises [Sycara, 1998].

The use of agents allows run-time changes on web services compositions, such as addition, removal, or replacement of services according to changes with the environment or users' requirements. In the multi-agent adaptation approach, the agents themselves are the main influencing factors when adapting web services choreographies. In these papers, the agents' behavior is not deeply explored.

Three studies explicitly use a multi-agent strategy to dynamically adapt web services choreographies: [Xu *et al.*, 2007], [Zhao *et al.*, 2009] and [Morreale *et al.*, 2007] (all of these studies clearly work with runtime adaptation). The strategies employed are supported by agents that are able to self-update knowledge, learn, self-analyze, and adjust their own behavior according to the environment. In other words, agents are autonomous and have self-adaptation ability, which in turn enables the self-adaptation feature of a multi-agent system [Xu *et al.*, 2007].

4.3.3.1 [Xu *et al.*, 2007]

Paper question/issue: Web services are increasingly being used in a wide range of applications. However, existing web services standards do not handle dynamic compositions. In general, web services compositions address users' request that cannot be accomplished by single services and includes discovering, adding, executing and monitoring a set of services. Thus, because of the complexity of these operations, a human controller handles composition process unsatisfactorily.

This paper proposes a framework using Spi Calculus MAS-based that includes mechanisms such as service selection and dynamic service binding in order to increase the adaptability of web service composition and agile the customers' requirements adaptation. The proposed framework solves the web service dynamic autonomous flexibility problems in two aspects: the agent's structure and the conversations among agents. An agent has self-adaptation ability because he can update, continually learn, he can self-analyze and adjust his own behavior. Further, an agent can learn from other agents their roles or teach him to another one. Therefore, if an agent has some faults, another agent can play the faulty agent's role actively, improving the system robustness.

The CSMWC (collaborative structure of MAS for the web services composition) framework is formally described using Spi Calculus and aims for security-aware dynamic web service composition. The framework structure has four types of agents: user-service-agent, web-service-agent, composite-service-agent, and database-agent. The user-service-agent (or USA), is responsible for interact with the user, providing application domain and achieving users requests. The USA seeks for a composite-service-agent and checks the web-service-agents that can provide web services in order to accomplish USA goals. The web-service-agent (or WSA) offers web services to others. The composite-service-agent (or CSA) is a connector responsible for the management and monitoring, and for guaranteeing that an appropriate WSA is involved, checking the context through a Spi Calculus formal procedure. The database-agent (or DA) facilitates access to the user database, handling registration and removal of negotiator agents from the global agent directory. Using this directory, agents can find available agents to connect.

Regarding influencing factors, we believe that the availability of these four types of agents may influence the effectiveness of the strategy. If some of these agents are unavailable or overloaded, the framework would have some issues that are not addressed in this paper explicitly.

The authors claim to have verified the features of the designed framework through the SPRITE tool. However no description about the validation procedure was explicitly presented.

4.3.3.2 [Zhao *et al.*, 2009]

Paper question/issue: In face to open and dynamic environments, orchestrations or web services' compositions with a centralized control are not flexible enough. This paper proposes a self-adaptive web services' composition model based on a choreography model and multi-agent system techniques.

The paper proposes the creation of three layers: (i) requirements layer, (ii) agent layer, and (iii) web services layer. The roles and their dependencies, in requirements layer, are abstracted from system designers' requirements. In the next layer (agent layer), there are two different Agents: Role Agents and Manage Agent.

The Role Agents are responsible for playing suitable roles, which can be designed with the following contractual responsibilities regarding defined actions: permission, obligation, and prohibition; such responsibilities are expressed in a first-order logic fashion. The Manage Agent is responsible for monitoring system behavior and Role Agents; it also deals with the systems evolution. Web services with similar functions in the web services layer register themselves in the same Role Agent.

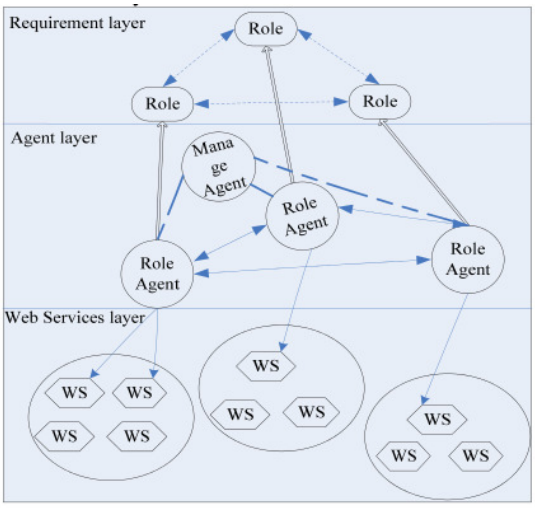

Figure 12:*Proposed model system [Zhao et al., 2009]*

The agents are modeled with the BDI (beliefs, desires, and intentions) architecture and with procedures operating upon a Knowledge Base (described with OWL – the Ontology Web Language) and a Rule Base. The autonomic features are archived by encapsulating actual web services with Role Agents. Thus, when a web service error occurs, the Agent chooses another web service by itself and replaces the faulted one dynamically. If the Role Agent has problems, the Manage Agent adjusts the system organization to achieve the new balance.

With relation to influencing factors, we believe that the availability of the Manage Agent can affect the algorithm effectiveness or performance. This agent centralizes the monitoring activities and is responsible for adjusting multi-agents organization.

4.3.3.3 [Morreale *et al.*, 2007]

Paper question/issue: In a scenario where business relationships are supported by a service-oriented architecture, the currently and widely adopted client/server pattern of communication presents some limitations. This communication does not meet the real needs of business interactions, since a company does not use the other company, but creates a collaborative relation aiming at achieving common objectives.

This paper argues that multi-agent system can provide useful theories and abstractions to web services choreography. Also, the ability of agents to dynamically interact among themselves allows the development of a real, flexible, and adaptive choreography. All the activities that control the conversation flow should automatically be handled by the agent in order for it to adapt itself and participate in unforeseen choreography conversations dynamically. An extended WS-CDL is presented in order to adapt the WS-CDL to the context of multi-agent system. This model requires a clear separation among the semantics of the web services, the interaction protocols, and the service binding.

The framework presented in this paper is the PRACTIONIST (piratical reasoning system), which supports the development of Belief-Desire-Intention agents. Each PRACTIONIST agent has to specify their own goals and the relationships among them; the means to pursue such goals; the *perceptors* to receive stimuli coming from the environment; the actions the agent can perform the corresponding effectors, and the set of rules or beliefs.

The choreography support in the PRACTIONIST framework relies on the idea that the developer should not pay attention to the sequence of messages exchanged: all the activities that control the conversation flow should automatically be handled by the agent, in order to let it dynamically adapt and participate to unforeseen conversations. The developer should be focused only

on defining the agent behavior according to the business logic beyond the conversation.

In this framework, each agent is provided with a choreography engine, which handles an extended WS-CDL. The framework also supports the definition of plans that implement the choreography operations, allowing an agent to participate in conversations and to play the assigned role actively, performing the required operations. We were not able to recognize any influencing factors in this strategy

4.3.4 Process calculus and FSM based choreography adaptation

Process calculus is a diverse family of related approaches to formally modeling concurrent systems. Process calculi provide a tool for the high-level description of interactions, communications, and synchronizations among independent agents or processes [Baeten, 2004]. Process calculi also provide algebraic laws that allow process descriptions to be manipulated and analyzed, and enable formal reasoning about equivalences between processes (e.g., using bi-simulation).In theoretical computer science, a bi-simulation is a binary relation between state transition systems, associating systems that behave in the same way in the sense that one system simulates the other and vice-versa[10]. Intuitively, two systems are bi-similar if they match each other's moves. In this sense, each of the systems cannot be distinguished from the other by an observer.

Leading examples of process calculi include CSP, CCS, ACP, and LOTOS. More recent additions to the family include the π-calculus (pi-calculus), the ambient calculus, PEPA, and the fusion calculus. Depending on the process calculus employed, it is also possible to derive finite state machines, finite state processes, or labeled transitions systems from the formal models. Although the variety of existing process calculi is very large (including variants that incorporate stochastic behavior, timing information, and specializations for studying molecular interactions), there are several features that all process calculi usually share [Pierce, 1996]:

- Representation of interactions between independent processes as communication (message-passing), instead of modification of shared variables.
- Description of processes and systems using a small collection of primitives, and operators for combining those primitives.
- Definition of algebraic laws for the process operators, which allow process expressions to be manipulated using equational reasoning.

In this category, the proposed choreography adaptation strategies are centered on the notions of process calculus and/or finite state models (FSM). Adaptations strategies are usually implemented by means of a formalization of a choreography specification in a process calculus.

There are four studies in this category [Brogi *et al.*, 2004], [Salaün and Roohi, 2009], [Rinderle *et al.*, 2006a], [Rinderle *et al.*, 2006b]. The study [Salaun and Roohi 2009] was classified in the runtime adaptation group, but all the others of the this strategy category were considered as performing a design-time adaptation, what may reveal that process calculus and FSM-based adaptation are also proper techniques to operate at design time.

The studies [Rinderle *et al.*, 2006a] and [Rinderle *et al.*, 2006b] will be presented together, since they are complementing papers of the same authors.

4.3.4.1 [Brogi *et al.*, 2004]

Paper question/issue: The authors address the following questions: What kind of properties can be inferred from the WSCI descriptions[11]? How such properties can be proved? In case two web services are not compatible to interoperate, is there a way to remedy such situation by adapting them?

Replaceability and compatibility are the two flip sides of the interoperability coin. In particular,

[10] http://en.wikipedia.org/wiki/Bisimulation
[11] Notes about WSCI choreography are given in Appendix C.

replaceability refers to the ability of a software entity to substitute another one, in a way that the change is transparent to external clients [8]. This issue is not difficult to solve at the WSDL level, as it simply consists in checking that the interface of a new web service contains all the operations of the service to be replaced. However, the situation is different at the behavioral level:

i. First, we need to check whether the services required by the substitute when implementing the old service's methods are a subset of those required by the old one. Otherwise, it may be necessary to add some new services to the system when replacing the old service with the new one.
ii. The behavior (i.e., relative order among incoming and outgoing messages) of the new-version service should be consistent with that of the old one.

The first issue (i) can be easily managed, since required operations are explicitly declared in the WSDL interface of the services. For the second issue (ii), the authors propose a specific mechanism for behavioral sub typing of processes (less restrictive than bi-similarity). Such mechanism, which relies on CCS process algebra, allows one to decide whether a given web service with behavioral description WS1 can be replaced by another one with behavioral description WS2, while keeping clients unaware of the change. More details about this mechanism are given in the paper.

A formal notion of web service compatibility (in the context of WSCI) is also given in the paper. We will not present the compatibility analysis and adaptor generation algorithm proposed by the authors since they are oriented towards service composition synthesis (see Appendix D).

The translation into CCS presented in the paper should be extended to support full WSCI, including constructs such as correlations, transactions, properties, and others that have not been considered within the study. No influence factors are explicitly given by the authors and we were not able to identify any candidates. The replaceability analysis presented is applicable in the context of WSCI choreographies. Furthermore, one should be able to translate WSCI interfaces into CCS process algebra in order to use the proposed replaceability analysis.

In a nutshell, the authors showed how WSCI Web service choreographies can be formalized using a process algebra approach (CCS) and the benefits that can be obtained from such formalization. These benefits include the definition of compatibility and replaceability tests between Web services and the automatic generation of adaptors that can bridge the differences between a priori incompatible Web services. Also, the authors showed how existing formal methods can be successfully applied in the context of Web services, providing useful and practical advantages.

4.3.4.2 [Salaün and Roohi, 2009]

Paper question/issue: Since choreography specification, correctness, realizability and implementation are crucial issues in SOC (Service-Oriented Computing), this study proposes a dynamic reconfiguration strategy that relies on choreography realizability check techniques.

In this study, a choreography is implemented as a set of independent peer processes and Chor calculus is used to describe such peers from a global view (Chor is a simple process language, and a simplified model of WS-CDL). The study proposes an encoding of Chor into process algebra, enabling the use of existing tools to compare peer behaviors and check choreography realizability. Each peer is described as a Labeled Transition System (LTS).

This paper also proposes some techniques to check if a reconfiguration is possible: a new choreography is defined in Chor language and the realizability of this reconfiguration is checked. A choreography specification is realizable if the reconfiguration maintains the behavior of composed peers, also keeping them free from deadlocks. Such reconfiguration corresponds to the addition or removal of service interactions at run-time (with respect to the original choreography).

The reconfiguration check process receives as input two choreographies (the initial one and

the reconfigured one). A trace which contains the history of interactions of a current execution is obtained from the initial choreography. If the trace can be executed in generated peers from the new choreography, then the new choreography is accepted and the actual peers matching an abstract description are taken from databases of peers, instantiated and executed (using the history stored in trace).

Nevertheless, this check only verifies if the actions before reconfiguration can be reproduced in the new choreography. In order to preserve the expected behavior of the former choreography, the test checks if all traces that can be executed from the former choreography state can also be executed from the new choreography state. A third possible reconfiguration test checks for each new peer if all interactions are reachable from its current state, even after all the modifications (added and removed interactions). We were not able to recognize any influencing factors in this strategy.

4.3.4.3 [Rinderle *et al.*, 2006a], [Rinderle *et al.*, 2006b]

Paper question/issue: Enterprises define choreographies across different organizations. However, the issue raised when one of the partners makes changes in this cross-organizational process is not adequately dealt. In general, it is not possible to assume that changes in own private process have no impact on partner's public or private process.

[Rinderle *et al.*, 2006a] and [Rinderle *et al.*, 2006b] present an approach that allows a controlled evolution of choreographies. The authors discuss how a change in a private process may affect the public processes of the own enterprise and how they can be propagated to others partners' process. In order to be able to determine if a change must be propagated, the papers introduce a formal model based on annotated Finite State Automata.

In the tool implemented and described in these papers, message sequences that can handled by public processes are represented using annotated Finite State Automata (aFSA) in order to be able to reason about the correctness of choreography definitions and changes. A specification of a public process can be obtained from the specification of a private one automatically.

When a process is modified, two steps are necessary: to check if the changes add message sequences to the public process automaton or remove message sequences from it, and to calculate if inconsistencies are raised. If there are inconsistencies, the changes in the public process are propagated to the partner's public process. Due to the autonomy of the partners and to the privacy of business decisions, an automatic adaptation on partner's private process is not desired, but the proposed framework assists the user in correctly accomplishing this task by suggesting adaptations.

In order to check if two public processes are consistent, the two aFSAs of the two processes are intersected. If the intersection is empty (this check is based on automaton emptiness text, or it is checked if the automaton contains a path to final stage), then these two processes are inconsistent. However, if the processes are consistent no change propagation is needed for both additive and subtractive modifications. We were not able to recognize any influencing factors in this strategy.

4.3.5 Miscellaneous

This category encompasses studies that did not fit into any of the previous categories. There are two studies in this category [Nabuco *et al.*, 2008] and [Decker and Riegen, 2007]. In particular, the study from Decker and Riegen discusses which existing techniques from other domains (such as object oriented programming) can be extended in order to deal with choreography adaptation issues. Therefore, since it does not exactly propose an adaptation strategy, no limitations and no influencing factors were identified.

4.3.5.1 [Nabuco *et al.*, 2008]

Paper question/issue: The effectiveness use of SOA requires service guarantees; in particular the services must be available, reliable, secure, and stable, in accordance to QoS requisites. However, some integration issues are still opened, such as dynamic handling of failure and recovery. Self-healability is a property that enables a system to make adjustments to restore itself automatically. This property is not only related to services absence, but also to inability of services to fulfill the QoS requirements

This paper argues that an ontology-based approach has a shared vocabulary, defining a formal language to describe classes' relationships with axioms and rules. Thus, in order to address the self-healing issues, the paper defines an ontology. The self-healing ontology presented has the objective to describe a self-healing model, creating a shared "healing" vocabulary to project developers. This ontology is divided in some levels: each level is responsible for describing one of the characteristics of the healing intervention.

The concepts used to construct this ontology were gathered from the WS-Diamond project (Web Service Diagnosability, Monitoring and Diagnosis). In WS-Diamond, self-healing is addressed in two levels: process and interaction.

The process level acts on the BPEL flows using SH-BPEL (Self-Healing BPEL) mechanisms. The mechanism works when the diagnoser component receives a symptom from BPEL or from its internal monitoring routines. Thus, it sends a failure notification to Recovery Selector, which is responsible for determining an appropriate action. The execution engine executes these self-healing actions on web services of a process.

On the other hand, the interaction level acts on communication protocols. It intercepts messages, extending headers with QoS parameters values or processing information from SOAP envelope content. The monitoring-level connectors manage QoS by monitoring, stamping, measuring, and logging requests and responses. Some connectors can be generated and deployed automatically, in order to enable the substitution of a faulty or deficient web service by an equivalent one that does not exhibits the same interface.

The WS-Diamond project proposed a self-healing platform for monitoring web services compositions, executing the diagnosis of functional faults and QoS degradation, and for executing repair actions. We were not able to recognize any influencing factors in this strategy.

4.3.5.2 [Decker and Riegen, 2007]

Paper question/issue: The main motivation for introducing choreographies was to enable a model-driven approach for service design and implementation. However, many application scenarios cannot be covered using such top-down design approaches. Existing services and processes are the starting point for identifying existing choreographies or for creating to-be choreographies.

The authors discuss three different choreography-design approaches: choreography identification, choreography context expansion, and collaboration unification. Missing techniques for their implementation are highlighted.

– Choreography identification

It is an approach to produce a choreography model of an already executing and not-specified choreography, which has several participants that only know about the interactions in which they are directly involved.

The main goal behind choreography identification is to optimize the overall collaboration. Only by having the global picture at hand, partners can see what interactions and dependencies exist globally and which of these might be removed or organized differently.

This process can be done either:

1. based on the process implementations and the processes interfaces or

2. based on the runtime behavior of the processes.

However, techniques for generating choreographies out of interconnected interface processes are missing in several languages.

– Choreography context expansion

This approach aims to broaden the reach of choreography, i.e. makes it applicable to a broader context, enabling new kinds of participants to join the choreography.

We have to consider the processes interfaces and the choreography model in order to get a new expanded choreography.

The presented strategy is about to identify what parts of interface processes apply to which context and to include, then, corresponding interactions into the expanded new choreography. We therefore need process model synthesis techniques, which can be found in the space of object-oriented computing [Koskimies and Makinen, 1994] or in the field of message sequence charts [Uchitel*et al.*, 2003].

The current limitations to this approach are: extensions for choreography languages are missing the notion of explicit variability points, where different variants of the choreography can be defined for different business contexts; process model synthesis techniques have to be conceived for integrating potential interactions into choreographies; and the topic of conformance, between the original choreography and the new one, has to be revisited since bi-simulation-based techniques are too restrictive.

– Collaboration unification

The goal is to enable the collaboration between participants from different and disjoint choreographies through a unified new choreography for all participants. Each disjoint group of collaborating services is called “islands of collaboration”.

Collaboration unification consists of two steps:

1. Choreography models for each island are built based on the minimal interaction constraints identified among participants.
2. The different choreographies have to be merged

The authors have seen that techniques for extracting minimal constraints in choreography and for merging conflicting process model structures into a unified choreography are missing.

4.4 Study evaluation criteria analysis

We evaluated each selected primary study according to seven pre-defined evaluation criteria (Section 3.5) in order to perform a meta-analysis. Table 10 shows the results of such assessment. Since the study from Decker and Riegen (2007) does not explicitly propose an adaptation strategy, it was discarded from the analysis.

In order to make the results of Table 10 more readable, we summarize the criteria questions (from Q1 to Q7) here:

Q1 – clear description of the strategy objectives;

Q2 – description of the strategy applicability context and pre-conditions;

Q3 – description of the strategy limitations and drawbacks;

Q4 – strategy evaluation;

Q5 – discussion on the strategy scalability;

Q6 – dynamic service reconfiguration support;

Q7 – implementation in a tool.

Study	Q1	Q2	Q3	Q4	Q5	Q6	Q7
[Xu *et al.*, 2007]	1.0	0.0	0.0	1.0	0.5	1.0	1.0
[Zhao *et al.*, 2009]	1.0	0.0	0.0	0.5	0.0	1.0	0.0
[Paspallis and Papadopoulos, 2005]	1.0	0.5	0.0	0.0	0.5	1.0	1.0
[Morreale *et al.*, 2007]	1.0	0.0	0.0	0.5	0.0	1.0	1.0
[Ezenwoye *et al.*, 2010]	1.0	0.5	0.0	0.0	0.0	1.0	0.0
[Fabra *et al.*, 2008]	1.0	0.0	0.0	0.5	0.5	0.5	1.0
[Rinderle *et al.*, 2006]	1.0	0.5	0.0	0.5	0.0	0.5	1.0
[Brogi *et al.*, 2004]	1.0	1.0	1.0	0.5	0.0	0.0	0.0
[Mahfouz *et al.*, 2009]	1.0	0.5	1.0	0.5	0.0	0.0	1.0
[Nabuco *et al.*, 2008]	1.0	0.5	0.0	0.0	0.5	1.0	0.0
[Moo-Mena *et al.*, 2007]	1.0	1.0	1.0	0.5	0.0	0.0	0.0
[Ezenwoye and Tang, 2010]	1.0	1.0	0.0	0.0	0.0	1.0	0.0
[Salaün and Roohi, 2009]	1.0	0.5	1.0	0.5	0.0	0.0	0.0
[Rinderle *et al.*, 2006]	1.0	0.5	0.0	0.5	0.0	0.5	1.0
[Yang *et al.*, 2009]	1.0	1.0	1.0	1.0	1.0	1.0	0.0
[Moo-Mena and Drira, 2007]	1.0	1.0	1.0	0.5	0.0	0.0	0.0
[Mahfouz *et al.*, 2009]	1.0	0.5	1.0	0.5	0.0	0.0	1.0
Sum	17.0	9.00	7.00	7.50	3.00	9.50	8.00
Average	1.00	0.53	0.41	0.44	0.18	0.56	0.47
Std. Deviation	0.00	0.36	0.49	0.29	0.29	0.45	0.50

Table 10:*Studies assessment*

Although the scores of a single study have little meaning, a vertical analysis corresponding to the average score and standard deviation value of each evaluation criterion provides meaningful information. Figure 13 shows the average score and standard deviation value for all evaluation criteria.

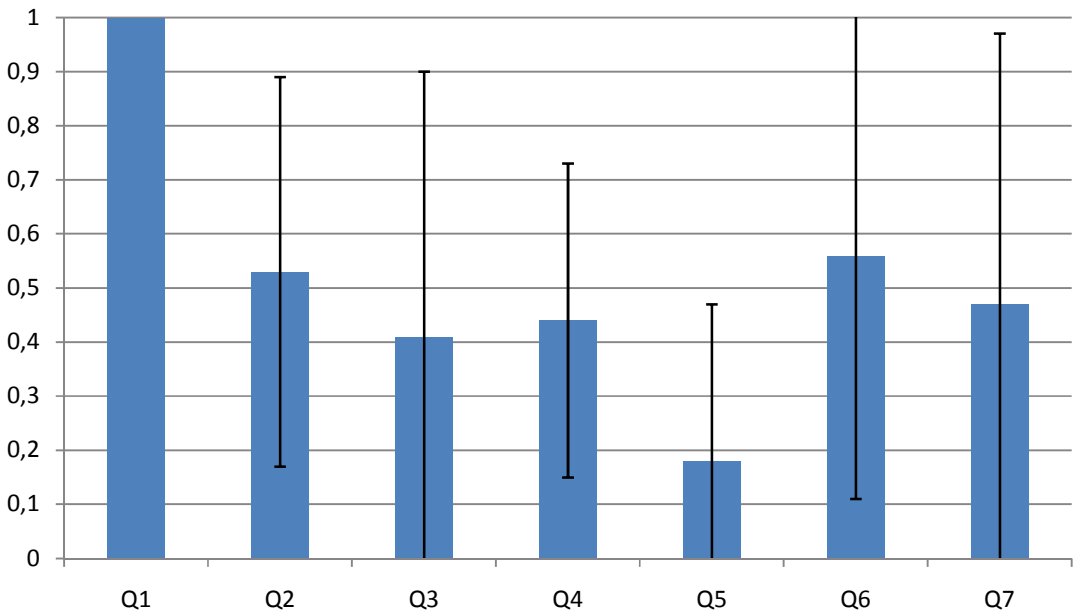

Figure 13:A*verage score and standard deviation value for each evaluation criterion.*

Every paper fully addressed the question Q1, i.e. they presented a clear description of the adaptation strategy goals. Question Q2 refers to the existence of a description of the strategy's applicability context and pre-conditions. The average value is approximately 0.5, which corresponds to

papers having an informal discussion of the applicability context and pre-conditions. However, the standard deviation value is high (0.38), meaning that there is a significant amount of papers whose Q2 value is either 0.0 or 1.0. Question Q3 refers to the existence of an explicit description of the strategy's limitations and drawbacks. Only 7 out of 17 studies explicitly described the limitations and drawbacks of the proposed strategy. This lack of information poses difficulties to choose an adaptation strategy to be employed in a real scenario. The high standard deviation value (0.48) was somehow expected, since this question could only be answered with either 0.0 or 1.0. Question Q4 regards the kind of evaluation method that was used to assess the strategy. The average value is close to 0.5, which corresponds to papers providing an example of usage or a proof of concept of the proposed strategy. Although the standard deviation value is lower than the previous criteria deviation, it still shows some dispersion of the values around the central tendency (Table 11).

Evaluation Method	Quantity	Percent
No evaluation	4	24%
Example of usage/Proof of Concept	11	65%
Experiment/study case/formal proof	2	12%

Table 11: *Strategies Evaluation Method*

Question Q5 concerns the existence of a discussion about the strategy's scalability. Interestingly, around 70% of the studies scored 0.0 in this criterion (Table 12), which corresponds to papers having no discussion about scalability. Since choreographies were designed to support a large number of interacting services acting in a peer-to-peer fashion, this lack of information raises doubts on whether the proposed strategies adequately support large scale choreographies[12].

Scalability Discussion	Quantity	Percent
No discussion	12	71%
Informal discussion	4	24%
Discussion through formal proof or experiments	1	6%

Table 12:*Strategies scalability*

Question Q6 relates to the strategy's support for dynamic service reconfiguration (including and removing services at runtime for the sake of adaptability). The average value is approximately 0.5, which corresponds to papers having a partial support for dynamic service reconfiguration. As only three studies scored 0.5, the standard deviation value went high (0.45). Finally, in question Q7, regards the implementation of the strategy in a tool. Only 8 out of 17 studies provided an implementing tool (the most part are prototypes). The high standard deviation value (0.50) was expected, since this question could only be answered with either 0.0 or 1.0.

5 Discussion and Lessons Learned

In this section we discuss our findings and lessons learned about choreography adaptation approaches and about the systematic review process itself.

5.1 Choreography adaptation strategies

The following are the lessons we learned with regard to choreography adaptation strategies:

[12]CHOReOS is a project that has promoted research on large scale choreographies for the future internet http://www.choreos.eu/bin/view/Main/

Model-Driven Strategies. Although the distribution of the studies by adaptation category is nearly uniform, model-based approaches accounted for more than a quarter of the results (28%). This may show a tendency to model-driven development approaches, given that it leads to development processes in which systems can be designed and understood by business teams in addition to technical teams.

Dynamism. Most of the studies were classified as presenting runtime adaptation strategy dynamism (72%). We also found out relationships between adaptation groups and dynamism categories: monitor and multi-agent based adaptation strategies are always related to runtime dynamism, sometimes presenting hybrid mechanisms. On the other hand, model-based and process calculus and FSM-based adaptations techniques seem to be more tailored to design-time adaptations, although they can also be used in the runtime context.

Influencing Factors and/or Lack Thereof. The main influencing factors we have identified were related to algorithm robustness and availability of system components, such as monitors or agents. However, in many cases it was not possible to clearly recognize the influencing factors, since most part of the studies only present a high-level description of the strategy. In particular, several fundamental aspects are commonly ignored or poorly discussed ("Study evaluation criteria analysis" section), such as scalability, limitations, drawbacks, and implementation. This lack of information occurs because most strategies are still immature and few concrete empirical examinations have been undertaken.

5.2 The systematic review process

The following are the lessons we learned along the five months spent conducting this review:

Process. Practice showed that we should have spent less time elaborating the review protocol and more time conducting iterative pilots. We have also learned that there is no way to proceed with a systematic review when inclusion/exclusion criteria are not clearly stated or not fully understood by every researcher participating in the review. Finally, we have noticed that conducting systematic reviews on emergent research topics is definitely hard, since a common terminology is not well-established. This poses difficulties to the elaboration of the search query and in the precise definition of the inclusion/exclusion criteria.

Sources. Regarding the sources, ACM Digital Library and IEEE Xplore were more accurate than Web of Science, since they returned fewer results in the first place and ended up with the same number of selected results in the last selection stage. In particular, these two sources presented almost the same number of papers in every stage of the selection process. Together, ACM and IEEE accounted for 12 of the 17 selected studies. Although Web of Science returned as much results as ACM and IEEE in the last stage, two of the identified studies were not identified by them. CiteSeerX, which headed the rank of world's top repositories in July 2010, identified only 4 studies in the last selection stage and none of them were exclusive to this source. SciVerse Scopus, which claims to be the largest database of abstracts and citations of scientific studies, was indeed the source that returned the greatest number of results in the first stage (more than twice than Web of Science). Although many results were eliminated of stage 4 to 5 due to the large number of indexed areas, all final selected studies were returned by this source. Therefore, we recommend using both IEEE and ACM for a reasonable research on service choreography adaptation. For a more thorough research in

the area, we recommend using Scopus along with a good citation management tool (e.g., Mendeley Desktop and Zotero).

We have also noticed that most of publication channels had exactly one paper published, what shows that studies on service choreography adaptation are spread among many different conferences and journals, which makes it harder to gather literature material about the topic.

Automation. We noticed that sources such as ACM and CiteSeerX present limited features for the export of results. We used the “Mendeley Web Importer" tool to overcome this kind of issue in an automated manner. In fact, automating data-intensive steps of the review process (such as data-collection and meta-analysis) is especially hard due to a lack of appropriate supporting tools.

6 Threats to Validity

In this section we present the search, selection and extraction biases, which describe decisions and actions that may have interfered in the systematic review results. After that, we present an informal sensitivity analysis. This kind of analysis supports result robustness reasoning, i.e. it helps investigating the uncertainties concerning the inclusion/exclusion of certain studies. Finally, we discuss the limitations of this systematic review.

6.1 Search, Selection, and Extraction Biases

We tried to avoid any bias related to the search string by employing as many synonyms as we could identify for the keywords. The query was also reviewed by more experienced researchers. The only search bias is related to the source selection step in the planning phase. The query was only executed in sources accessible from the Computer Science Department of University of São Paulo.

Although the well-defined inclusive approach for the selection process helped to avoiding partiality, some bias may have been introduced since only two reviewers participated throughout the whole process. More reviewers would allow the improvement of the inclusion/exclusion criteria and reduce bias. Regarding extraction bias, most part of the papers does not explicitly present the applicability context, pre-conditions, limitations, and drawbacks. Therefore, reviewers had to deduce and rationalize about these topics in some occasions. Besides that, all influencing factors were deduced by the reviewers based on previous knowledge of the area. Contacting authors and experts in the area could be an option to avoid this kind of bias. Finally, each reviewer was responsible to read and summarize (in structured abstracts) a certain portion of the selected studies. If all studies had been analyzed by both reviewers, some bias could have been avoided.

Publication bias, which refers to the problem that positive results are more likely to be published than negative results, was not addressed in this systematic review.

6.2 Sensitivity Analysis

Sensitivity analysis deals with result robustness, i.e. it consists in an investigation on whether there were uncertainties about including or excluding certain studies. We conducted an informal sensitivity analysis by computing the Cohen Kappa coefficient of agreement [Cohen, 1960] and the agreement percentage for stages 3, 4 and 5 of the selection process (Table 13).

Variables	Stage 3	Stage 4	Stage 5
Cohen's Kappa coef.	Near complete agreement (0.8060)	Fair agreement (0.2426)	Fair agreement (0.2865)
Standard Error	0.0987	0.0990	0.1089
Agreement	94.06%	84.71%	68.29%

percentage

Table 13: *Sensitivity Analysis*[13]

Both agreement percentage and Kappa coefficient value in stage 3 are high, indicating a great level of agreement between the reviewers. Although the Kappa values for stages 4 and 5 reveal the occurrence of a "fair agreement", the agreement percentages for these same stages are high (84%) and moderate (68%) respectively. A deeper investigation is needed to fully understand the discrepancy between Kappa results and agreement percentage.

The values for the agreement percentages presented in Table 13 were somehow expected, since the selection protocol had defined a more inclusive approach, i.e. controversial papers tended to persist along the selection process. The conflicts on stage 5 were resolved through oral-discussion meetings with the reviewers. Reviewers exposed the reason(s) why they decided upon the inclusion/exclusion of a particular study and then a discussion was carried out in order to reach a consensus.

6.3 Limitations of this review

It is possible that the coverage of this systematic review might be hindered due to the exclusion of rich databases such as SpringerLink, International Journal of Web Service Practices and International Journal of Web Service Research. SpringerLink was discarded because the Department of Computer Science from University of São Paulo has limited access to it (not all papers are available to download). The other two International Journals will be incorporated in a future version of this systematic review. Although we have tried several versions of the search query, it is possible that the employed one is not the most complete due to potential missing synonyms.

In this work, we applied restrictive inclusion/exclusion criteria (Section 3.2). This helped us to separate choreography-focused studies from others that dealt with web service incompatibility or dynamic replacement in more general or non-specified contexts. Nevertheless, the systematic review scope could definitely be expanded by considering less restrictive inclusion/exclusion criteria. For instance, Tosic *et al.* propose two different perspectives on dynamic service composition adaptation, namely (i) the renegotiation of custom-made Service Level Agreements (SLAs) and (ii) the manipulation of service offerings (classes of service) [Tosic *et al.*, 2006].

7 Conclusion and Future Work

Research in service choreography is still in its initial steps. In particular, the adaptation of service choreographies has not been widely explored. By the means of a systematic review, this study provided an exhaustive summary of literature relevant to this topic.

Applying our systematic review protocol, we were able to pinpoint 18 relevant studies to our research topic, which were then classified according to their adaptation strategies and dynamicity. We have noticed that fundamental aspects are commonly ignored or poorly discussed, such as scalability, limitations, drawbacks, and implementation. The deficiency in the treatment of these aspects points opportunities for improvement of existing adaptation strategies, as well as for future research in the area. The main influence factors we have identified were related to algorithm robustness and availability of system components, although in many cases, it was not possible to identify the influence factors. We have also found out that most of the works employs runtime adaptation (72%), although the leading adaptation category (model-based adaptation) is more frequently associated with design-time adaptation.

Results of this systematic review can be applied in distinct ways, such as for aiding the design and implementation of an adaptation strategy. Moreover, the identified influencing factors expose the essential aspects that can interfere in the studied strategies' behavior.

[13] The interpretation of Cohen Kappa values was based on information provided in: http://www.stattools.net/CohenKappa_Exp.php

Future work encompasses incorporating additional search sources and search string keywords synonyms. Occasionally, the review scope could be broadened by applying less restrictive inclusion criteria and considering alterative perspectives on the topic, such as adaptation strategies that result from a renegotiation of custom-made Service Level Agreements [Tosic2006]. Furthermore, rejected studies could be grouped in rejection categories.

Acknowledgements

The research leading to these results has received funding from HP Brasil under the Baile Project and from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement number 257178 (project CHOReOS - Large Scale Choreographies for the Future Internet).

Marco Gerosa has receive individual grant from CNPq.

We would also like to thank both Victoriano A. P. Diaz and Fernando Hattori for working with us in the early versions of this document and Giovanna Avalone for the text review.

Appendix A – Studies included in the review

[Xu *et al.*, 2007] Xu, D.-H.; Qi, Y.; Hou, D.; Chen, Y. & Liu, L.A Formal Model for dynamic Web Services Composition MAS-Based and Simple Security Analysis Using Spi Calculus *Next Generation Web Services Practices, 2007.NWeSP 2007. Third International Conference on,* **2007**, 69 -72

[Zhao *et al.*, 2009] Zhao, L.; Li, B. & Wu, J.A multi-agent collaborative model for Web Services composition *Computing, Communication, Control, and Management, 2009.CCCM 2009. ISECS International Colloquium on,* **2009***, 1*, 145 -148

[Paspallis and Papadopoulos, 2005] Paspallis, N. & Papadopoulos, G. A. Distributed Adaptation Reasoning for a Mobility and Adaptation Enabling Middleware, **2005**.

[Morreale *et al.*, 2007]Morreale, V.; Puccio, M.; Cammarata, G. &Francaviglia, G. Dynamic conversations between agents with the PRACTIONIST Framework *Industrial Informatics, 2007 5th IEEE International Conference on,* **2007***, 2*, 1065-1070

[Ezenwoye *et al.*, 2010] Ezenwoye, O.; Busi, S.; Sadjadi, S. Dynamically reconfigurable data-intensive service composition *WEBIST 2010 - Proceedings of the 6th International Conference on Web Information Systems and Technology,* **2010***, 1*, 125-130

[Fabra *et al.*, 2008] Fabra, J.; Pena, J.; Ruiz-Cortes, A.; Ezpeleta, J. Enabling the Evolution of Service-Oriented Solutions Using an UML2 Profile and a Reference Petri Nets Execution Platform *Internet and Web Applications and Services, 2008.ICIW '08. Third International Conference on,* **2008**, 198 -204

[Rinderle *et al.*, 2006a] Rinderle, S.A.; Wombacher A.B.; Reichert M.B. Evolution of process choreographies in DYCHOR *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* **2006***, 4275 LNCS - I*, 273-290

[Brogi *et al.*, 2004] Brogi, A.; Canal, C.; Pimentel, E. &Vallecillo, A. Formalizing Web Service Choreographies, **2004**.

[Mahfouz *et al.*, 2009a] Mahfouz, A.; Barroca, L.; Laney, R. &Nuseibeh, B. From Organizational Requirements to Service Choreography *Services - I, 2009 World Conference on,* **2009**, 546 -553

[Nabuco *et al.*, 2008] Nabuco, O.; Ben Halima, R.; Drira, K.; Fugini, M.; Modafferi, S. &Mussi, E. Model-Based QoS-Enabled Self-Healing Web Services *Database and Expert Systems Application, 2008. DEXA '08. 19th International Workshop on,* **2008**, 711 -715

[Moo-Mena *et al.*, 2007] Moo-Mena, F.; Drira, K.; Filipe, J&Cordeiro, J. *(Eds.)* Modeling architectural level repair in web services*WEBIST 2007: Proceedings of the Third International Conference on Web Information Systems and Technologies, Vol IT - INTERNET TECHNOLOGY ,* **2007**, 240-245

[Ezenwoye and Tang, 2010] Ezenwoye, O. & Tang, B. Monitoring Decentralized Interacting Services with a Global State Choreography Model *Web Services (ICWS), 2010 IEEE International Conference on,* **2010**, 671 - 672.

[Salaün and Roohi, 2009] Salaün, Gwen. &Roohi, Nima On realizability and dynamic reconfiguration of choreographies, *WASELF-'09,* September 2009.

[Rinderle *et al.*, 2006b] Rinderle, S.; Wombacher, A. & Reichert, M.On the Controlled Evolution of Process Choreographies*Data Engineering, 2006.ICDE '06. Proceedings of the 22nd International Conference on,* **2006**, 124 – 124

[Yang *et al.*, 2009] Yang, L.; Dai, Y. & Zhang, B. Performance Prediction Based EX-QoS Driven Approach for Adaptive Service Composition *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING,* **2009***, 25*, 345-36

[Moo-Mena and Drira, 2007] Moo-Mena, F. &Drira, K. Reconfiguration of Web Services Architectures: A model-basedapproach*Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on,* **2007**, 357 -362

[Mahfouz *et al.*, 2009b] Mahfouz, A.; Barroca, L.; Laney, R. &Nuseibeh, B. Requirements-Driven Collaborative Choreography Customization, **2009**

[Decker and Riegen, 2007] Decker, G. &Riegen, M. V. Scenarios and techniques for choreography design **2007**

Appendix B – [Mahfouz *et al.*, 2008a]: Choreography derivation algorithm

The authors developed a choreography customization framework that can be summarized by the following steps (Figure 14). An overview of (1) and the details of (3) and (4) are presented in the paper, whereas (2) and more validation are given in [Mahfouz *et al.*, 2008b] and [Decker and Riegen, 2007].

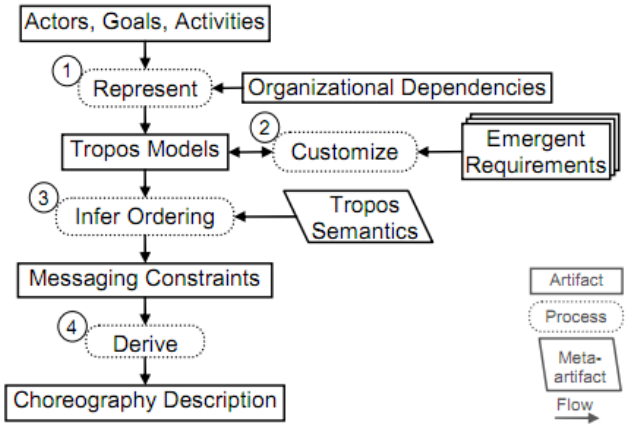

Figure 14:*Overview of the choreography customization framework [Mahfouz et al., 2009a].*

1. Representing choreographed interactions at the level of organizational dependencies
2. Performing required customizations in models of organizational requirements, to benefit from the acquired domain knowledge
3. Inferring constraints on messaging from requirements models.
4. Deriving the resulting choreography description in an automated manner from the messaging constraints.

Ordering constraints on choreographed-messaging are inferred from relations embedded in AD and GA models. More precisely, such constraints are obtained from three kinds of relations between activities. We now present some of the terminology used by the authors and these activities relations:

- Terminology:

Cr(X): creation event of activity X, i.e. when the actor has to start performing the activity

Fi(X): fulfillment event of activity X, i.e. whenever an activity is completed

M^s, M^r: event of sending and receiving a message M respectively

- Relations between activities that embed ordering constraints:

- Dependency: An activity X is said to depend on another activity Y if X requires for its fulfillment information produced when Y is fulfilled, where X and Y are performed by different actors. That is: $Fi(X) \rightarrow Fi(Y) \wedge X.actor \neq Y.actor$

We denote the dependency as D(X,Y)

- Precedence: An activity X is said to precede Y if the fulfillment of X has to precede the creation of X, i.e. $Cr(Y) \rightarrow Fi(X)$. We denote precedence, which is a transitive relation, as Pr(X,Y)

- Refinement: An activity X is said to be refined into sub-activities Y_1 to Y_n if the fulfillment of all the sub-activities leads to the fulfillment of X. That is: $R(X,Y_i)$: $Fi(X) \leftrightarrow Fi(Y_1) \wedge \ldots \wedge Fi(Y_i) \wedge \ldots \wedge Fi(Y_n)$. Also, by definition, X comes to existence before any Y_i. That is, the creation event of each $Y_i$can only occur after that of X. Hence, $Cr(Y_i) \rightarrow Cr(X)$. We use $R(X,Y_i)$ to denote refinement, for all $Y_i$that is a sub-activity of X.

The next step involves inferring message constraints from the activities and their relations. Let *E* denote the set of all events occurring in the interaction. Each element of E can be one of: Cr(X), Fi(Y), M^s or N^r, where X and Y are activities while M and N are messages. We define a partial ordering relation P over events in *E*. Each pair of P is in the form $P(e_1,e_2)$, where each of e_1 and e_2 belong to *E*. $P(e_1,e_2)$ denotes that "event e_1 has to precede e_2". By populating P with the *Tropos* models, constraints on the choreographed messaging are captured. We now present some of the rules stated by the authors for inferring what pairs belong to P.

- Receiving after sending: By definition, a pair $P(M^s,M^r)$ is added to P for every message M to specify that sending of M must precede receiving of M.

- Precedence: For any pair Pr(X,Y), since activity X has to be fulfilled before Y comes to existence, then P(Fi(X),Cr(Y)) belongs to P.

- Refinement: For any pair R(X,Y) the fulfillment of Y is required for the fulfillment of X, which means that the pair P(Fi(Y), Fi(X)) is in P. Also, since X comes to existence before Y the pair P(Cr(X),Cr(Y)) is in P.

The followingstep involves deriving the choreography description from P. Ordering constraints are aggregated into an event precedence Directed Acyclic Graph (DAG) G and the ACDL description is then generated from G. Constructing G comprises (i) performing a transitive reduction on P to remove redundant pairs and (ii) choosing activities to choreograph, since activities fulfilled by means other than service interaction should be discarded in ACDL. The authors propose an algorithm for transforming G into an ACDL description that takes advantage from the potential parallelism while it is enforcing all ordering constraints in G (seeFigure 15). A detailed explanation of the algorithm steps is presented in the paper. Also, advices are given for simplifying the generated choreography, such as merging two consecutive messages (with no intervening physical activity) that are both sent by participant P1 to participant P2 in a single message that aggregates the information held in both messages.

```
Push S to stack
Output root "Sequence"
Assign S to root "Sequence"
While (stack is not empty)
   Pop a vertex v from stack
   Output M(v) as a "Message" to its assigned "Sequence"
   Mark v as processed
   If v has a single child u then
      If all parents of u have been processed then
         Assign u to nearest sequence that encloses all its parents
         Push u to stack
      End If
   End If
   Else If v has multiple children
      Output a "Parallel" to the sequence for v
      For each child u of v
         If u has multiple parents then
            Output "Error: unsupported structure.
         Else
            Create a new "Sequence"
             Assign u to the new sequence
             Push u to the stack
             Output the new sequence as a branch to the "Parallel"
         End Else
      End For Each
   End Else If
End While
If G has any unprocessed nodes
   Output "Error: graph has cycles."
```

Figure 15:

Appendix C – Notes on WSCI

The WSCIchoreography language can raise the level of expressiveness currently provided by web service descriptions. WSCI captures not only static information about the signature and direction of the operations supported by a given web service, but also the dynamic information (i.e. the behavior or protocol of the service [9]). This dynamic information describes the partial order in which messages are expected to be exchanged during the collaborations in which the web service may engage in.

On the top of WSDL, WSCI is a proposal of an XML-based language used to describe the flow of messages exchanged by a web service. It allows the description of the observable behavior of a web service in a message exchange. WSCI describes a one-sided interface for a single web service, i.e., the message exchange is described from the point of view of each web service. In a sense, a WSCI choreography specification is given by a group of WSCI interfaces. In the most common scenario, a web Service has only one WSDL interface and multiple WSCI interfaces for multiple contexts.

A WSCI global model permits a global view of the overall message exchange among the set of web services involved in a conversation. It imports all the WSCI descriptions of all communicating participants, and links the names of individual operations on each service. More precisely, it provides a set of connections (or mappings) between pairs of individual operations from communicating participants.

Appendix D – [Brogi et al., 2004]: Web service compatibility analysis

We will consider that a software system, formed by the composition of several entities specified in process algebra, is compatible when it terminates without requiring any interaction with its environment, i.e. when it always performs a finite number of silent actions τ leading to the inaction 0. However, the majority of client/server systems is not terminating, since servers usually provide their services running on an infinite loop. Therefore, we must extend this definition in order to accommodate also infinite sequences of silent actions. Thus, we give a negative definition, saying that a system (isolated from its environment) fails whenit may perform a finite sequence of silent actions leading to a process which is structurally different from the inaction, and that cannot perform any further action by itself. Now we can say that a system is compatible(or succeeds) when it does not fail. Then, for analyzing compatibility, we proceed recursively by matching the input and output actions that the different system components may perform at a given point, until we arrive either to inaction, or to a repeated state in the analysis.

In the following example, it is evident that both choreography interfaces are not compatible: the local choice in the service between actions bookingConfirmation and bookingRefusal is not supported by the client, who just assumes the former as the only possible answer of the service. More details about this compatibility analysis are given in the paper.

```
Traveller =  TravellerToBS/Book/bookingRequest!().
TravellerToBS/Book/bookingAck?().
TravellerToBS/Book/bookingConfirmation?(). 0

[[ BookTrip ] ] = BStoTraveller/Book/bookingRequest?().
BStoTraveller/Book/bookingAck!().
( τ. BStoTraveller/Confirmation/bookingConfirmation!(). 0
                  + τ. BStoTraveller/Refusal/bookingRefusal!(). 0
                  )
```

By formalizing the WSCI interface of each participant in terms of CCS algebra, the authors' approach aims at providing a notation for specifying the required adaptation between two services in a general and abstract way. The adaptor specification consists of a set of correspondences between actions and parameters of the two services. For instance, the adaptor specification expressing the intended adaptation for the BookTrip process and the following Client process can be simply given by:

```
Client = ClientToBS/BookFlight/request!().
ClientToBS/BookFlight/reply?(answer). 0

S = { request!()           <>bookingRequest?(), bookingAck!();
reply?("Confirmed")  <>bookingConfirmation!();
reply?("Refused")    <>bookingRefusal!() }
```

Appendix E – Search Engine Forms

Search Engine: IEEE XPlorer
Date: November 18th, 2010
Number of Results: 37
Number of Distinct Valid Results: 36
Search String:

```
(
 ("Abstract":choreography OR
  "Abstract":"decentralized composition" OR "Abstract":"decentralized service
composition" OR
  "Abstract":"distributed composition" OR "Abstract":"distributed service
composition" OR
  "Abstract":"decentralized interacting services")
 AND
 (
  ("Abstract":customizable OR "Abstract":customizing OR "Abstract":customization)
  OR
  ("Abstract":"self-configurable" OR "Abstract":"self-configuring" OR
"Abstract":"self-configuration" OR
   "Abstract":"auto-configurable" OR "Abstract":"auto-configuring" OR
"Abstract":"auto-configuration" OR
   "Abstract":"self-healing")
  OR
  ("Abstract":adapt* OR "Abstract":reconfig*)
 )
)
```

Search Result URL:
http://ieeexplore.ieee.org/search/searchresult.jsp?action=search&sortType=&rowsPerPage=37&matchBoolean=true&searchField=Search%20All&queryText=%28%20%28.QT.Abstract.QT.:choreography%20OR%20%0A%20%20.QT.Abstract.QT.:.QT.decentralized%20composition.QT.%20OR%20.QT.Abstract.QT.:.QT.decentralized%20service%20composition.QT.%20OR%20%0A%20%20.QT.Abstract.QT.:.QT.distributed%20composition.QT.%20OR%20.QT.Abstract.QT.:.QT.distributed%20service%20composition.QT.%20OR%20%0A%20%20.QT.Abstract.QT.:.QT.decentralized%20interacting%20services.QT.%29%20%0A%20AND%20%0A%20%28%0A%20%20%28.QT.Abstract.QT.:customizable%20OR%20.QT.Abstract.QT.:customizing%20OR%20.QT.Abstract.QT.:customization%29%20%0A%20%20OR%20%0A%20%20%28.QT.Abstract.QT.:.QT.self-configurable.QT.%20OR%20.QT.Abstract.QT.:.QT.self-configuring.QT.%20OR%20.QT.Abstract.QT.:.QT.self-configuration.QT.%20OR%20%0A%20%20%20.QT.Abstract.QT.:.QT.auto-configurable.QT.%20OR%20.QT.Abstract.QT.:.QT.auto-configuring.QT.%20OR%20.QT.Abstract.QT.:.QT.auto-configuration.QT.%20OR%20%0A%20%20%20.QT.Abstract.QT.:.QT.self-healing.QT.%29%20%0A%20%20OR%20%0A%20%20%28.QT.Abstract.QT.:adapt*%20OR%20.QT.Abstract.QT.:reconfig*%29%0A%20%29%20%0A%29

Applied Filters: None

Search Engine: ACM Digital Library
Date: November 18th, 2010
Number of Results: 32
Number of Distinct Valid Results: 32
Search String:

```
 (
 (Abstract:choreography OR
  Abstract:"decentralized composition" OR Abstract:"decentralized service
composition" OR
  Abstract:"distributed composition" OR Abstract:"distributed service composition"
OR
  Abstract:"decentralized interacting services")
 AND
 (
  (Abstract:customizable OR Abstract:customizing OR Abstract:customization)
  OR
  (Abstract:"self-configurable" OR Abstract:"self-configuring" OR Abstract:"self-
configuration" OR
   Abstract:"auto-configurable" OR Abstract:"auto-configuring" OR Abstract:"auto-
configuration" OR
   Abstract:"self-healing")
  OR
  (Abstract:adapt* OR Abstract:reconfig*)
 )
)
```

Search Result URL:
http://portal.acm.org/results.cfm?within=%28%0D%0A+%28Abstract%3Achoreography+OR+%0D%0A++Abstract%3A%22decentralized+composition%22+OR+Abstract%3A%22decentralized+service+composition%22+OR+%0D%0A++Abstract%3A%22distributed+composition%22+OR+Abstract%3A%22distributed+service+composition%22+OR+%0D%0A++Abstract%3A%22decentralized+interacting+services%22%29+%0D%0A+AND+%0D%0A+%28%0D%0A++%28Abstract%3Acustomizable+OR+Abstract%3Acustomizing+OR+Abstract%3Acustomization%29+%0D%0A++OR+%0D%0A++%28Abstract%3A%22self-configurable%22+OR+Abstract%3A%22self-configuring%22+OR+Abstract%3A%22self-configuration%22+OR+%0D%0A+++Abstract%3A%22auto-configurable%22+OR+Abstract%3A%22auto-configuring%22+OR+Abstract%3A%22auto-configuration%22+OR+%0D%0A+++Abstract%3A%22self-healing%22%29+%0D%0A++OR+%0D%0A++%28Abstract%3Aadapt*+OR+Abstract%3Areconfig*%29%0D%0A+%29+%0D%0A%29&CFID=92392020&CFTOKEN=83137526&adv=1&COLL=DL&qrycnt=48&DL=ACM&Go.x=35&Go.y=0&termzone=all&allofem=&anyofem=&noneofem=&peoplezone=Name&people=&peoplehow=and&keyword=&keywordhow=AND&affil=&affilhow=AND&pubin=&pubinhow=and&pubby=&pubbyhow=OR&since_year=&before_year=&pubashow=OR&sponsor=&sponsorhow=AND&confdate=&confdatehow=OR&confloc=&conflochow=OR&isbnhow=OR&isbn=&doi=&ccs=&subj=

Applied Filters: None

Search Engine: SciVerse Scopus
Date: November 18th, 2010
Number of Results: 95
Number of Distinct Valid Results: 79
Search String:

```
ABS(
(choreography OR "decentralized composition" OR "decentralized service composition"
OR
  "distributed composition" OR "distributed service composition" OR
  "decentralized interacting services")
 AND
 (
  (custom*)
  OR
  ("self-configurable" OR "self-configuring" OR "self-configuration" OR
   "auto-configurable" OR "auto-configuring" OR "auto-configuration" OR
   "self-healing")
  OR
  (adapt* OR reconfig*)
 )
)
```

Search Result URL:
http://www.scopus.com/results/results.url?sort=plf-f&src=s&nlo=&nlr=&nls=&sid=zzIzzTkOWXWN94Hsl87v65C%3a60&sot=a&sdt=a&sl=446&s=ABS%28++%28choreography+OR+%22decentralized+composition%22+OR+%22decentralized+service+composition%22+OR+++++%22distributed+composition%22+OR+%22distributed+service+composition%22+OR+++++%22decentralized+interacting+services%22%29++++AND++++%28++++%28custom*%29+++++OR+++++%28%22self-configurable%22+OR+%22self-configuring%22+OR+%22self-configuration%22+OR++++++%22auto-configurable%22+OR+%22auto-configuring%22+OR+%22auto-configuration%22+OR++++++%22self-healing%22%29+++++OR+++++%28adapt*+OR+reconfig*%29+++%29+++%29&cl=t&offset=1&origin=resultslist&ss=plf-f&ws=r-f&ps=r-f&cs=r-f&cc=5&txGid=zzIzzTkOWXWN94Hsl87v65C%3a8

Applied Filters: None

Search Engine: ISI Web of Knowledge - Web of Science
Date: November 18th, 2010
Number of Results: 45
Number of Distinct Valid Results: 45
Search String:

```
TS=(
 (choreography OR "decentralized composition" OR "decentralized service
composition" OR
  "distributed composition" OR "distributed service composition" OR
  "decentralized interacting services")
 AND
 (
  (custom*)
  OR
  ("self-configurable" OR "self-configuring" OR "self-configuration" OR
   "auto-configurable" OR "auto-configuring" OR "auto-configuration" OR
   "self-healing")
  OR
  (adapt* OR reconfig*)
 )
)
```

Search Result URL: N/A

Applied Filters:
Refined by: [excluding] Subject Areas=(NANOSCIENCE & NANOTECHNOLOGY OR CELL BIOLOGY OR OPTICS OR CHEMISTRY, MULTIDISCIPLINARY OR IMMUNOLOGY)
Timespan=All Years. Databases=SCI-EXPANDED, CPCI-S

Search Engine: CiteSeerX
Date: November 18th, 2010
Number of Results: 19
Number of Distinct Valid Results: 19
Search String:

```
(
 (choreography OR
  "decentralized composition" OR "decentralized service composition" OR
  "distributed composition" OR "distributed service composition" OR
  "decentralized interacting services")
 AND
 (
  (custom*)
  OR
  ("self-configurable" OR "self-configuring" OR "self-configuration" OR
   "auto-configurable" OR "auto-configuring" OR "auto-configuration" OR
   "self-healing")
  OR
  (adapt* OR reconfig*)
 )
)
```

Search Result URL:
http://citeseerx.ist.psu.edu/search?q=abstract:((++(choreography+OR++++%22decentralized+composition%22+OR+%22decentralized+service+composition%22+OR++++%22distributed+composition%22+OR+%22distributed+service+composition%22+OR++++%22decentralized+interacting+services%22)+++AND+++(+++(custom*)++++OR++++(%22self-configurable%22+OR+%22self-configuring%22+OR+%22self-configuration%22+OR+++++%22auto-configurable%22+OR+%22auto-configuring%22+OR+%22auto-configuration%22+OR+++++%22self-healing%22)++++OR++++(adapt*+OR+reconfig*)++)++))&submit=Search&sort=cite&ic=1

Applied Filters: None

References

[Alonso *et al.*, 2004] Alonso, G., Casati, F., Kuno, H., &Machiraju, V. (2004). Web services - Concepts, architectures and applications. Berlin Heidelberg: Springer-Verlag.

[Baeten, 2004] J.C.M. Baeten: A brief history of process algebra, Rapport CSR 04-02, VakgroepInformatica, TechnischeUniversiteit Eindhoven, 2004

[Barker *et al.*, 2009] Barker A, Walton CD, Robertson D. Choreographing Web Services.*IEEE Transactions on Services Computing*. 2009;2(2):152-166.

[Basten and van der Aalst, 2001] T. Basten and W. M. P. van der Aalst. Inheritance of behavior. JLAP, 47(2):47–145, 2001.

[Biolchini *et al.*, 2005] J. Biolchini, P. Gomes, A. Cruz, and G. Orta, "*Systematic Review in Software Engineering",* TECHNICAL REPORT. RT – ES 679 / 05. System Engineering and Computer Science Department COPPE/UFRJ, 2005.

[Busi *et al.*, 2005] N. Busi, R. Gorrieri, C. Guidi, R. Lucchi, and G. Zavattaro. Choreography and Orchestration: A Synergic Approach for System Design. In B. Benatallah, F. Casati, and P. Traverso (Eds.): ICSOC 2005, LNCS 3826, pages 228–240. Springer Verlag, 2005.

[Cambronero *et al.*, 2009] Cambronero, M.E.; Diaz, G.; Martinez, E.; Valero, V.; A comparative study between WSCI, WS-CDL, and OWL-S .

[Cohen, 1960] J. Cohen, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1960) 37–46.

[Crawford *et al.*, 2003] M. Crawford *et al.*ebXML Core Components Technical Specification 2.01. Technical report, UN/CEFACT, November 2003.

[Daniel and Pernici, 2006] Daniel, F.; Pernici, B. Insights into Web Service Orchestration and Choreography, 2006

[Decker and Weske, 2007] G. Decker and M. Weske. Behavioral Consistency for B2B Process Integration.In Proceedings of the 19th International Conference on Advanced Information Systems Engineering (CAiSE), LNCS, Trondheim, Norway, June 2007. Springer Verlag.

[Dyba and Dingsøyr, 2008] T. Dyba, T. Dingsøyr, Empirical studies of agile software development: A systematic review, Inform. Softw. Technol. (2008), doi:10.1016/j.infsof.2008.01.006 7

[Hamming, 1968] Richard Hamming 1968 Turning Award Lecture

[Higgins and Green, 2008] J.P.T. Higgins, S. Green (Eds.), Cochrane Handbook for Systematic Reviews of Interventions, Version 5.0.0 (updated February 2008), The Cochrane Collaboration, 2008. Available from: <www.cochrane-handbook.org>.

[IBM Websphere Process Server] IBM Websphere Process Server https://www-304.ibm.com/support/docview.wss?uid=swg27008938&aid=1]

[Lesser, 2000] Lesser, Victor.Multi-Agent Systems. Encyclopedia of Computer Science, Volume 4th Edition, Ralston, Reilly &Hemmendinger; London: Nature Publishing Group, pp. 1194-1196. 2000.

[Kitchenham and Charters, 2007] Kitchenham, B., & Charters, S. "*Guidelines for performing Systematic Literature reviews in Software Engineering"* version 2.3 EBSE Technical Report.Keele University and University of Durham. 2007.

[Koskimies and Makinen, 1994] K. Koskimies and E. Makinen. Automatic synthesis of state machines from trace diagrams. Software - Practice and Experience, 24(7):643–658, 1994.

[Mahfouz *et al.*, 2008] Mahfouz, A.; Barroca, L.; Laney, R. &Nuseibeh, B. "Customizing Choreography: Deriving Conversations from Organizational Dependencies", presented at Enterprise Distributed Object Computing Conference (EDOC), Munich, Germany, 2008.

[Mellor *et al.*, 2003] Stephen J. Mellor, Anthony N. Clark, Takao Futagami, "Guest Editors' Introduction: Model Driven Development," *IEEE Software*, vol. 20, no. 5, pp. 14-18, Sep./Oct. 2003, doi:10.1109/MS.2003.1231145

[Ortiz *et al.*, 2006] Ortiz, G.a, H. J. b. C. P. b. Reusable web services composition patterns and their extra-functional properties *Computer Systems Science and Engineering,* 2006*, 21*, 30

[Papazoglou *et al.*, 2007] Papazoglou, M., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: State of the art and research challenges. Computer 40(11), 38–45 (2007)

[Peltz,2003] Peltz, Chris; Web Services Orchestration and Choreography.

[Petticrew and Helen, 2005] Petticrew, Mark and Helen Roberts.“*Systematic Reviews in the Social Sciences: A Practical Guide, Blackwell Publishing”*, 2005, ISBN 1405121106.

[Pierce, 1996] Benjamin Pierce: “Foundational Calculi for Programming Languages”, The Computer Science and Engineering Handbook, pp. 2190-2207, CRC Press, ISBN 0-8493-2909-4.

[Ross-Talbot, 2005] S. Ross-Talbot, “Orchestration and Choreography: Standards, Tools and Technologies for Distributed Workflows”

[Schnieders and Puhlmann, 2006] A. Schnieders and F. Puhlmann. Variability Mechanisms in E-Business Process Families. In W. Abramowicz, H. Mayr (Eds.): 9th International Conference on Business Information Systems (BIS 2006), volume P-85 of LNI, pages 583–601, Bonn, Germany, 2006.

[Staples and Niazi, 2007] Staples, M. and Niazi, M. Experiences using systematic review guidelines. Article available online, JSS.

[Selic, 2003] Bran Selic, "The Pragmatics of Model-Driven Development,"*IEEE Software*, vol. 20, no. 5, pp. 19-25, Sep./Oct. 2003, doi:10.1109/MS.2003.1231146

[Srivastava and Koehler, 2003] Srivastava, B., Koehler, J.: Web Service Composition - Current Solutions and Open Problems. Proceedings of ICAPS 2003 (2003)

[Steinmacher *et al.*, 2010] STEINMACHER, I. ; CHAVES, A.P. ; GEROSA, M. A. . Awareness Support in Global Software Development: A Systematic Review Based on the 3C Collaboration Model. In: 16th CRIWG Conference on Collaboration and Technology, 2010, Maastricht. Lecture Notes in Computer Science, 2010. v. 6257. p. 185-201

[Sycara, 1998] Katia Sycara. MultiAgent Systems. AI Magazine 19(2): Summer 1998, 79-92.

[Tosic *et al.*, 2006] Tosic, V.; Pagurek, B.; Esfandiari, B. &Lutfiyya, H. Analytical comparisons of switching of web services and switching of service offerings 2006

[Uchitel *et al.*, 2003] S. Uchitel, J. Kramer, and J. Magee. Synthesis of behavioral models from scenarios. IEEE Trans. Softw. Eng., 29(2):99–115, 2003.

[van der Aalst and Basten, 2002] W. M. P. van der Aalst and T. Basten. Inheritance of workflows: an approach to tackling problems related to change. Theor.Comput. Sci., 270(1-2):125–203, January 2002.

[Xu *et al.*, 2007] Dong-Hong Xu; Yong Qi; Di Hou; Ying Chen; Liang Liu; , "A Formal Model for dynamic Web Services Composition MAS-Based and Simple Security Analysis Using Spi Calculus," Next Generation Web Services Practices, 2007. NWeSP 2007. Third International Conference on , vol., no., pp.69-72, 29-31 Oct. 2007.

[W3C WS Chor. Working Group] W3C’s Web Services Choreography Working Group

[Zhao *et al.*, 2009] Lingjun Zhao; Bin Li; Jun Wu; , "A multi-agent collaborative model for Web Services composition," Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on , vol.1, no., pp.145-148, 8-9 Aug. 2009