# Towards Collaborative, Dynamic & Complex Systems (Short Paper)

Vasilios Andrikopoulos, Santiago Gómez Sáez, Dimka Karastoyanova, Andreas Weiß

Institute of Architecture of Application Systems,
University of Stuttgart, Germany
{vasilios.andrikopoulos, santiago.gomez-saez, dimka.karastoyanova, andreas.weiss}@iaas.uni-stuttgart.de

BibTeX:

```
@inproceedings {INPROC-2013-55,
  author      = {Vasilios Andrikopoulos and Santiago G{\'o}mez
                 S{\'a}ez and Dimka Karastoyanova and Andreas Wei{\ss}},
  booktitle   = {Service-Oriented Computing and Applications (SOCA), 2013 IEEE
                 6th International Conference on},
  title       = {Towards Collaborative, Dynamic and Complex Systems (Short
                 Paper)},
  publisher   = {IEEE},
  institution = {University of Stuttgart, Faculty of Computer Science,
                 Electrical Engineering, and Information Technology, Germany},
  series      = {IEEE Computer Society},
  year        = {2013},
  month       = {Dec},
  pages       = {241-245},
  doi         = {10.1109/SOCA.2013.35}
}
```

# Towards Collaborative, Dynamic & Complex Systems (Short Paper)

Vasilios Andrikopoulos, Santiago Gómez Sáez, Dimka Karastoyanova, Andreas Weiß
Institute of Architecture of Application Systems (IAAS)
University of Stuttgart, Stuttgart, Germany
{vasilios.andrikopoulos, santiago.gomez-saez, dimka.karastoyanova, andreas.weiss}@iaas.uni-stuttgart.de

***Abstract*—Service orientation has significantly facilitated the development of complex distributed systems spanning multiple organizations. However, different application areas approach such systems in domain-specific ways, focusing on particular aspects relevant only for their application types. As a result, we observe a very fragmented landscape of service-oriented systems, which does not enable collaboration across organizations. To address this concern, in this work we introduce the notion of Collaborative, Dynamic and Complex (CDC) systems and position them with respect to existing technologies. In addition, we present how CDC systems are modeled and the steps to provision and execute them. We also contribute an architecture enabling CDC Systems with full life cycle coverage that allows for leveraging service-oriented and Cloud-related technologies.**

***Keywords*—collaborative, dynamic & complex systems; service orchestration & choreography; pervasive computing; service networks; context-awareness**

## I. Introduction

Complex software systems involving multiple, independent partners and software components collaborating in order to achieve one or more goals find predominant application in the current IT landscape. Cases of such systems from different domains are for instance business applications targeting enactment of complex business transactions and service networks, scientific workflows providing one approach for scientific experimenting in eScience, and pervasive systems representing one flavor of ubiquitous computing. Our research work towards building support systems for the development and execution of such applications lets us conclude that while all the above-mentioned application areas concentrate on creating complex systems with very specific features critical for the corresponding domain, there are requirements valid across all domains. Our experience also shows, that synergies between these domains can be exploited and potential benefits realized through reuse of research results and available software systems.

In this respect, in this work we investigate the requirements towards software systems in the above mentioned application areas with the purpose of identifying overlaps and differences. As we are going to show, the overlaps are significant and the differences are mainly due to the special focus on critical aspects in each domain, and not because the solutions are not relevant in the other domains. Based on these findings we introduce the innovative notion of *Collaborative, Dynamic and Complex (CDC)* systems aiming to cover all identified requirements and allowing to apply already existing technologies and software systems.

The contributions of this work aim at enabling three aspects of CDC systems, namely their modeling, provision and execution, and can be summarized by:

- The synthesis of existing technologies and approaches from the service-oriented computing paradigm and beyond, into a new, unified type of Collaborative, Dynamic and Complex (CDC) systems.
- The specification of the architecture for a framework that supports the various aspects of CDC systems.

The remaining paper is structured as follows. Section II looks into different application areas that deal with large, complex, dynamic and collaborative systems in order to highlight their similarities and establish the minimum set of requirements for our work. Section III presents our proposal for CDC systems and positions them with respect to existing approaches. Section IV introduces the architecture of a CDC-supporting framework. Finally, Section V presents related works, and Section VI concludes the paper.

## II. Motivation

In the following, we look into the areas of pervasive systems, service networks and scientific workflows. Our experience in research projects[1] shows, that despite the differences, the available approaches from these areas have many commonalities.

*Pervasive Systems:* Pervasive systems strive towards enabling the paradigm of ubiquitous computing and have been a subject of interdisciplinary research. Advances in pervasive systems have focused on the aspect of context-awareness, i.e. taking into account the context of physical and virtual entities, which is in fact a view of the physical environment, and the influence of the context on the applications the entities are using or participating in [1]. A major requirement in these systems is the ability to adapt their behavior with respect to the context. Another major challenge is the optimization of the distribution of applications based on context data and resource consumption. The distribution of pervasive applications across multiple software system and hardware devices require their integration and coordination towards enabling a collaboration among participating devices and systems. Due to the dynamic characteristics of the environment of pervasive applications, with participants and devices appearing and disappearing constantly, supporting context sharing, adaptation, and scalability

[1] For example SimTech http://www.simtech.uni-stuttgart.de/, S-Cube http://www.s-cube-network.eu/, 4CaaSt http://www.4caast.eu/, ALLOW Ensembles http://www.allow-ensembles.eu/.

are particularly challenging. Since recently, Cloud Platforms in the scope of the Internet of Things and Smart Systems initiatives have been investigated from the point of view of enabling scalability, multi-tenancy and adaptability [2].

*Service Networks:* Service Networks (SNs) [3] are considered a specialized view on business processes, focusing on assisting business experts to evaluate the value of participating in a collaborative business activity. SNs are modeled as a network of business services exchanging offerings. Basically, the perceived composite value of the exchanged offerings with the other services determines the value of participating in the network to one participant. Typical examples of SNs are supply chains. There is a significant gap between the meta-models used by business experts when designing the SNs, and the technological realization that needs to be bridged by means of software engineering techniques like model-driven development and code generation, whereas both top-down and bottom-up approaches are required. In addition, service networks are inherently collaborative activities and therefore imply efforts towards integration of applications across organizations. A SOA-based realization of service networks, as well as a meta-model and graphical notation are presented in [4], [5]. The high-level meta-model has been mapped on choreographies of composite services (i.e. organization-specific business processes) enabling the coordination of the services in a network, and thus addressing another important requirement. Changes in the perceived value of a network to a participant may initiate changes in the individual partners or in the network as a whole, which have to be propagated to their technological realization. Despite some preliminary attempts to support only some types of service networks adaptation [6] this requirement has not yet been thoroughly addressed. Measuring the value of an SN for a participant can only be derived based on monitoring data provided by the execution environment for choreographies, orchestrations and services. Approaches based on business activity monitoring, like [7] and [8] are only first steps towards the necessary technological support.

*Scientific Workflows:* Scientific workflows enable the modeling and execution of scientific experiments and are part of the technology landscape in eScience [9]. A major requirement in this field is first and foremost the user friendliness of the approach, so that scientists do not face a high learning burden when using the experiment modeling tools. The division between the way scientists model an experiment and the meta-models used in the supporting IT systems is significant and there are different approaches towards eliminating it [9]. Both top-down and bottom-up approaches are required to enable the use of existing software and the development of experiments from scratch. The distributed nature of complex scientific experiments requires integration and composition of scientific computing software, which presents an additional challenge. Reusability is hampered by the heterogeneous landscape of applications and the huge effort required for integration. Since scientific discovery is based on exploring physical phenomena, huge amounts of data are collected via numerous types of mobile devices and sensors (e.g. simulations of the distribution of $CO_2$ in the soil, weather forecasts, biological system simulations, manufacturing systems simulations, etc.) which need to be processed. Computations in scientific workflows are typically time-consuming and adaptation during the modeling and execution of scientific workflows is a must [9], [10].

Despite the different focus of the application domains and resulting systems described above, they all exhibit overlapping characteristics that can be leveraged in a unified manner across the various areas. The following section presents our proposal toward this goal.

## III. CDC Systems

We define *Collaborative, Dynamic and Complex* (CDC) systems as distributed systems enabling collaboration among participants across different organizations. Participants of CDC systems are services, representing software systems of different granularity, virtual and physical devices, and individuals. CDC participants join and leave the system at will in order to fulfill their individual goals. CDC systems are capable of adapting with respect to different triggers in the system and/or in their environment. CDC systems consist potentially of a large amount of participants dealing with large amounts of data as part of multiple interactions between them, following one or more coordination protocols. CDC systems have three fundamental aspects: *Modeling*, *Provision* and *Execution*.

With respect to *modeling*, we use choreographies to define the high-level, domain-specific models of CDC systems. *Choreographies* describe the interaction protocol of the involved participants and the participant roles' definitions. In SOA environments, individual participant roles are implemented by service *orchestrations* exposed as services, whereas their service interfaces are compliant with the participant role definitions modeled in the choreography. The *services* composed by the orchestrations are either available in the software landscape of the participating organizations, or are discoverable in global service registries. Utilizing these SOA-based approaches provides a flexible way of composing applications in complex systems and facilitates application integration. To enable *context-awareness*, choreographies and orchestrations, as well as involved services, have to incorporate in their models context information and define its use and reaction to potential changes. Since context information may be part of correlation data of orchestrations belonging to an enacted choreography, a mapping between context and *correlation* mechanisms has to be in place.

*Performance indicators*, like KPIs, utility, value, performance, status information, etc. are an inseparable part of the CDC system models. On the one hand, they are used to define the indicators according to which users will measure and evaluate whether they achieve their goals in a collaboration. On the other hand, this is the information needed to derive the data to be monitored during the execution of the CDC system. Therefore choreographies, orchestrations and services models have to contain elements defining the necessary *monitoring* information. In order to enable the *dynamic features* of CDC systems, constructs accommodating *adaptation* mechanisms in the choreographies and orchestrations have to be incorporated. Available approaches from the fields of workflow adaptation, flexible scientific workflows and pervasive dynamic flows, e.g. [8] or [9] can be applied individually or in combination. Change propagation across all levels of the CDC systems and thus adaptation of choreographies can be identified as a major research challenge.

As identified in Section II, two types of approach in modeling are required: top-down and bottom-up. *Top-down*

CDC system modeling entails starting the development of the system with a choreography representing a realization of a high-level (domain-specific model), like an SN, scientific workflow, or pervasive application. Techniques required to map the choreography into orchestrations and services, like code generation and transformations, are available from software engineering and existing SOA-enabling systems. The *bottom-up* approach involves deriving a meaningful choreography model based on existing orchestrations and/or services. In this case, deriving fault handling, monitoring and adaptation information is based on the corresponding capabilities of the involved services and correctness of the derived choreography.

The *provisioning* aspect of CDC systems entails the provision of the choreography, which in turn requires the deployment of orchestrations onto execution engines, their provision as services, populating the system with the corresponding context and correlation data, and configuring the used monitoring infrastructure with the monitoring requirements from the CDC model. The provision of orchestrations and services are available mechanisms in service composition systems and scientific workflows. Solutions for mapping monitoring requirements to monitoring probes are available in pervasive systems and service-based applications. The provision of a choreography results into adaptive and context-aware orchestrations available as services. A choreography can be initiated multiple times and can be started by any of the participating orchestrations or services allowed to do so by the choreography definition. Any underlying infrastructure should therefore enable sharing of resources across different CDC systems while correlating interactions to tenants and users.

Running a choreography is therefore realized as a distributed *execution* of the collaboration among participating orchestrations and services. Since context-awareness is inherent to the CDC system model, the execution environment has to be able to support this property. Adaptation mechanisms, predefined in the system model (like abstract activities, binding strategies for services, reactions to context change, etc.) and such that are orthogonal to the model (like manual adaptation, forced termination, substituting a service endpoint, etc.) need to be implemented by the execution environment. Furthermore, the CDC systems execution environment must scale with their number of participants and their interactions, as well as the volumes of data exchanged. Monitoring information is necessary in order to enable such scaling.

## IV. CDC Framework Architecture

Figure 1 provides an overview of our proposal for a framework supporting the modeling, provision and execution of CDC systems. Starting from the modeling aspect, a *Choreography Editor* is required to create, visualize and manage the choreography models of the CDC systems. A *Transformer* component can then either generate orchestration templates that the CDC participants are meant to implement (in the top-down approach), or derive possible choreographies from existing orchestrations (in the bottom-up approach). In either case, an *Orchestration Editor* (not necessarily but preferably in the same environment as the Choreography Editor) should be available for orchestration visualization and manipulation. The transformer components also requires as input the service descriptions of the used orchestrations in the bottom-up approach or generates (abstract) service descriptions for derived orchestrations.

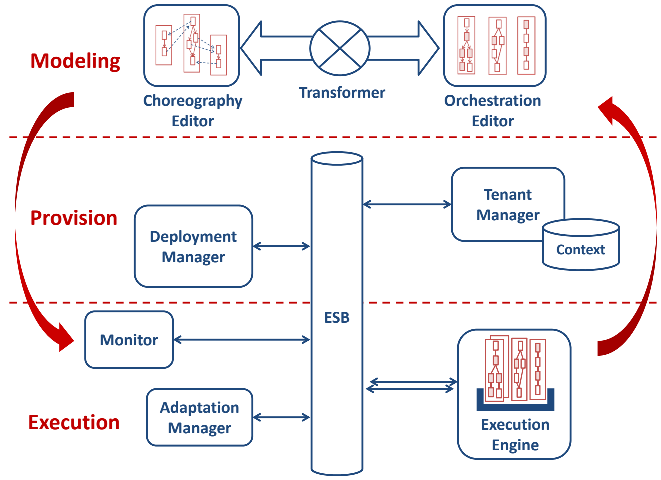


Figure 1. Architectural view of the CDC-supporting framework

Moving to the provisioning aspect, the *Deployment Manager* allows the assignment of the necessary for operation computational resources to the orchestrations involved in the modeled choreographies. Beyond physically deploying the necessary artifacts on an *Execution Engine*, this additionally entails the creation of all service endpoints necessary for accessing the orchestration logic by the system participants. The Deployment Manager also handles the information needed for late and dynamic binding to concrete service endpoints and provides it to the ESB during the execution of orchestrations.

In principle, multiple organizational domains may be using the same instantiation of this framework for different CDC systems. It is therefore necessary to offer *multi-tenancy capabilities* out of the box for all components *in the provision and execution aspect* of the framework. A *Tenant Manager* is responsible for this role, and implements administration and management capabilities for existing and new *tenants* (organizational domains) and their *users* (individuals or subsystems in the same domain). The Tenant Manager is also meant to implement access control to both choreography and orchestration models, and to the computational resources allocated to them during the execution of CDC systems. Furthermore, any collected *context information* relevant for tenants and users representing their environment, e.g. their physical location or the quality of observed data, is stored and accessed through the Tenant Manager.

While the Deployment and Tenant Managers play prominent roles in the provision aspect of CDC systems, they are also heavily involved during CDC system execution, since both of them need to interact with the actual *Execution Engine* that runs the orchestrations. Furthermore, the Execution Engine has to provide fault handling capabilities, both for pre-defined fault and compensation handlers in the orchestration models, and for failures during execution like service failures and unavailability of other components in the framework (e.g. access to the Deployment Manager). The *Adaptation Manager* is responsible for triggering and managing the adaptivity features of CDC systems by providing mechanisms for different types of adaptations across the levels of the systems. It implements and/or coordinates the actions necessary to enable the adaptation constructs from the CDC system model and the ones implemented only on the level of the execution environment. The Adaptation Manager collaborates also with the Deployment Manager when necessary, e.g. for re-binding service endpoints, and with the Execution Engine, e.g. for injecting a new activity and control connectors into an existing orchestration or deploying a new orchestration in case a choreography has been changed. The Adaptation Manager acts on information provided by the *Monitor* component which monitors and analyses the behavior and performance of the executed orchestrations, of the enacted choreographies, and also of the execution components in the framework. The Monitor must be configurable based on the monitoring information required for the CDC system and is responsible for providing to the users of choreographies and orchestrations personalized views of the relevant monitoring information on their devices.

Leveraging the SOA paradigm, all components in the framework relevant to execution (Execution Engine, Adaptation Manager, Monitor, Tenant Manager and Deployment Manager) should be provided as services and communicate through an *Enterprise Service Bus (ESB)* solution to facilitate their integration. Furthermore, each component should be designed and implemented allowing for both types of scalability: *horizontal* (increase/decrease of number of available instances as required) and *vertical* (adjustment of available computational resources for each component) [11].

## V. Related Work

As discussed in [23], the interaction between participants in a choreography can be modeled following the interaction, or interconnection modeling approaches. The former approach models atomic interactions between participants through *interaction activities*, while the latter interconnects the communication activities of each participant of the choreography. The WS-CDL [24] language standard supports the interaction approach. Using the WS-CDL language as the basis, the Savara[2] project aims to provide tooling support for a top-down choreography modeling approach. Interconnection modeling approaches are supported in the CHOReOS Integrated Development and Runtime Environment[3], in the Open Knowledge European project[4], and in BPEL4Chor [25]. The CHOReOS environment supports the choreography specification using BPMN 2.0 collaborations [26], and encompasses choreography adaptation based on service availability and QoS assurance. The Open Knowledge framework employs a multiagent protocol to control the interactions between participants in the choreography. Therefore, participants must be specified and deployed prior to the choreography enactment, and adaptation based on context modifications is not considered. As discussed in the previous sections, BPEL4Chor wraps the choreography specification in a layer atop of WS-BPEL which contains the choreography control flow, its participants description and message links between them, and the mapping support to their concrete communication descriptions (WSDL). BPEL4Chor does not support the explicit specification of rules for context-aware adaptation purposes, but decouples the choreography specification from communication specific details, enabling extensibility for dynamic context-aware choreography adaptation.

Context-aware systems have been widely studied in the scope of Ubiquitous Computing. Baldauf et al. present in [1] a set of context-aware systems, and provide a comparison focusing on the architectural principles of context-aware middleware and framework to ease the development of context-aware applications. The CoWSAMI middleware infrastructure utilizes Web services for managing location context in open ambient intelligence environments [27]. The utilization of an ESB as the central piece for communication support in context-aware systems is discussed in [28], where a Context-aware ESB (CA-ESB) is proposed to discover and orchestrate services based on the users' location and available services in specific regions. Concerning different context views in pervasive environments, Abdulrazak et al. present in [29] the micro and macro context-awareness modeling approaches. The former describes the users' surroundings and aims to provide access to local context data, while the latter aggregates local context data to provide a global perspective of different spaces. Self-configuration operations in micro context-awareness models involve coordination of peers in a decentralized manner, making choreographies suitable for modeling the coordination between peers. Furthermore, in [30] it is demonstrated that a decentralized coordination of entities collaborating in context construction and decision making activities in open intelligence spaces ensures high availability of the system.

Wieland et al. present context-aware workflows as an approach for easing the development of context-aware applications [21]. Thus, they propose Context4BPEL, a WS-BPEL [13] extension for explicitly modeling the influence of context on workflows. However, WS-BPEL [13] supports orchestration of services within a business process, while choreography modeling approaches demand a further semantic support for specifying process interactions from a global view. Further research on workflow flexibility has been conducted by integrating support of human interactions during the execution of scientific workflows in [31]. This approach triggers human interactions for non-automated activities via a framework supporting a multi-protocol communication between a scientific workflow management system and pluggable communication devices. An approach for modeling services and their contracts at different levels of abstraction, and enabling dynamically service binding to new service instances is presented in [32]. Dynamic adaptation of services in this approach is based on replacement of components and new service instantiation. All these approaches are focusing on only one particular aspect of CDC systems.

## VI. Conclusions and Future Work

Our investigation into different application areas like pervasive systems, service networks and scientific workflow systems that have been influenced by service-orientation and Cloud computing identified a series of overlapping characteristics that have not been leveraged across these domains so far. Toward this purpose, in this work we introduced the notion of Collaborative, Dynamic and Complex (CDC) Systems as dynamic distributed systems that allow participants from different organizations to

[2]http://www.jboss.org/savara

[3]CHOReOS: Large Scale Choreographies for the Future Internet: http://www.choreos.eu/

[4]Open Knowledge: http://www.openk.org/

collaborate to fulfill their goals. We discussed three fundamental aspects of CDC systems: modeling, provision and execution, and presented the architecture of a framework that supports these aspects.

Currently, we are working on improving the state of CoDyCo, our prototype implementation of the CDC-supporting framework discussed in Section IV in order to address deficiencies we identified during a case study on a context-aware pervasive application. Our current work aims at enabling multi-tenancy for choreographies and orchestrations and integrating it with our multi-tenant ESB solution. Future work is aimed at finalizing the different aspects of our proposal, like bottom-up modeling and the provisioning aspect of CDC Systems, context-awareness in choreographies and orchestrations, and the realization of a context management system, management dashboard integrating monitoring of KPIs, business transactions and choreographies. In terms of adaptation, available approaches for context-aware, automatic adaptation of orchestrations have to be integrated in CoDyCo. Finally, the scalability features of the CoDyCo components have to be investigated further in the scope of our Cloud computing research.

## ACKNOWLEDGMENTS

The research leading to these results has received funding from the EU's Seventh Framework Programme (FP7/2007-2013) projects 4CaaSt (grant agreement no. 258862) and ALLOW Ensembles (grant agreement no. 600792), and from the German Research Foundation (DFG) within the Cluster of Excellence (EXC310) in Simulation Technology.

## REFERENCES

[1] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.

[2] S. Distefano, G. Merlino, and A. Puliafito, "Enabling the cloud of things," in *Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2012, pp. 858–863.

[3] N. S. Caswell *et al.*, "Estimating value in service systems: A case study of a repair service system," *IBM Systems Journal*, vol. 47, no. 1, pp. 87–100, 2008.

[4] O. Danylevych, D. Karastoyanova, and F. Leymann, "Service Networks Modelling: An SOA & BPM Standpoint," *Journal of Universal Computer Science (J.UCS)*, vol. 16, no. 13, pp. 1668–1693, September 2010.

[5] M. Bitsaki *et al.*, "An Architecture for Managing the Lifecycle of Business Goals for Partners in a Service Network," in *Towards a Service-Based Internet, First European Conference, ServiceWave 2008*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, December 2008, pp. 196–207.

[6] S. Wagner *et al.*, "State Propagation-based Monitoring of Business Transactions," in *Proceedings of the 2012 IEEE International Conference on Service-Oriented Computing and Applications*. IEEE, 2012, pp. 1–8.

[7] S. Guinea *et al.*, "Multi-layered monitoring and adaptation," in *ICSOC*. Springer, 2011, pp. 359–373.

[8] B. Wetzstein *et al.*, "Preventing KPI Violations in Business Processes based on Decision Tree Learning and Proactive Runtime Adaptation," *Journal of Systems Integration*, vol. 3, no. 1, pp. 3–18, Januar 2012.

[9] M. Sonntag and D. Karastoyanova, "Next Generation Interactive Scientific Experimenting Based On The Workflow Technology," in *Proceedings of the 21st IASTED International Conference on Modelling and Simulation (MS 2010), 2010*. ACTA Press, July 2010.

[10] M. Sonntag and D. Karastoyanova, "Ad hoc Iteration and Re-execution of Activities in Workflows," *International Journal On Advances in Software*, vol. 5, no. 1 & 2, pp. 91–109, 2012.

[11] L. Vaquero, L. Rodero-Merino, and R. Buyya, "Dynamically scaling applications in the cloud," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 45–52, 2011.

[12] G. Decker *et al.*, "Modeling Service Choreographies Using BPMN and BPEL4Chor," in *Proceedings of CAiSE*. Springer, 2008, pp. 79–93.

[13] OASIS, "Web services business process execution language version 2.0," April 2007. [Online]. Available: http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html

[14] P. Reimann, "Generating BPEL Processes from a BPEL4Chor Description," Universität Stuttgart, Institut für Architektur von Anwendungssystemen, Studienarbeit 2100, 2007.

[15] M. Sonntag, M. Hahn, and D. Karastoyanova, "Mayflower - Explorative Modeling of Scientific Workflows with BPEL," in *Proceedings of the Demo Track of the 10th International Conference on Business Process Management (BPM 2012)*. CEUR Workshop Proceedings, 2012, pp. 1–5.

[16] D. Schumm *et al.*, "Fragmento: Advanced Process Fragment Library," in *Proceedings of the 19th International Conference on Information Systems Development (ISD'10), Prague, Czech Republic, August 25 - 27, 2010*. Springer, 2010, pp. 659–670.

[17] A. Weiß *et al.*, "Modeling Choreographies using the BPEL4Chor Designer: an Evaluation Based on Case Studies," IAAS, University of Stuttgart, Tech. Rep. 2013/03, 2013.

[18] R. Khalaf, D. Karastoyanova, and F. Leymann, "Pluggable framework for enabling the execution of extended bpel behavior," in *Proceedings of the 3rd International Workshop on Engineering Service-Oriented Application (WESOA'2007)*. Springer, 2007, pp. 376–387.

[19] S. Strauch *et al.*, "$\text{ESB}^{\text{MT}}$: Enabling Multi-Tenancy in Enterprise Service Buses," in *Proceedings of CloudCom'12*. IEEE Computer Society, December 2012, pp. 456–463.

[20] S. Strauch *et al.*, "Enabling Tenant-Aware Administration and Management for JBI Environments," in *Proceedings of SOCA'12*. IEEE Computer Society, December 2012, pp. 206–213.

[21] M. Wieland *et al.*, "Towards context-aware workflows," in *CAISE07 Proceedings*. Springer, 2007, pp. 11–15.

[22] M. Knappmeyer *et al.*, "Contextml: a light-weight context representation and context management schema," in *2010 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*. IEEE, 2010, pp. 367–372.

[23] A. Barker, C. D. Walton, and D. Robertson, "Choreographing web services," *IEEE Transactions on Services Computing*, vol. 2, no. 2, pp. 152–166, 2009.

[24] N. Kavantzas *et al.*, "Web services choreography description language version 1.0," 2005.

[25] G. Decker *et al.*, "BPEL4Chor: Extending BPEL for Modeling Choreographies," in *Proceedings of the IEEE 2007 International Conference on Web Services (ICWS 2007)*. Salt Lake City: IEEE Computer Society, 2007, pp. 296–303.

[26] CHOReOS Consortium, "CHOReOS Dynamic Development Model Definition (D2.1)," 2011.

[27] D. Athanasopoulos *et al.*, "Cowsami: Interface-aware context gathering in ambient intelligence environments," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 360–389, 2008.

[28] J. Chanda *et al.*, "CA-ESB: Context Aware Enterprise Service Bus," *International Journal of Computer Applications*, vol. 30, no. 3, pp. 1–8, September 2011.

[29] B. Abdulrazak *et al.*, "Macro and micro context-awareness for autonomic pervasive computing," in *iiWAS2010 Proceedings*. ACM, 2010, pp. 427–434.

[30] P. Roy, B. Abdulrazak, and Y. Belala, "Approaching context-awareness for open intelligent space," in *Proceedings of MoMM2008*. ACM, 2008, pp. 422–426.

[31] D. Karastoyanova *et al.*, "Service-based Integration of Human Users in Workflow-driven Scientific Experiments," in *Proceedings of the 8th IEEE International Conference on eScience (eScience 2012)*. IEEE Computer Society Press, 2012, pp. 1–8.

[32] H. Giese and B. Becker, "Modeling and verifying dynamic evolving service-oriented architectures," Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam, Tech. Rep. 75, 2013.