Pushing dynamic and ubiquitous interaction between services **L**everaged in the Future Internet by **A**ppl**Y**ing complex event processing

Project number:	258659		
Instrument:	Collaborative Project		
Thematic Priority:	ICT-2009.1.2: Internet of Services, Software and Virtualisation		
Start Date:	01/10/2010	**Duration:**	36 months

D1.1 – State of the Art

WP 1:	PLAY conceptual framework

Due date:	M4
Submission Date:	day/month/year
Organization Responsible for the Deliverable:	ICCS
Version:	V 3.0
Status:	FINAL
Abstract:	

Author(s):	G. Mentzas	ICCS
	D. Apostolou	ICCS
	Y. Verginadis	ICCS
	V. Tsalikis	ICCS
	N. Stojanovic	FZI
	R. Stühmer	FZI
	Jean-Pierre Lorré	EBM
	Christophe Hamerling	EBM
	F. Baude	INRIA
	F. Bongiovanni	INRIA

	Anne-Marie Barthe Philippe Gibert	ARMINES FT
Reviewer(s):	Frederick Benaben Roland Stühmer	ARMINES FZI

Project co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission)	

* - put **X** in proper field

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.2	01/12/2010	First version	FZI
1.1	06/12/2010	Added draft section 5	ICCS
1.2	07/12/2010	Added sections 3.2	EBM
1.4	10/12/2010	Added first draft versions for all sections	FZI, ICCS
1.5	20/12/2010	Pre-final version of section 5	ICCS
1.7	21/12/2010	Additions and corrections in all sections	ICCS, INRIA
1.8	12/01/2011	Added related projects section	ICCS
1.9	14/01/2011	Re-Arranged the whole 3.2	INRIA
1.10	17/01/2011	Adding details to related projects, privacy in pub/sub	FZI, INRIA
2.0	19/01/2011	Final adjustments in sections 5,6,8	ICCS, FZI, INRIA, FT, ARMINES
2.0.3	27/01/2011	Internal Review Comments	ARMINES, FZI
3.0	31/01/2011	Final Deliverable	ICCS, FZI, EBM, INRIA, FT

Table of Contents

1 Executive Summary

The PLAY project, as proposed in the Description of Work, aims to define an elastic and reliable architecture for dynamic complex event driven interaction in large highly distributed and heterogeneous federated service systems. This platform (see Figure 1) will allow building very large scale Event Driven Architecture able to settle complex business at the Internet level.

The PLAY conceptual architecture is organised into 2 layers. The Federated middleware layer which will provide highly distributed publish/subscribe mechanism allowing storing and retrieving events coming from heterogeneous services infrastructures organised into domains as well as the corresponding semantic. The second one is the Complex Event Processing layer which will allow deducing business events from low level events based on complex event patterns. These events will trigger the recommendation of service adaptations that will ensure system's quality and proactive responsiveness in problematic situations. The Governance platform will be in charge of the management of design time artefacts such as business events, event patterns, user profiles and rights, SLA contracts and service policies, while the monitoring platform will provide such facilities as the ubiquitous monitoring and management of events through a graphical user interface available over the Web.

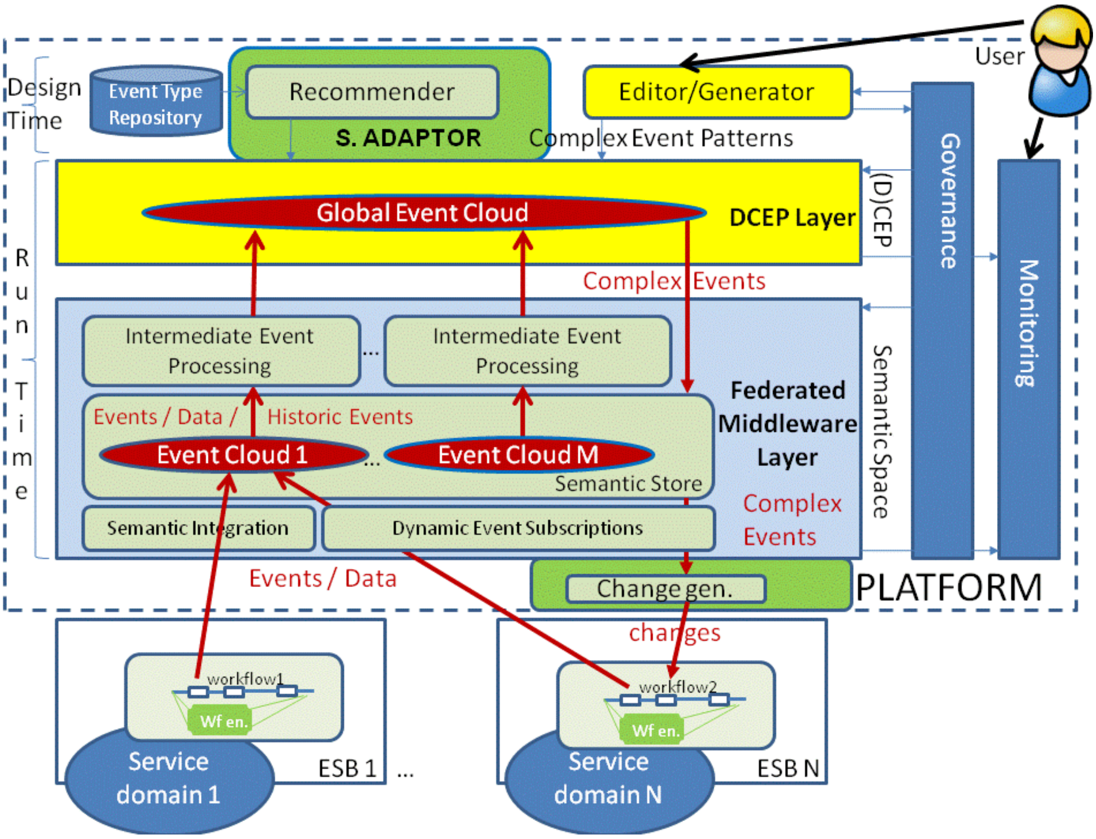


Figure 1: PLAY conceptual architecture

The present document provides the state of the art about all the above issues that will be considered in PLAY. In particular, in section 2, we present the purpose of the PLAY project along with the way that different techniques (discussed in latter sections) can be used for accomplishing its objectives. In section 3, we discuss middleware approaches for event-based and federated service-oriented architectures (SOA), while in section 4 we present complex event processing approaches used so far. Then in section 5, we indicate the state of the art on interactions between people, services and events that encapsulate principles and tools for triggers and event-based application of adaptations in service-based systems. In section 6, we present the overview of relevant research projects, mostly funded by the European Commission, while in section 7 we refer to the standards that are relevant to PLAY project. We conclude this document with the positioning of PLAY in the current state of the art.

2 Introduction

2.1 Purpose

Main goals of this deliverable are

- to give a systematic overview of the current research in the areas relevant for the project and
- to create awareness about:
  a) concrete contributions we can provide in selected research areas (novelty)
  b) importance of these contributions for the corresponding research area (research impact)
  c) applicability of these contributions in various domains (application potential)
  d) possibilities to collaborate with other projects

in order to define:

1. a baseline for own research
2. a mechanism for continual monitoring of the related work

2.2 Organization

The method used for achieving above mentioned goals is as follows:

1. for each topic we:
- analysed the most relevant work
- defined the role of that topic in PLAY and
- discussed how our research can advance state of the art and
2. we defined a list of relevant projects (research and application-oriented) and analysed shortly the relation to our goals

2.3 List of Acronyms

Acronym	Definition
AO4BPEL	Aspect-Oriented extension for BPEL
AOD	Aspect-Oriented Design
AOM	Aspect-Oriented Modelling
AOP	Aspect-Oriented Programming
BPEL	Business Process Execution Language
CEP	Complex Event Processing
CF	Collaborative Filtering
DAS	Dynamic Adaptive Systems
(D)CEP	Distributed Complex Event Processing
EDA	Event Driven Architecture
IaaS	Infrastructure as a Service

KPI	Key Process indicator
MDE	Model Driven Engineering
PaaS	Platform as a Service
SaaS	Software as a Service
SBA	Service-Based Application
SLA	Service Level Agreement
SOA	Service-Oriented Architecture
SON	Structured Overlay Network
WS	Web Service
WSDL	Web Service Description Language
XML	eXtensible Markup Language

2.4 Glossary

Distributed Hash Table A Distributed Hash Table is a class of distributed system which provides the same services as a hash table namely push and get methods for (key, value) pairs. Therefore, DHTs only directly support exact matching search.

Event Cloud An event cloud is an abstraction of a distributed publish/subscribe architecture based on the cloud computing notion in order to dynamically adapt the load due to the unpredictable quantity of events to manage. The event cloud adapts itself to the load and offers several services as monitoring, fault-tolerance, event categorization, etc.

Latency Latency is a measure of time delay experienced in a system where computer nodes exchange information through communication channels. The definition of latency strongly depends on the type of system and the time being measured. In our context, latency has to do with the time delay of the events that are disseminated in the overlay network.

Notifications In pub/sub systems, notifications can signify various types of events depending on the perspective. From a publisher perspective, advertisements are a type of notification used to describe the kind of notfication the publishers are willing to send. From a subscriber perspective, a notification is an event that matches the subscription(s) of consumers.

P2P overlay a Peer-to-Peer Overlay Network is a network built on top of the physical network topology. This means that peers are connected by virtual or logical links. In this Overlay, each peer or machine have the same role, which differs from the client/server model where the communication is usually from and to a central server.

Peer-to-Peer The term "Peer-to-Peer" usually refers to a form of collaboration between computers where the central directory has been replaced with a dissemination process whereby each computer connects to random members in a peer-to-peer network and queries its neighbours who act similarly until a query is resolved. This graph of computers forms an overlay network that is an application-level network on top of the Internet transport with its own topology and routing.

Publish/Subscribe Publish/Subscribe is a messaging paradigm where publishers (i.e. entities which produces information or events) are loosely coupled to the subscribers (i.e. entities which receive some events). Subscribers express their interest in some kind of events and only receive events that are of interest, without the knowledge of who are the publishers. The decoupling of publishers and subscribers allows better scalability than usual client-server paradigm.

Resource Description Framework RDF is a data model for representing objects and relationships between them. A RDF data is represented as a triple (subject, predicate, object) where the subject denotes the resource to describe, the predicate denotes a property or a characteristic of the subject and the object presents the value of the property. A triple is also known as a RDF statement.

Semantic Space A Semantic Space is a distributed infrastructure, based on a structured Peer-to-Peer overlay network, for storing and retrieving semantically correlated RDF data in large scale settings.

Structured Overlay Network Structured Overlay Networks are P2P Overlays where peers are organized following a given topology as opposed to Unstructured Overlay Networks where peers have no organization. In Structured Overlay Network, peers can take advantage of the mathematical properties associated to the topology for efficient routing algorithms in terms of time and space.

Subscription, topic-based A topic-based subscription is a method of expressing interest for the kinds of event which an actor would like to receive. Event senders categorise events by topic. Receivers can choose from these topics and will receive only the corresponding events.

Subscription, content-based A content-based subscription is a method of expressing interest for the kinds of event which an actor would like to receive. An event receiver selects the desired events by providing a Boolean predicate in its subscription. The predicate is evaluated against the payload (content) of each event. Only in case of success is the event delivered to the receiver.

Throughput Throughput in the federated middleware context has to do with the number of events that a node can process per unit of time (events/sec).

3 State of the Art in Middleware for Event-based and Federated SOA

From the middleware point of view, PLAY aims at bringing a large-scale infrastructure that ensures the storage and the dissemination of simple and complex events from various sources to multiple consumers. Scalability in this context is twofold: (i) from a node perspective, the infrastructure should scale at the Internet level in a controlled way without incurring too much state overhead on the nodes participating in the computation and (ii) from an event perspective, the middleware should be able to manage potentially a large quantity of events per unit of time. P2P technologies have proved to a valuable substrate for building large-scale distributed applications (Rodrigues & Druschel, 2010) and consequently will be used in PLAY.

The middleware has also the intent of supporting the establishment of (D)CEP (see section 4) by relying upon a Cloud computing, on-demand approach. The Cloud promises to deliver data center hardware and software to end-users with a certain level of quality. The goal is to deploy the whole PLAY infrastructure on such an Internet scale while ensuring some level of robustness with regards to the events it manages. Because of the best-effort nature of the Internet, large-scale applications should include adequate protocols for ensuring the event availability, especially for their dissemination between the different parts of the cloud.

The last decades of distributed and parallel computing research have spawned clusters, grids, P2P, multicore processing and finally cloud computing. The combination of the P2P communication model with the aforementioned technologies represents an ideal infrastructure since they complement each other perfectly. Decentralization and fault-tolerance are ensured by P2P technologies while performance in terms of latency and throughput are the number one priority of parallelism and scientific high performance processing (HPC), and most recently of any high-demanding computing area. Cloud computing on the other hand, adds economy of scale because of just-fitted on-demand use of resources. This gives even more scalability, elasticity and location agnosticism compared to traditional large-scale data storage and computing offered by data and computing grids.

3.1 Background in P2P and Grid/Cloud Computing

3.1.1 P2P computing

The P2P communication model has drawn much attention and has emerged as a powerful architecture to build large scale distributed applications. P2P architectures are classified into three main categories: *unstructured*, *hybrid* and *structured* P2P Overlays.

In *unstructured* P2P overlay networks, there is no constraint on the P2P architecture as the overlay is built by establishing random links among nodes (*Gnutella RFC*, 2003). Despite their simplicity, they suffer from several issues: limited scalability, longer search time, higher maintenance costs, etc.

A *hybrid* P2P overlay is an architecture in which some parts are built using random links and some others, generally the core of the overlay, are constructed using a specific topology (Yang & Garcia-Molina, 2003).

Structured Overlay Networks (SONs) emerged to alleviate inherent problems of unstructured overlays. In these systems, peers are organized in a well-defined geometric topology (e.g., ring, torus, etc.) and, compared to unstructured networks, they exhibit stronger guarantees in terms of search time and nodes' maintenance (Castro et al., 2005). By providing a *Distributed Hash Table (DHT)* abstraction, by the mean of *Put* and *Get* methods, they offer simple mechanisms to store and fetch data easily in a distributed environment. However, even if this abstraction is practical, it has its limits when it comes to *efficiently* supporting complex types of queries. The first wave of DHTs (Stoica et al., 2001; Ratnasamy et al., 2001) focused on scalability, routing performance and peer churn resilience. The main issue of these DHTs is that they only support *exact queries*, i.e., querying for data items matching a given key (*lookup(key)*). To overcome such limitation, a second wave of DHTs led to a set of P2P approaches such as (Aberer et al., 2003; Cai et al., 2003) having the capabilities to manage more complex queries such as range queries (Aberer et al., 2003) or multi-attribute queries (Cai et al., 2003).

Motivated by the realization of the Semantic Web vision at the Internet level, solutions combining Semantic Web technologies and the P2P communication model arose quite naturally, and, as stated in (Staab & Stuckenschmidt, 2006), both address the same need but at different levels. The Semantic Web allows users to model, manipulate and query knowledge, represented using the Ressource Description Framework (RDF) data model (Resource Description Framework 1999)), at a conceptual level with the intent to exchange it. P2P technologies, on the other hand, enable users to share information using the decentralized organization principle. P2P systems have been recognized as a key communication model to build scalable platforms for distributed applications such as file sharing (e.g., Gnutella (*Gnutella RFC*, 2003)) or distributed computing (e.g.,SETI@home (*SETI@home*, n.d.)).

Leveraging orthogonal ideas, such as *federation*, could provide further control on the overall scalability and autonomy of the system by allowing different entities to keep the control on their own data while still collaborating. Garces *et al.* in (Larces-Erice & Urvoy-Keller, 2003) introduced a two-tier hierarchical structure using Chord at the top level and another DHT at the lower level. This structure, mimicking *Autonomous Systems* (ASes) found on the Internet, is used to connect heterogeneous groups (at the lower level) together. This is done by having a subset of peers, within each group, acting as gateways at the top level. This hierarchical structure inspired the work proposed by Baude *et al.* (Baude et al., 2010) to create *semantic spaces*. A semantic space groups together semantically related RDF triples by storing them in a three-dimensional CAN overlay network. These spaces are referenced at the top level by Chord nodes. This structure has several benefits in terms of QoS management since a semantic space is administered by a single organizational group which can react faster in case of problems for instance.

This semantic space will constitute the basis of the Event Cloud, see below section 3.2.

3.1.2 Cluster, Grid and Cloud Computing Mechanisms

The PLAY middleware is to be hosted at large-scale, thus this section briefly presents what sorts of resource combinations could be set up to this aim. As the PLAY middleware aims to be elastic so to scale at will, we focus on solutions that are able to encompass more than one single resources provider and consequently we put the focus on multi domain, multi provider platforms.

Combining multiple platforms into a multi-domain platform is not an easy task (Jha et al., 2009). Application and/or middleware developers must configure the underlying environment and respect usage policies of each particular resource provider. Next subsections discuss some multi-domain platform combinations. For each combination, we present the motivation, main usages and requirements for a smooth transition from single-domain to the presented multi-domain platform configuration.

3.1.2.1 Mixing clusters and Grid(s)

Mixing resources from private clusters and Grids or multiple Grids has been one alternative to gather more resource performance without additional costs. In order to allow a smooth usage of clusters and Grids together, a set of issues like resources reservation, access and communication must be handled. Depending on the application requirements, network topology and load balancing must also be taken into account. Clusters and Grids present different kinds of resource managers. To allow an easy transition from cluster execution to Grid and mixed cluster/Grid execution, resource management must be handled externally to applications, in a transparent way. Depending on usage policies presented by each resource provider, nodes of private clusters and grids might not be able to communicate directly with external nodes. In order to execute applications that require inter-domain communication, the access and communication between resources must be configured. At the application side, issues like network topology and load balancing are also important for the efficient usage of multi-domain resources. Multi-domain environments generally present heterogeneous machines and networks. Network and computation-intensive applications are more susceptible to performance degradation in face of heterogeneity of resources and, therefore, they must carefully handle these characteristics. Some previous works have already shown the combination of clusters and Grids to be efficient for the execution of distributed and parallel applications (Georgakopoulos & Margaritis, 2008; Goldchleger et al., 2004), specially for applications that require simple communication patterns.

3.1.2.2 Mixing Clouds with clusters and/or Grids

Already owned private resources provide processing power at lower costs i.e. there is no need to additionally pay for external computing resources from a Cloud. This processing power, however, is limited by the number of available machines. Indeed, for cost and management reasons, the provisioning of resources in companies and research institutes is often done based on average usage. Renting resources from Cloud providers allows handling spikes in performance requirement on demand, without the need for buying and managing more computational resources. Besides, by balancing the computational load between in-house and Cloud resources, a trade-off between price and performance can more easily be reached.

We separate mixed Cloud/Grids usage in two approaches, according to the motivation that leads to the usage of a combined cluster/Grid/Cloud platform:

- *Cloud Bursting:* when a sudden increase in computation happens, it is possible to offload some of the computation to a Cloud. This is often referred to as *Cloud Bursting*. Motivation for *Cloud Bursting* includes not only sudden increases of load, but also the need for more processing power, to handle a bigger dataset or to handle failures or shortage in private resources offer, for instance.
- *Cloud Seeding:* in some cases, a distributed application, composed of dependent tasks, can perform a vast majority of its tasks on a Cloud, while running some of them on another infrastructure. This is particularly useful when applications require resources that are not available within the Cloud: specialized hardware (e.g. GPU specialized processors, sensors or other hardware-based data sources), a software protected by a license, private data or algorithms which should not get out to a possibly untrusted environment as a public Cloud.

Cloud Seeding aims at providing a solution to address such problems by increasing the capabilities of a Cloud with specific external resources.

Mixing Clouds with clusters and/or Grids present the same issues of cluster/Grid hybrid environments, i.e. resources allocation, access and communication, network topology and load balancing. In addition, economic concerns also become important, as users must choose the most adequate combination of resources considering application requirements, the available budget and resource offerings and characteristics.

3.1.2.3 Multi-cloud

Usage of Cloud Computing (Armbrust et al., 2010) is growing fast and this reflects in the offerings of Cloud platforms. Each Cloud provider presents different characteristics, different solutions and pricing plans. In (Buyya, 2009) and (Keith Jeffery, n.d.), authors advocate the emergence of Cloud marketplaces for companies that want to buy and sell hosted server capacity online. Indeed, startups like Zimory (*Zimory Website*, n.d.) and GoGrid (*GoGrid Website*, n.d.) are already pushing the creation of such marketplaces. The evolution of Cloud computing shows that it is likely that, in a near future, the so-called Cloud will be, in fact a multi-cloud platform composed by a mixture of private and public Clouds to form an adaptive environment. The mixture of private and public Clouds is motivated by the existence of in-house resources and the need to meet a trade-off between costs and performance. The Cloud will also have to be adaptive to cope with changes in resources offer and users requirements. Commercial solutions like RightScale (*RightScale Website*, n.d.) and FlexiScale (*FlexiScale Website*, n.d.) and research initiatives like Eucalyptus (*Eucalyptus Systems Website*, 2010) work on the integration of Cloud providers to build multi-Cloud platforms. Issues related to multi-cloud usage depend on characteristics offered by each Cloud provider. Public IaaS providers (like Amazon EC2) offer a more open environment that can be configured according with user needs, including firewall configuration, public IP addresses and so on. Private Clouds (like those managed by OpenNebula or Eucalypus), however, depend on the usage policy of the administrative domain that manages the Cloud. In spite of these characteristics, issues related to heterogeneous network and resources are present and the economic aspect is a central concern.

3.2 Middleware for events storage and brokering: Event Cloud

In the context of PLAY, the Semantic Space will be used as a substrate for building our next generation of middleware: the Event Cloud. An Event Cloud has to efficiently manage high loads of simple events coming from the deployed SOA applications and complex events resulting from the (D)CEP, which could be represented by multiple linked RDF triples. Management of events encompass both their storage (short or even longer term) and an adequate delivery to interested subscribers (thus acting as a notification broker). One of the key challenges behind the event cloud will be the usage of adequate indexing and storage mechanisms which are in adequacy with the temporal nature of events while ensuring at the same time a proper level of reliability.

3.2.1 Synchronous and asynchronous (publish/subscribe) data retrieval

Pure SONs offer a simple "one-time query" mechanism, in which you query for something and, if it does exist in the overlay, you get the value back with some guarantees. In the context of highly dynamic systems in which there is a lot of volatile events decoupled in time, we have to go a bit further and add a support for long-standing queries. The Publish/Subscribe paradigm (see (Eugster et al., 2003) for a more detailed survey) introduces query decoupling in space, time and synchronization between participants.

3.2.1.1 Overview of the Publish /Subscribe paradigm

In a publish/subscribe system, subscribers, also called consumers, can express their interests in an event or a pattern of events, and be notified of any generated event by the publishers (producers) that matches those interests. The events are propagated asynchronously to all subscribers. Therefore, the overall system is responsible for matching the events to the subscriptions and for the delivery of those relevant events from the publishers to the subscribers which are distributed across a wide area network via Notifications.

In ubiquitous computing systems, publish/subscribe communication paradigm is usually used to exchange information between applications, services and devices, for its ability to decouple communication participants. The publish/subscribe communication paradigm is frequently chosen over the request/response one to exchange information, mainly for its asynchronous nature and the loose coupling of communication participants (Rodriguez-Dominguez et al. 2010; Cugola and Jacobsen 2002). Anonymity and immediate communication that is supported in publish/subscribe systems (Eugster et al. 2003) constitute another advantage. The aim of these systems is to disseminate published information to a set of subscribers. Each subscriber receives only the information that matches their subscription. In dynamic and highly distributed environments, there are many information providers that if they were indiscriminately and constantly pushing information, they could flood any system. So, not all of this data is to be pushed to the consumers but only the relevant to the users'/services' preferences.

The consumers express their interests by subscribing to events of a given type. When a new event is published, the system compares it against all existing subscriptions and notifies interested parties. In the state of the art in publish/subscribe systems both operations are done explicitly (Berkovsky and Eytani 2005). Users define, at design time, the events they publish and manually inform the system about the events they are interested in. Publish/subscribe systems act as natural mediators between information providers and information consumers in distributed environments (Aguilera et al. 1999). Information providers publish information in a form of events, while information consumers subscribe to a particular category of events. The system is responsible to check the event against all current subscriptions and deliver it to the consumers, whose subscriptions match the event category.

Overall, the key components of a publish/subscribe system can be summarized into the following concepts:

- **Subscriptions**: A subscription describes a set of notifications (i.e., events) a consumer is interested in. The goal behind the subscription process is that subscribers will receive notifications matching their interests from other peers in the network. Subscriptions are basically *filters*, which can range from simple Boolean-valued functions to the use of a complex query language. The expressiveness of the subscriptions in terms of filtering capabilities depends directly on the data model and the filter model used.
- **Notifications**: In publish/subscribe systems, notifications can signify various types of events depending on the perspective. From a publisher perspective, *advertisements* are a special type of notification used to describe the kind of notification the publishers are willing to send. From a subscriber perspective, a notification is an event that matches the subscription(s) of consumers[1] . An event notification service is the mediator which is responsible for conveying notifications to subscribers. Several peers within the network can actively or passively participate in the dissemination of those notifications.

Publish/subscribe systems have several ways for identifying notifications that can be based either on a *topic* or the *content*. In the *topic-based* model, publishers annotate every event they generate with a string denoting a distinct topic. Generally, a topic is expressed as a rooted path in a tree of subjects. For instance, an online research literature database application (such as IEEE Xplore) could publish notifications of newly available articles from the *Semantic Web* research area under the subject "/Articles/Computer Science/Proceedings/Web/Semantic Web". This kind of topic will then be used by subscribers which will, upon subscription's generation, explicitly specify the topic they are interested in and for which they will receive every related notifications. The topic-based model is at the core of several systems such as Scribe (Castro et al., 2002), Sub-to-Sub (Voulgaris et al., 2006). A main limitation of this model lies in the fact that a subscriber could be interested only in a subset of events related to a given topic instead of all events. This comes from the tree-based classification which severely constrains the expressiveness of the model as it restricts notifications to be organized, using a single path in the tree. A tree-based topic hierarchy inhibits the usage of multiple super-topics for instance, even if some inner re-organizations are possible, this classification mechanism remains too rigid. On the other side, a *content-based* model is much more expressive since it allows the evaluation of filters on the whole content of the notifications. In other words, it is the data model and the applied predicates that exclusively determine the expressiveness of the filters. Subscribers express their interests by specifying predicates over the content of notifications they want to receive. These constraints can be more or less complex depending on the query types and operators that are offered by the system. Available query predicates range from simple comparisons, regular expressions, to conjunctions, disjunctions, and XPath expressions on XML.

So to sum up, first, subscribers and publishers do not need to participate in the relation at the same time. Secondly, senders and receivers are not required to have a prior knowledge of each other and can even be located in separate domains. Finally, they are not blocked when generating events and subscribers can get the notifications in an asynchronous manner.

3.2.1.2 Expected non functional properties associated to the Publish /Subscribe paradigm

The Event Cloud PLAY will define has to act as the broker of events: one of its key functionalities is to convey events to interested subscribers, given their subscriptions; another one is to store those events for future delivery, logging or traceability purposes, etc. Brokering and storage of events can be seen as services that, as usual, should feature a given level of quality (QoS). QoS encompasses many aspects, including performance, scalability, reliability, security, privacy, etc. Some of these QoS features are further addressed below. Notice that PLAY Event Cloud design and implementation will focus on scalability and reliability, mainly because addressing all non functional properties, and in particular security, would require much more expertise and efforts than those actually committed within the consortium.

---

[1] In fully decentralized pub/sub systems, every node can perform the matching process.

3.2.1.2.1 Performance and robustness

One key aspect of the Event Cloud on which we will focus our attention is *throughput* with respect with the number of events that should be processed per time unit. Recent work such as (Guerraoui et al., 2010) introduces a protocol for achieving throughput-optimal broadcast primitives that could for instance be adapted to the Event Cloud DHT topology that will be selected. Designing a distributed application, especially a wide-area DHT, for high throughput and low latency is a challenge (Dabek et al., 2004). The main challenge would be to achieve the right balance between scalability, performance (in terms of latency and throughput) and robustness in presence of faults. Fault tolerance in our context means to ensure that stored data is not lost in case of storage nodes possible failure and churn, and that, any notification that should be delivered is eventually delivered. To this aim, fault-tolerance usually relies upon replication-based techniques, but which can be detrimental to performance. In the context of PLAY, we aim to propose replication-based mechanisms targeting wide-area networks with the goal to decrease as much as possible message exchanges on high-latency links, in order to improve the overall performance of the replicated system.

Deploying a large application across independent clouds potentially brings new challenges from fault-tolerance and performances points of view, as underlined in (Vukoli, 2010). Indeed, these cloud platforms being heterogeneous, they naturally offer different quality levels regarding the services they offer, and on which event clouds will be built, such as virtualized computing and storage node basic performances, physical nodes sharing degree, intra and inter clouds networking performances, end-user data isolation and protection, etc. As a matter of fact, when an application spans across multiple independent Event Clouds, ensuring a good application level QoS is going to be critical.

3.2.1.2.2 Security and privacy

Event-based systems give the potential for active information sharing. This communication style has been adopted enthusiastically for financial processing, healthcare applications and sensor-based systems. However, some data are sensitive, and their visibility must be controlled carefully for personal and legal reasons. To meet the requirements, event-based systems have to cope with crucial issues like information confidentiality (to perform content-based routing without revealing the content of the infrastructure), subscription privacy (subscribers specify filters without revealing their interest to the infrastructure) and publisher confidentiality (publishers have to be sure that only the intended subscribers get the data).

In this section we provide a brief overview of previous work in the area of security in publish/subscribe systems. Preliminary work on security issues under publish/subscribe semantics can be found in (Wang et al. 2002). The authors analyze the security issues and requirements that arise in content-based publish subscribe systems. They mainly identify classical security problems (like authentication, integrity or confidentiality) and adapt them to the content-based publish/subscribe case. However, they do not provide concrete or specific solutions to these new problems. In the work by Miklos (Miklós 2002), upper bound filters on advertisements and subscriptions in Siena are proposed but the confidentiality of event publications within the publish/subscribe system is not guaranteed. Some other works like the Narada Brokering project (Pallickara et al. 2003, Yan et al. 2003) include a distributed security framework. Within the Narada Brokering project, the security framework uses access control lists to control event publishers and subscribers, limiting the scalability. Cryptographic keys for encrypting publications are centrally managed by a Key Management Center. An event publisher can choose to use a central topic key from the KMC or the public keys of all event subscribers for encryption, which contradicts decoupled publish/subscribe semantics. Access control can only be provided at the granularity of whole events, and event brokers are implicitly trusted, rather than using different trust levels as proposed in (Belokosztolszki et al. 2003).

More recently, two interesting works concerning confidentiality in content-based publish/subscribe systems have been published. First, in (Raiciu & Rosenblum. 2006), the authors focus on notification and subscription confidentiality only. They define the confidentiality issues in a formal model and propose then few solutions depending on the subscription and notification format. However, the proposed matching techniques are quite costly, since even the basic one featuring equality tests only is six times slower than the protocol without confidentiality. Moreover, they assume that publishers and subscribers share a secret, which reduces the decoupling of content-based publish subscribe. Furthermore, in their attacker model, only the brokers are honest-but-curious, the publishers and subscribers are assumed to be trustworthy. This assumption is very strong because the group of publishers and subscribers may be very large. Such a scheme does not protect subscribers' privacy against other curious subscribers for example, let alone against malicious subscribers. Second, in (Srivatsa & Liu 2007), authors propose a specific key management scheme and then a probabilistic multi-path event routing to prevent frequency inferring attacks. In their threat model all nodes (publishers, subscribers and brokers) are assumed to be honest-but-curious. The main weakness of the scheme is the requirement for a Key Distribution Center which is a centralized authority that is trusted not to be curious and decipher all the communication messages. Concerning content-based event routing, this scheme considers that events have some routable attributes which are tokenized in order to become pseudorandom chains and prevent dictionary attack. Like in (Raiciu & Rosenblum. 2006) they adapt the protocol of Song et al. (Song et al. 2002) but they do not motivate the use of this particular solution against lighter ones. Furthermore their way of ensuring privacy is through multiple path routing thus affecting the performance. To conclude, an interesting work which may be used as a strong basis for privacy in content-based publish/subscribe systems, is proposed in (Shikfa et al. 2009). In this paper, the authors offer a scheme which enables privacy preserving routing with no shared secret between end-users by cryptographic means while leveraging strengths of past works.

3.2.1.3 Automating the subscriptions, and dynamic events

Many research efforts conjecture that utility of publish/subscribe systems in many domains, especially those in heterogeneous environments, can be increased by automating the management of subscriptions (Brenna et al. 2006; Laliwala et al. 2006; Hinze et al. 2009; Rodriguez-Dominguez et al. 2010). Mechanisms for automatic or semi-automatic subscriptions to events based on the consumer's preferences and context are discussed below, including references to some visible efforts and results.

The work of Brenna et al. (2006) consists of an approach that shows how subscriptions in publish/subscribe systems can be automated with a novel application of information retrieval techniques. They propose the Reef framework that is geared toward monitoring Web browsing history and making recommendations to a user through a browser extension, in an automatic way. Reef monitors the behaviour of users (i.e. the Web sites they visit, the documents they read and texts they write) and feeds the logged data to recommendation services. Well-known information retrieval techniques, including correlation of logs from multiple users, were used to produce both topic- and content-based subscriptions. The user can receive relevant information without even subscribing for them, as the system delegates to a recommendation service the task of creating, refining, and removing subscriptions in a publish/subscribe mechanism. The events from subscriptions are displayed either as text or as a small image with annotation. The architecture of Reef can be broken down into four components (i.e. attention recorder, attention parser, recommendation service, subscription frontend). The attention of a user is captured by an *attention recorder*. This recorder runs in the Web browser and captures the URIs viewed by the user. This raw data is processed by an *attention parser*, which looks for tokens that match the specification of name-value pairs of the publish/subscribe system. Using the tokens found by the parser, a *recommendation service* makes recommendations on what subscriptions to place and which to remove. In response, a *subscription frontend* activates or deactivates subscriptions, as well as receives and displays the events that arrive. Reef has been developed as a Firefox plug-in written in Javascript along with a server side deployed as a standard “LAMP” setup (i.e. Linux, Apache, MySQL, PHP, and Python). Both centralized and peer-to-peer designs have been implemented. Nevertheless, Reef presents high scaling cost while the fact that its recommendation functionality implicitly gains extensive knowledge on the habits and interests of every user creates privacy issues. In addition, Reef relies on term weighting mechanism for selecting 30 terms that encompass a user's general interests, so the problem of supporting users with many diverse interests, hasn't been addressed.

Another approach has been proposed in (Laliwala et al. 2006) where an architecture has been presented for semi-automatic topic-based subscriptions to events from the correlation of user defined criteria with semantics. They demonstrate their work in real life Agricultural Information Systems (AIS). The specific approach requires a user registration based on his/her preferences, context and location. A dedicated component, the WS subscription manager, uses a minimal amount of user inputs; it extracts additional criteria from user profile and other services and correlates them with semantics, by querying an ontology server. This module also carries out the task of managing subscription and delivering event notification to the targeted subscribers. Using that knowledge the system basically, searches the available event sources and suggests subscriptions to the user. Therefore depending upon the user's requirement and context, the system is responsible for identifying and generating events that will be subsequently pushed to the end-users. This work mainly depends upon WS-Notification (WSN) which is a collection of specifications and white papers namely, Publish-Subscribe Notification for WS (Graham et al. 2004), WS Topics, WS Base Notification and WS Brokered Notification. WS Base Notification is used in order to define the interface of WSN consumer and producer while WS-Topic defines the hierarchy of interested item “topic”. WS Brokered Notification defines the interface of the Notification Broker that manages the message exchanges, aggregates them and acts as a mediator.

Berkovsky and Eytani (2005) try to replace the traditional notion of manual event subscription with a new form of automatic context-aware subscription. They present their work in mobile commerce applications as they focus on obtaining context-aware personalization through publish/subscribe systems (Liu and Plale 2003), based on the user's location, personal preferences and interests. They propose a semantic approach to the publish/subscribe paradigm, where the user's dynamically changing context may serve as an implicit subscription. The system automatically translates the user's context into a semantic query, performs matching and notifies the other side. In this way the users are not required to specify their interests manually and personalized information is delivered automatically. They developed publish/subscribe functionality over UNSO's (Ben-Asher and Berkovsky 2004) semantic infrastructure (flexible ontology-based hypercube graph topology for P2P networks) as a flexible way to match between service providers and the interested users. A hashing mechanism (mapping each property in the context to the respective hypercube coordinate) was proposed that automatically translates user's context into a semantic query. The query acts as an implicit subscription to the underlying hypercube data structure.

When the user context changes, a new subscription is dynamically created. The new query is matched with the existing service providers and the subscriber is notified.

In (Hinze et al. 2009) an architecture was proposed as an implementation of a tourist information provider (TIP) that provides information about sights to travellers based on their context (i.e., their personal preferences and their location). This architecture introduced an event-based middleware, with which every local service interacts. The component representing this middleware layer is called the broker. For client/server interaction, co-operating services exchange their information via their local brokers, i.e., every communication from a client to a server service or vice versa is only handled by the brokers and is transparent to the services. According to this approach, when a service is started, it connects to a broker, advertises the provided information and subscribes for other information. The broker maintains this information in a registry and is responsible for service registration, service deregistration, publishing and filtering events. A service can register and un-register itself from a broker. That means it advertises events and subscribes for events that it is interested in. Which events a certain service subscribes to or which it is going to provide depends on the type of the service, i.e., purpose of the service. The service in (Hinze et al. 2009) provides a service description, an advertisement, functional conditions and subscription rules. These rules enable the service to prioritize certain event types over others or to choose between several publishers. The evaluation result of such a rule leads to an event subscription.

The work in (Rodriguez-Dominguez et al. 2010) tries to extend the publish/subscribe communication paradigm by providing events with dynamic redefinition capabilities for coping with the needs of mobile ad-hoc networks (MANETs). They present an event model and a set of operations to publish events and subscribe to them. Events are semantically related to certain topics and are defined as communication elements (composed of a set of pieces of information) that can be produced by communication participants as results of a change in their state. These pieces of information are encapsulated into event nodes. An event node is structured as an identifier-type-value tuple. These events are coded as dictionary structures that are collections of unique pairs of keys and values. Keys are strings that represent the identifier of each event node and values are arrays of bytes. These arrays of bytes represent values as specified by the IceP communication protocol (Henning and Spruiell 2009) that has been used to codify network messages. In order to address the dynamic nature of ubiquitous systems, they have introduced the notion of *dynamic events* to allow adapting events to the changes that take place in the system (e.g. to add devices, to remove them, to modify the connections between the existing ones, etc.). This adaptation takes place by redefining the set of event nodes at run-time. Additionally, event topics can be dynamically inferred as it will be the same with event node topic in case the event is composed of only one event node, or it is deduced from the semantic association between the topics of each event node. The subscription to an event was defined as a “filter over a portion of the event content, expressed through a set of constraints” (Corsaro et al. 2006). The proposed events can support both topic and content-based subscription variants based on whether the communication participants show interest in receiving events that are semantically related to a topic or they need in addition a set of conditions over the event nodes to hold.

3.2.2 Relevant works for event storage and retrieval using a pub/sub approach

Events will be represented as interlinked RDF data (basically RDF triples, but that may be extended into quadruples if needed), so this part focuses on RDF data storage and associated RDF data retrieval mechanisms.

The storage and retrieval of RDF data in a distributed P2P environment raise specific challenges, because of the highly distributed nature of the storage and of the form of the queries to retrieve this data. In essence, as the RDF data model supports not only simple, but also, complex queries such as conjunctive and disjunctive queries, an additional effort to design adequate algorithms and techniques to support advanced querying beyond simple keyword-based retrieval and range queries is required. Consequently, a particular attention has to be made regarding the data indexing since it has a great impact on the query processing phase. Overall, three main aspects have to be taken into consideration while investigating RDF data storage and retrieval in P2P systems:

Data indexing How can we take advantage of the format of the RDF data model in order to efficiently index RDF triples?

Data retrieval How can we take advantage of P2P search mechanisms to efficiently query and retrieve RDF data in large scale settings? Moreover, what is the impact of the data indexing methods on the query processing algorithms?

Data integration [2] As a query, composed by a set of sub-queries, can be answered by several nodes, how to efficiently combine RDF data residing in different locations and provide a unified view of this data?

The combination of P2P communication model with RDF data has become a very active research area aiming at sharing and processing a huge amount of data. Several works (see survey (Filali et al., n.d.) for further details) present solutions based on structured P2P overlay networks (Aberer et al., 2004; Battré et al., 2006; Cai & Frank, 2004; Gu et al., 2007; Harth & Decker, 2005; Karnstedt et al., 2008; Koubarakis et al., 2006; Liarou et al., 2006; Matono et al., 2007; Nejdl et al., 2003; Della Valle et al., 2006). In addition to the basic challenges related to such large scale infrastructure (network partition, network maintenance, etc.), enabling complex querying of RDF data on top of such infrastructure requires advanced techniques for data indexing and query processing algorithms. Regarding data indexing, several approaches, e.g., RDFPeers (Cai & Frank, 2004), Atlas (Koubarakis et al., 2006), YARS (Harth & Decker, 2005), RDFCube (Matono et al., 2007), Battré *et al.* in (Battré et al., 2006), while based on different overlay topologies, share almost the same data indexing model by *hashing* the RDF triple elements. The main advantage of such indexing strategy is that triples with the same subject, object and predicate are stored on the same node and thus can be searched locally and without needing to be collected from all data sources. However, individual nodes, responsible for overly popular triples (e.g., rdf:type, dc:title), can be easily overloaded resulting in poor performance. One possibility to address this issue is to use multiple hash functions to ensure a better load distribution as proposed in (Ratnasamy et al., 2001) and recently in (Mu et al., 2009). The second category of RDF-based P2P approaches harness the semantic of RDF data either to build a "semantic" layer on top of the P2P overlay (e.g., (Cudré-Mauroux et al., 2007; Karnstedt et al., 2008)) or to adopt a semantic clustering approach and organize the P2P layer as function of the semantic of the stored data (e.g., Edutella (Nejdl et al., 2003), DSS (Gu et al., 2007), S-RDF (Zhou et al., 2009), Bibster (J. Broekstra and Marc Ehrig and Peter Haase and Frank van Harmelen and Maarten Menken and Peter Mika and Bjorn Schnizler and Ronny Siebes, 2004)). Others have extended the basic RDF data model by adding the "context" concept as in YARS (Harth & Decker, 2005) and PAGE (Della Valle et al., 2006). By taking into consideration data semantics in the data indexing phase, these approaches try to improve data lookup. However, this requires an additional effort to maintain the mapping between the semantic and the overlay levels.

Most of the presented works aim at adding data *availability* feature to the RDF storage infrastructure through *data replication*. However, data replication further raises several issues. Thereof, three main challenges have to be taken into consideration: which data items have to be replicated; where to place replicas and finally how to keep them consistent. In P2P systems there has been a lot of work on managing data replication. In (Ktari et al., 2007), Ktari *et al.* investigated the performance of several replication strategies under DHTs systems including neighbour replication, path replication and multi-key replication. As argued in (Ktari et al., 2007), the data replication strategy can have a significant impact on the system performance. Further effort may go into exploring more "dynamic" and adaptive replication approaches as function of data popularity or average peer online probability (Kneevi et al., 2009). Although the replication techniques increase the data availability, they come not only at the expense of more storage space but can also affect the *data consistency* (e.g., concurrent update for the same triple). Moreover, the data inconsistency issue becomes even more intricate under the partitioning of the P2P network. Thus, an update operation might not address all replicas as a node storing a replica can be offline during the update process. Therefore, trade-offs are made between high data availability, data consistency and partition-tolerance. Brewer brought all these tradeoffs together and presented the CAP theorem (Gilbert & Lynch, 2002) which states that with Consistency, Availability, and Partition-tolerance, we can only ensure two out of these three properties. Recognizing which of the "CAP" properties the application really needs is a fundamental step in building a successful distributed, scalable, highly reliable system.

On the query processing point of view, sharing RDF data imposes new challenges on the distributed storage infrastructure related to supporting advanced query processing algorithms beyond simple keyword-based retrieval. Therefore, we need to take a deeper look at how the query can be optimized before being processed. The various approaches for data retrieval and integration are mainly achieved either by fetching triples to the query's originator which coordinates the query evaluation, or forwarding intermediate results (e.g., (Liarou et al., 2006; Cai & Frank, 2004)) whenever the query is partially resolved. However, we firmly believe that the first approach may not efficiently resolve conjunctive queries where each sub-query leads to a huge result set while the final join operation between them conducts to a small set. *Caching* results (Battre, 2008; Liarou et al., 2006; Zhou et al., 2009) can alleviate this challenge, at least for similar queries, but can also affect the data consistency. Thereby, a trade-off is made between the network resource usage on one side and the information staleness on the other side. Others approaches such as in (Battre, 2008) decide at runtime whether the current result set has to be fetched to the query's originator or is continued to be forwarded to others neighbours [3] . As the query processing

[2] Generally, this operation is combined with the data retrieval phase.

[3] The decision is taken based on the estimated traffic of each operation.

becomes more crucial especially when processing huge data sets such as the Billion Triple Challenge 2009 (BTC2009) dataset[4] , some works have been proposed to reduce the large data sets to the interesting subsets as in (Gregory Todd Williams & Hendler, 2009). To do so, BitMat (Atre et al., 2008), based on Bit Matrix conjunctive query execution approach, is used to generate a compressed RDF structure. Therefore, a dominant challenge related to the distributed query processing (Kossmann, 2000) is how to improve the query performance and find an "optimal" query plan in order to enhance the query performance and reduce the communication cost. An already explored direction towards the query optimization plan is introduced by OptARQ (Bernstein et al., 2007). It is based on Jena ARQ[5] , uses the concept of triple pattern selectivity[6] estimation[7] . It aims to find the query execution plan that minimizes the intermediate result set size of each triple pattern by join re-ordering strategies. Thereof, the smaller the selectivity the less intermediate results it is likely to produce and the earlier it should be executed in the query execution plan. For a similar purpose, RCQ-GA (Hogenboom et al., 2009) uses genetic algorithms in order to perform an efficient evaluation of RDF chain queries.

Further mechanisms have also to be devised in order to take into account long-standing queries for RDF data. Publish/subscribe works focusing on processing RDF data (Cai et al., 2004; Gu et al., 2007; Alex et al., 2004; Koubarakis et al., 2006; Liarou et al., 2005, 2008; Ranger & Cloutier, 2005) have slightly different mechanisms for dealing with notifications and subscriptions. Most of them, being based on structured peer-to-peer overlay directly take advantage of the underlying indexing mechanisms. Triples are indexed and retrieved using cryptographic functions (except for (Ranger & Cloutier, 2005)) and this, generally, for each RDF triple element and their combination. This multi-indexing technique induces a non negligible processing load, and has pros and cons.

Events, because of their dynamicity are challenging to manage in large scale settings. Typically, in publish/subscribe systems, from a QoS point of view, the focus is generally on improving the subscription mechanisms and the event notification process. Most of the works surrounding publish/subscribe systems for RDF data processing do not extensively offer QoS guarantees from the notification point of view (or even for subscriptions for that matter). Some use replication techniques to ensure data availability, but few if any discuss other aspects of QoS in depth such as the latency, bandwidth, delivery semantics, reliability and message ordering. As argued in (Mahambre et al., 2007), a lot of efforts are underway to build a generation of QoS-aware publish/subscribe systems. As a matter of fact, adaptation and QoS-awareness constitute major open research challenges that are naturally present in RDF-based publish/subscribe systems. Delivery semantics, reliability and message ordering represent the main challenges from a notification point of view. Ensuring an "at-most once" delivery semantic, taking into account link reliability or causal delivery between events are some examples of such key issues.

Since we plan to build the Event Cloud on top of structured overlay networks, one should take advantage of the present structure to disseminate events in a smarter way, as in Scribe (Castro et al., 2002) or using efficient dissemination algorithms atop structured p2p networks (El-Ansary et al., 2003). The problem with Scribe is that it only provides best effort guarantees, even if authors state that strengthening the algorithm to make it reliable should not be that difficult. One possible solution, that can be used for a large family of structured overlay networks, would be the use of a *pseudo-reliable* broadcast abstraction as presented in (Ghodsi, 2006) which ensures the correct dissemination of messages provided the root of the tree which is constructed and used to broadcast the events does not crash.

The study of these algorithms is now being revisited by the Distributed Systems research community in the context of *dynamic distributed systems* (Baldoni et al., 2007; Baldoni & Shvartsman, 2010). A dynamic distributed system is, according to the same authors, *a continuously running system in which an arbitrary large number of processes are part of the system during each interval of time and, at any time, any process can directly interact with only an arbitrary small part of the system.* This informal definition sheds a light on the nature of such systems, which is a coupling of two distinct concepts that are *large-scale* and *dynamicity*. The Event Cloud falls into the realm of this definition combining the two aforementioned concepts; not only publishers and subscribers come and go at will (churn), but data itself is dynamic (events are continuously fed into the system, sometimes rendering prior events obsolete).

---

4 http://vmlion25.deri.ie/.

5 http://jena.sourceforge.net/ARQ

6 Selectivity of triple pattern *T* is the fraction of triples that satisfying this pattern.

7 This is performed by collecting statistical information about the ontological resources.

3.3 Middleware for integration and governance of SOA

The PLAY platform is a middleware whose aim is to help integrate multiple SOA applications (i.e., simple and composite services) at large scale by relying upon an event-driven approach (EDA). Integrating services to constitute SOA applications needs an underlying middleware (coined as EAI and ESB). A more recent concern in SOA in general is how to federate multiple and independently deployed service-oriented architectures in a way that allows an additional yet loosely coupled integration of the respective applications. As a consequence, the PLAY middleware has also to be capable of publishing and notifying events from and between multiple SOAs hosted by independent yet federated SOA application hosting middlewares. Subsection 3.3.1 briefly gives an associated state of the art.

Another and very important matter for SOA is the notion of governance. This encompasses a multi-layer activity by which composite services applications are constantly monitored and eventually managed (adapted) in order to fulfil some possibly high-level QoS objectives following some governance rules. SOA governance needs to be extended in order to incorporate the fact that service integration is to be event-driven. Subsection 3.3.2 briefly presents the key concepts and state of the art.

3.3.1 Support for SOA and federated SOA

The Enterprise Service Bus (ESB) concept is a follow-up of the Enterprise Applications Integration (EAI). The main goal of the EAI solutions is to provide integration of heterogeneous applications. ESB technology leverages best practices from EAI mechanisms and service-oriented architectures SOA. Remote applications integration is based on sophisticated mechanisms such as brokering; message transformation and routing, quality of service support, service orchestration, etc. Moreover, an ESB offers a backbone implementation for service-oriented architectures by connecting distributed services consumers with providers (Schmidt et al., 2005). The main functionalities of an ESB are supported by components that can be both centralized or distributed over a network. Besides, ESB technology allows several ESB to be connected over a large network, achieving a pervasive integration, which is strongly required in wide area and dynamic systems. Other key characteristics are listed in Chappell's book (Chapell, 2004) namely, standardization, service discovery, high distribution and reliability, etc. Finally, standard-based ESB concepts are strongly attracting industrials. This section considers mainly technological ESB architectures and how they can be leveraged to provide federated SOA architectures.

Existing proprietary and open source ESB approaches provide powerful integration and orchestration solutions for distributed services. For instance, BizTalk server (BizTalk Website, 2010) is a proprietary Microsoft ESB based on an extended server. A toolkit offering a collection of tools and libraries supporting a loosely coupled and dynamic messaging architecture is provided. It functions as a middleware that provides tools for mediation between services and their consumers. Open ESB (OpenESB Website, 2010) is a java-based open source enterprise service bus providing application integration. It supports open standards as SOAP, WS-*, XML standards, and a Net beans-based IDE. Both BizTalk server and Open ESB rely on centralized architectures and do not provide native event-based service interaction but rather rely on the request-response paradigm only.

Meanwhile, even if this generation of ESB provides powerful means of integration and services orchestration, it is still not adapted to highly distributed environments. Therefore, a more flexible and dynamic vision is needed to cope with scalable distribution and particularly scalable choreography. Actually, widely distributed systems are all about the large size of all the surrounding dimensions including persons dealing with the system, amount of data routed, number of services interacting, etc. Brokering and routing such a large number of services using a unique centralized bus creates a bottleneck and a lack of flexibility. These issues can be tackled by using the high distribution ESB feature. It represents a key point since the integration logic can be broken-up into distinct manageable pieces. It is worth underlining the fact that, when compared to the centralised ESB, the distributed ESB (DSB) is more adapted to widely distributed environments.

Existing distributed Enterprise Service buses are both open source and proprietary. Based on Apache ServiceMix ESB (ServiceMix Website, 2010), Fuse ESB (Fuse ESB Website, 2010) is an open source OSGi-based distributed ESB. It supports BPEL processing, and both OSGi (OSGI Website, 2010) and JBI (JBI specification, 2005) deployment and runtime. It is promoted to be distributed since it offers a remote console to control the runtime bus.

From our point of view, the Fiorano ESB (Fiorano ESB Website, 2010) has a different and more scalable vision of distribution. It is built upon a hybrid architecture relying on a hub and spoke management layer and on a peer-to-peer system. This allows ESB peers dissemination over a wide network. It supports the main ESB features such as application integration, service orchestration, event management; etc. The latter is a proprietary Fiorano product.

Sonic ESB (Sonic ESB Website, 2010) is the Progress Software Corporation distributed ESB. It achieves services integration and distributed operations management. Moreover, it focuses on security in distributed domains and achieves BPEL orchestration. Sonic ESB is a proprietary product.

Petals ESB (Petals ESB Website, 2010) is an open source distributed ESB based on the JBI specification. While JBI is an integration specification, Petals ESB developers extended the specification to provide a distributed environment since the first major version of the product in 2006. This distributed and highly configurable approach is one of the main advantages of Petals ESB, which is completely different from other Open Source products. In this way, Petals ESB provides a multi-node platform leading to a moderately scalable architecture. It handles BPEL process orchestration, business process monitoring and management, as well as heterogeneous application integration through several protocols connectors. Around the bus, a set of open source tools providing process design and configuration, services management and monitoring, are offered. Distribution is also addressed through the leveraging of the bus service registry. In (Callaway, 2008) a remote and wide service discovery is achieved by distributing the service registry. Meanwhile, although this solution is interesting due to the way that it helps a more global service discovery and awareness, a larger vision is required. In order to adapt to distributed systems, Petals ESB combines both distributed service registry and multi-site architecture.

While some of these products already provide distributed features with the capability to build a virtual ESB from several ESB nodes (running on many hosts), few of them already address the federation aspect. Moving from a standard SOA to a federated SOA is a trend topic these days. The main questions to be addressed are: how to scale the SOA infrastructure across the enterprise, enabling service interactions and facilitating service reuse not only in the same management domain but also between different ones including with external partners in a governed way? Federating SOA means federating ESBs, but also service registries, identities and service management.

As being involved in research projects, Petals ESB features are continuously improving through innovative works. A current research work in FP7 SOA4All (SOA4All Website, n.d.) European project deals with evolving Petals ESB to a Distributed Service Bus (DSB) and to DSB federations for large scale SOA (Baude et al., 2010) using the Internet as a communication channel between management domains. At the architecture level, service registries, message routers and cross service bus message transport layers are the components that are scaled to reach the federation level.

Technically, by using the Petals ESB highly dynamic software architecture based on a component model, the internal Petals components have been extended to provide the federation feature. This customization is also based on a federation framework developed within the SOA4All project which acts as the core communication layer and provide advanced connectivity between domains in a secure and efficient way. More specifically, it has been developed with multi-domain as a target, enabling to freely mix clusters, grids and clouds nodes as required (Baude et al., 2010, Mathias, 2010). This federation framework is in charge of routing the federation messages to the right endpoints by using innovative communication paradigms as described in final SOA4All runtime implementation (SOA4All D1.4.2B deliverable, 2010). Based on this federation framework, a federation API has been introduced to be used by DSBs to connect and communicate with the federation. This API contains all the operations, which have to be implemented and exposed by domains to provide the core federation feature at the DSB level. It means that, for example, the API contains operations to lookup remote endpoints, to invoke services and to join or leave a federation. All federation actors have to provide endpoints implementing this API in order to join a federation and to be reachable.

3.3.2 Governance

"SOA: The building blocks of EDA" (Taylor, 2009), is a catchy chapter title and is the real baseline of the EDA governance.

By following this, it is quite clear to say that an efficient EDA platform should always be built on top of an efficient SOA one. By accepting this rule, the way an EDA platform must be governed is directly linked and potentially extends the way an SOA platform is governed.

In this section, we first introduce the state of the art of SOA governance solutions, followed by the steps to achieve the move from SOA to EDA governance are detailed.

3.3.2.1 SOA Governance

Since their inception, Service-Oriented Architecture was revealed as being the de facto paradigm for future systems. They combine best practices paradigms inspired from previous application models. Modularity, encapsulation, fine-grained granularity, publication and discovery help SOA to be widely used by developers and users. As a consequence, we have noticed enterprise systems moving to this new trend. For instance, a market study made by AMR research institute in 2005 surveying over 134 different companies has shown a great interest for SOA technology, 20% has already implemented SOA in their systems, 50% were planning to implement it under 2 years and only 26% were not interested in it.

As systems are moving from classical IT to innovative SOA, essential functions need also to be exported and adapted. Governance is a first and foremost function in IT systems. It ensures the best interests of an organization to be met and dong so through corporate decisions from strategy to execution (Marks, 2008). MIT research scientist Peter Weill; gives a definition that goes in the same direction, according to him, IT governance is "specifying the decision rights and accountability framework to encourage desirable behaviour in the use of IT." IT managers are then concerned with decisions, processes, and policies to encourage the behaviour that contributes to success. It can even go further including leadership and organisational structures and processes that ensure that the organisation's IT sustains and extends the organisation's strategies and objectives (Afshar, 2007).

In (Brown 2006), the authors give a global definition for governance as:

- Establishing chains of responsibility, authority and communication to empower people (decision rights)
- Establishing measurement, policy and control mechanisms to enable people to carry out their roles and responsibilities

SOA governance extends IT governance for the purpose of ensuring the SOA success. Lack of governance can be a serious impediment to success and the most common reason for the failure of SOA projects (Afshar, 2007). In (Papazoglou, 2007), authors identify SOA governance as a major research area in the field of SOA design and development. Nevertheless, SOA governance is not clearly defined in literature. We rely on the following definitions to identify key topics in SOA governance:

- In (Afshar, 2007) Oracle researchers define SOA governance as the interaction between policies (what), decision-makers (who), and processes (how) in order to ensure SOA success. SOA governance is able to ensure that all of the independent (SOA) efforts come together to meet enterprise requirements. It covers the following levels: design, development, deployment, and operations of a service.
- In (Brown, 2006) IBM researchers assume SOA governance is an extension of IT governance specifically focused on the lifecycle of services, metadata and composite applications in an organization's service-oriented architecture. SOA governance extends IT governance by assigning decision rights, policies and measures around the services, processes and lifecycle of SOA to address such concerns as: service registration, versioning, ownership, funding, monitoring, auditing, publishing, discovery, etc.

> *SOA governance is doing the right SOA things (**processes, decisions**) the right way (**policies**) for the SOA stakeholders* (***decision-makers***) (Marks 2008)

- In (Marks, 2008) the author aggregates several definitions to give his own one: SOA governance is the definition, implementation and on-going execution of an SOA stakeholder decision model and accountability framework that ensures an organization is pursuing an appropriate SOA strategy aligned with business goals, and is executing that strategy in accordance with guidelines and constraints defined by a body of SOA principles and policies. SOA policies are enforced via a policy enforcement model, which is realized in the form of various policy enforcement mechanisms such as governance boards and committees; governance processes, checkpoints, and reviews; and governance enabling technology and tools.

Through the previous definitions the essence of SOA governance is revealed in the trilogy (decision, process and policy):

SOA governance needs to be achieved at several levels from strategy to execution. Governance tools provide the functionality required to support the governance processes associated with a specific SOA initiative including the following:

- SOA policy management and enforcement
- Registry/repository and metadata management
- Statistical and KPI data collection
- Governance of services in the cloud
- Monitoring and management
- Application and service life cycle management
- Interoperability with other SOA governance technologies

The following approaches address SOA governance. This state of the art presents both academic and industrial solutions. Examples from industry are more numerous as SOA governance methodology is mostly driven by SOA vendors.

Authors in (Derler, 2007) address the SOA governance by proposing a generic model and two governance tools. Services are described according to their life cycle. Actually, activities and roles, which are relevant during the service life cycle, are considered. Three roles are identified, the service developer, the product manager and the administrator. The product manager determines customer requirements, specifies a service for the business logic needed and is responsible for associating the service with a product. The developer is responsible for implementing the service and decides how the service is structured on a technical level. Meanwhile, the generic model contains elements describing clients and service proposals. It also contains elements for products and service modules. Basing on the previous model the authors propose two tools: a service repository console and a service browser. The Service Repository Console is used for creating service proposals and service descriptions, for specifying service relationships, and for defining service installations. The Service Browser is used for searching and browsing the service repository and for investigating service details, service relationships and service status. This approach proposes a standard way and a solid model to consider SOA governance. Nevertheless, the authors don't address governing practices for neither orchestration nor choreography of services. Neither do they consider ultra large scale system requirements.

A methodological approach is presented in (Schepers, 2008); it relies on a six-step governance life cycle. The authors claim that SOA governance is more than a process, it is all about continuously aligning strategic goals. Actually, authors define six steps (1) defining an SOA strategy; this phase aims to aligning SOA with business requirements, (2) aligning organization to SOA by assigning responsibilities and establishing project groups, (3) managing service portfolio ensures that a sound method is used consistently to decide which services need to be developed; (4) controlling service lifecycle concerns the development and delivery of individual services in a SOA (service granularity and consistency, management procedures, etc.), (5) incorporating policy enforcement refers to performing service checks to verify it complies with policies, (6) service level management specifies the contract stating service levels and possible fees. This should be specified for each service. In this paper, the authors give clear guidelines to achieve SOA governance; however no details are given for describing practical governance prototypes.

IBM's approach is presented in (Brown, 2006) as a four-step life cycle approach to governance. It consists of a (1) planning phase during which the need for governance is established and the existing mechanisms are assessed, a definition phase (2), during which the desired governance framework, including new and modified principles, processes, organizational structures and roles are established, an enabling phase (3), where the new governance framework is introduced into the enterprise and finally, a measurement phase (4), during which the metrics are gathered and analysed to refine the governance process. Relying on this framework IBM proposes IBM WebSphere Service Registry and Repository (WebSphere Website, 2010), which offer a governance solution in IBM SOA; it supports service discovery, and accessing. Besides, it offers features for service metadata management. Advanced management capabilities are provided by IBM Tivoli product (Tivoli Website, 2010).

Oracle Company proposes a framework and a six-step-based solution. Relying on a previously stated SOA maturity model, the authors in (Afshar, 2007) give a roadmap for SOA governance. Key leverage points for policies are captured according to business areas. These are: architecture, technology infrastructure, information, finance, portfolios, people, projects and operations. Then, policies need to be designed and enacted across the cited areas. Depending on the area, policies model and medium may be different; some policies can be captured in technology solutions or simply in policy documents. For instance, operational policies such as governing services at runtime may be addressed through the adoption a technological solution as a registry/repository or a Web service management, whereas, architectural and funding policies can be captured through documents distributed through the organization. Meanwhile, the authors give six generic steps and best practices in order to apply and benefit from SOA governance, these are: (1) Defining goals, strategies, and constraints, (2) Defining standards, policies, and procedures for financial, portfolio, project, services, (3) Defining metrics for success, (4) Putting Governance mechanisms in place, (5) Analysing and Improving existing processes, and (6) Refining and going to the next level of SOA maturity.

Basing on the previous framework, Oracle proposes a proprietary product Oracle SOA Governance 11g (Oracle Registry Website, 2010) (Oracle Repository Website, 2010) and the recently acquired Amberpoint, consisting of an enterprise repository, a service registry, an enterprise manager, and a Web service manager. Both Oracle and IBM are leading companies in this domain and their products cover most basic governance features.

Mule Galaxy (Mule Galaxy Website, 2009) is a service-oriented architecture governance platform. It provides an SOA registry/repository. Galaxy aids in the management of SOA by supporting features such as lifecycle, dependency and artefact management, service discovery and reporting, and application deployment management. It is an open source product. Although Mule ESB Enterprise includes a service registry/repository that assists in artefact management and publishing, its policy enforcement capabilities (particularly, the implementation and modification of life cycle management processes) fall short of similar capabilities in closed-source offerings.

WSO2 Registry (WSO2 Registry Website, 2010) is an open source WS-based SOA governance registry. It is standardized and supports basic SOA governance and integration capabilities such as tracking SOA resources, managing services life cycles and controlling resource access. Moreover, WSO2 Registry provides a repository where resources and collections can be stored and managed, tagged, rated, logged, etc. Besides, SOA governance tools support common schema validation policies as WSDL validation, Web service discovery, lifecycle management, dependency relationship management and remote link support. Finally, WSO2 provides advanced features as ATOM protocol support, and local and remote registry management. Though WSO2 is seen as a visionary governance tool it may be improved by implementing a scalable subscription model and supporting some basic mechanisms dedicated to business applications.

Petals Master SOA Governance Solution (Petals Master Website, 2010) is an open-source governance tool. It is UDDI-based, providing basic governance features as service Registry/Repository, organization management, Service Level Agreement management and Integration with Service Runtime Environment. The Registry/Repository also provides management of policies that govern the behaviour of users (persons or systems), dependency management: between services and other SOA assets, lifecycle management, and reporting: usage indicators, policy violations, etc. Petals Master can be deployed as a standalone tool or as integrated in the service bus Petals ESB. Though it offers common governance functionalities and SLA and WS-agreement support, Petals Master needs to be improved by enforcing policies and providing scale-up capabilities in order to suit large distributed systems. The involvement of Petals Link in research projects gives an opportunity to implement new innovative features in Petals Master.

Other commercial products exist as Progress's SOA tool CentraSite (CentraSite Website, 2010), Hitachi's product Cosminexus (Cosminexus Website, 2010) or Governance Interoperability Framework (GIF Website, 2010), etc. They typically provide service discovery, dependency management, policy management, change notification, authentication and identity management, policy management, and federation with other repositories. An ideal SOA governance tool should benefit from best practices from the existing tools.

3.3.2.2 EDA Governance

As described in previous paragraphs, there are many SOA governance solutions, all based on their own vision and approach. The main problem to be addressed is how an event-based communication solution can impact existing ways to govern systems both at design-time and at runtime:

1. At design time, governing an EDA is equal to governing a SOA system. This design phase needs event providers' and consumers' descriptions as a SOA also provide service descriptions. The state of the art of SOA governance points out that governance will not be possible without using well described services based on standard descriptions such as the well-known Web Service Description Language (WSDL). It seems also interesting to study how this type of governance will impact (or not) the way Service Level Agreement (SLA) contracts are described.
2. At runtime, it seems important to study how this event-based communication pattern impacts the way Key Process Indicators (KPI) are used in standard SOA monitoring and enforcements solutions.

3.3.2.2.1 Design Time

As described above, governing an event-based platform should be like governing a service-based platform. An efficient way to achieve this task is the capability to describe event-producers with standards.

Both W3C and OASIS consortiums provide a collection of specifications for event-based mechanisms, respectively the W3C WS-Eventing specification (WS-Eventing specification, n.d.) and the OASIS WS-Notification specification family (WS-Notification specification, n.d.). These two specification families are based on well-known standards (WSDL, XML, XSD) and focus on messages exchanged between actors mainly for subscribe/unsubscribe operations. Describing service and business message payload (the notify message content) are not the goal but are introduced, or can be deduced. The current study goal is not to compare specifications but to introduce them and to quickly show how events can be described.

The W3C WS-Eventing specification "describes a protocol that allows Web services to subscribe or to accept subscriptions for event notification messages" and "relies on other Web service specifications to provide secure, reliable, and/or transacted message delivery and to express Web service and client policy". In (WS-EventDescriptions specification, n.d.) "Event Types and Event Descriptions" introduces the way to describe and retrieve event descriptions by using the W3C WS-EventDescriptions specification. This specification "describes a mechanism by which an endpoint can advertise the structure and contents of the events it might generate". As expected and like all the WS-* specifications, this specification uses XML schema to describe the business message content format, but also define how to retrieve this description from any WS-Eventing based endpoint i.e. what mechanisms a client (the governance tool) may use to retrieve this description. In the current case, the specification just recommends that the event description should be available by using the WS-Eventing EventSource policy assertion.

The OASIS WS-Notification family provides "a standardized way for a Web service, or other entity, to disseminate information to a set of other Web services, without having to have prior knowledge of these other Web Services. They can be thought of as defining Publish/Subscribe for Web services". This family is composed of three specifications: WS-BaseNotification (WS-BaseNotification specification, n.d.) dealing with standard mechanisms and description of publish/subscribe messages; WS-BrokeredNotification (WS-BrokeredNotification specification, n.d.) describing message exchanges involved in publish/subscribe of a message broker; and WS-Topics (WS-Topics specification, n.d.) defining mechanisms to organize and categorize items of interest for subscriptions known as topics.

Event description is definitely not the goal and is not addressed in the OASIS specifications. But there are ways to store and retrieve them by extending parts of WS-Notification based systems. Topics used in notification brokers uses namespaces, which can be anything. A solution is to use these namespaces to define links to service description stored in a service description repository. By using this type of implementation, the governance tool will be able to retrieve and govern the EDA platform.

By building an EDA using an existing SOA system, it looks interesting to reuse and extend the SOA governance mechanisms to create the EDA governance tool. It implies that event producers and consumers are considered as services and that the description of these services and their message format is available and can be retrieved from the governance tool. This role is clear for notification consumers, which must be able to receive notification messages, and in this case which implement a notification-aware interface. Defining notification producers as services is less obvious and semantically opposed to the service notion since a message is initiated by what can be seen as a client. However, in order to govern an EDA platform, it seems mandatory to define producers as services, to describe the notification message payloads in standard format and so to be able to retrieve these notification services from the governance tool.

Both Web service notification mentioned before provide some ways to register and retrieve notification producers and consumers from an EDA-enabled governance tool. Once available in the governance tool, notification actors can be governed i.e. classify services, describe and apply Service Level Agreement to services, and use other governance features like it is done with standard SOA services.

3.3.2.2.2 Runtime Level

At the runtime level, an EDA platform has to be monitored and managed like any SOA platform. Currently, services containers should be able to monitor activity in the platform and should be able to send (or store) monitoring events or raise alerts based on SLA contracts.

There are differences that are introduced by the runtime governance of an EDA platform: Service Level Agreements are quite natural to define and apply for services but are not for all event actors. If we go back to events and forget the services parallelism, can we talk about "Event Level Agreement"? Is there some need to define an agreement between an event consumer and an event producer like "The event consumer does not want to receive more than N messages in a defined period?". If yes, the main questions are "where to inject this agreement?", "What happens if the number of notifications is higher than defined?"…

Monitoring an SOA platform means the need to use standards mainly because platforms can be heterogeneous. The OASIS Web Service Distributed Management (WSDM) specifications (WSDM specification, n.d.) has been developed by management vendors and define notifications message formats which must be used by WS-Notification based systems to send monitoring and management information to any interested parties (subscribers).

OASIS WSDM uses Web services as a platform to provide essential distributed computing functionality, interoperability, loose coupling, and implementation independence. The OASIS WSDM working group has defined two specifications which can be used to provide efficient monitoring and management APIs, namely MUWS (Management Using Web Services) and MOWS (Management Of Web Services).

- MUWS defines how to represent and access the manageability interfaces of resources as Web services. Standard manageable resource definitions create an integration layer between managers and the different management protocols used to instrument resources.
- MOWS defines how to manage Web services as resources and how to describe and access that manageability using MUWS. It provides mechanisms and methodologies that enable manageable Web services applications to interoperate across enterprise and organizational boundaries. The MOWS specification allows integration of management with Web services-based business applications and processes.

Although this standard is well defined and really extensible, there are few references and implementations available. Like in few high-level domains and technologies, vendors' references are not published nor available.

The main references to be studied are both Open Sources implementations. The Apache Open Source consortium hosts the Apache Muse project (Muse Website, n.d.), which provides a Java framework upon which users can build Web services interfaces for manageable resources without having to implement all the 'plumbing', described in the WSDM specification.

A more interesting approach is defined in the FP7 SOA4All research project (SOA4All Website, n.d.), which also uses the WSDM specification to build the monitoring layer of the Federated Distributed Service Bus (SOA4All D2.3.1 deliverable, 2008). By using Web service notification mechanisms and WSDM as standard ways to communicate, the SOA4All platform is monitored in real-time. Concretely, it means that a Web-based graphical monitoring tool (acting as notification consumer) is able to draw response time diagrams in real time and to display alerts based on Service Level Agreements injected into the platform if, for example, a service response time does not match an injected rule.

3.4 Role of Middleware for Event-based and federated SOA in PLAY

The main role of the middleware in PLAY is the *storage*, *retrieval* and *brokering* of simple and complex events which will be used or delivered by the (D)CEP and by the SOA applications that rely on the PLAY middleware for being loosely-coupled along an EDA. As such, the PLAY middleware as a whole encompasses the respective middleware (ESB) the SOA applications are being deployed onto. Consequently, SOA governance extended to EDA must also include the non-functional properties and expected quality of service of the event cloud itself.

The event cloud will be comprised of federated P2P overlay networks combining high-performance elastic data processing infrastructure (grids, clouds, ...). Each node participating in the overlay networks will be responsible for managing parts of the events and will help in matching potential looked up events and disseminating them in a collaborative manner. Each node is also potentially an event broker responsible for managing long-standing subscriptions. In PLAY, the (D)CEP will emit multiple subscriptions for the different events and their content it wants to be notified upon. The main difference between a traditional database is that CEP is interested in the events "that just happened" or "that are about to happen" in the future.

In order to provide such features, the event cloud will be built on top of previous work done at INRIA by the OASIS and SARDES teams. The Semantic Space, which manages RDF data, will serve as the primary P2P substrate for the event cloud. That means that the complex events that the event cloud will manage will be represented as RDF triples (or a variant). An emphasis will be put on building the content-based publish/subscribe layer for managing long standing subscriptions for complex events. Further works around throughput optimality for group communication primitives (Guerraoui et al., 2010) and fault-tolerant mechanisms will be also investigated in the context of the Event Cloud.

3.5 Discussion

The various points in previous sections around the Event Cloud underline the fact that large scale applications with highly dynamic and volatile data built for the Cloud bring interesting challenges, especially regarding performance and QoS. Therefore, a key challenge in building a scalable distributed system is how to efficiently index events, how to manage subscriptions for complex events among peers participating in the P2P network in order to ensure efficient lookup services, and thus improve the efficiency of applications and services integrated into event-driven service-oriented architectures. The first evolution of our Semantic Space will be to include the support for events, that is, take into account the temporal nature of the data it stores. This evolution will have also to include further work around an efficient publish/subscribe layer on top of our Semantic Space along with fault-tolerant dissemination mechanisms. Finally, the interactions between the Event Cloud, the (D)CEP and the monitoring/governance tools will have also to be thoroughly discussed as integrated SOA applications will heavily rely on the Event Cloud capabilities.

4 State of the Art in Complex Event Processing

PLAY is pursuing the idea of bringing Complex Event Processing (CEP) (Luckham 2001) to the cloud. On the one hand, the cloud is a new paradigm for parallel computing which should enable large scale CEP applications.

Complex event processing is a technology, which allows correlating basic events to complex events closely aligned with the semantics of business processes related to the events. Climbing up the ladder from low level (e.g. sensors) to business events, complex event processing decouples providers of technical information and consumers of semantically rich information. As a consequence, changing technical event sources will not affect the delivery of semantically rich events to business applications. Most solutions in industry implement complex event processing by performing correlations in a centralized fashion. However, many applications such as location-aware or Radio Frequency Identification solutions are highly demanding in terms of the generated event load itself, as well as in terms of the computational complexity which requires more hardware capacity than just a single event processing engine. For example, it is estimated that large companies experience 0.1M – 10G business events per second. And at the same time only 10 % of the business events are handled (Gartner). Furthermore, if correlation of events happens over multiple domains, centralized solutions are expected to perform badly with respect to response times and are vulnerable to availability of event-driven information processing. In the next subsection we present the related work.

Recently, there has been a significant paradigm shift towards real-time computing. Previously, queries against databases and data warehouses were concerned with looking at what happened in the past. On the other hand, complex event processing (CEP) is concerned with processing real-time events, i.e., CEP is concerned with what has just happened or what is about to happen in the future.

An event represents something that occurs, happens or changes the current state of a system or the world. For example, an event may signify a sensor reading, a price-change signal, some piece of information becoming available, a deviation and so forth. An event can also represent something that did not happen (e.g., an absence of an event within a certain time frame).

4.1 Complex Event Processing

Complex event processing is a very active field of research and is being approached from many angles. A multitude of languages are proposed to formulate complex event patterns and different event processing paradigms are proposed to match these patterns over events (Luckham 2001).

Our approach for CEP is based on declarative (logic) rules. It has been shown elsewhere (Lausen, Ludäscher & May 1998) (Paschke, Kozlenkov & Boley 2007) (Bry & Eckert 2007) (Haley 1987) that logic-based approaches for event processing have various advantages. First, they are expressive enough and convenient to represent diverse complex event patterns. They come with a formal declarative semantics. Moreover declarative rules are free of side-effects (e.g. confluence problem). Second, integration of query processing with event processing is easy and natural (e.g. processing of recursive queries). Third, our experience with use of logic rules in implementation of the main constructs in CEP as well as in providing extensibility of a CEP system is very positive and encouraging (e.g. number of code lines in logic programming is significantly smaller than in procedural programming). Ultimately, a logic-based event model allows for reasoning over events, their relationships, entire state, and possible contextual knowledge available for a particular domain. Reasoning about temporal knowledge (i.e., events) and static or evolving knowledge (i.e., facts, rules and ontologies) is a feature beyond of the state of the art in CEP e.g., (Agrawal et al. 2008).

Apart from the above mentioned strengths, event processing systems elsewhere (Lausen, Ludäscher & May 1998) (Haley 1987) (Paschke, Kozlenkov & Boley 2007) based on various logic formalisms have some shortcomings, too. One significant shortcoming is data or event-driven computation. Deductive systems are rather suited for a request-response computation. That is, for given a request, an inference engine will evaluate available knowledge (i.e. rules and facts) and respond with an answer. This means that the event inference engine needs to check if this request can be deduced or not. The check is performed at the time when such a request is posed. If satisfied at the time when the request is processed, a complex event will be reported. If not, the event pattern in the request is not detected until the next time the same request is processed. However, because events typically arrive at arbitrary points in time (asynchronously) the pattern might become satisfied at any time between requests. In such a case, no immediate response is possible in such systems and thus the resulting event is not created in real-time. Contrary to this, event processing demands data-driven computation as handled by various approaches such as non-deterministic finite automata (NFA) (Agrawal et al. 2008), Petri Nets (Gatziu & Dittrich 1992), Rete algorithm (Forgy 1982) etc.. Unfortunately approaches grounded on NFA and Petri Nets do not feature reasoning capabilities; and Rete-based approaches may be integrated with deductive rules (Alves 2009) but have difficulties to handle aggregates over event streams, and to implement different event consumption policies (Chakravarthy & Mishra 1994).

The approach from (Paschke, Kozlenkov & Boley 2007) follows the mentioned request-response (or query-driven) approach. It proposes to define queries that are processed repetitively at given intervals, e.g. every 10 seconds, trying to discover new events. However, generally events are not periodic or if so might have differing periods and nevertheless complex events should be detected as soon as they occur (not in a predefined time window). To overcome this issue, in (Bry & Eckert 2007), an incremental evaluation was proposed. The approach is aimed at avoiding redundant computations (particularly re-computation of joins) every time a new event arrives. The authors suggest utilizing relational algebra evaluation techniques such as incremental maintenance of materialized views.

Our language for CEP, ETALIS (Anicic et al. 2009), is developed to close the gap between event-driven and logic-based systems. We present a rule-based language with a clear syntax and a declarative formal semantics. The language is powerful enough to effectively express and evaluate all thirteen Allen's temporal relationships (Allen 1983). Unlike other non-logic-based CEP languages (Agrawal et al. 2008) (Gatziu & Dittrich 1992), our language features inference capabilities; and unlike other logic-based approaches, it has a different execution model that compiles complex event patterns into logic rules and enables timely, event-driven detection of complex events. Finally unlike Rete-based approaches, recursive rules of our language enable processing of unbound event streams and applying aggregation functions on them.

Taking inference capability into account, logic-based CEP goes beyond the state of the art providing a powerful combination of deductive capabilities and temporal features.

4.2 Distributed Complex Event Processing

Web applications today create a lot of events. To make sense and react to these events, generic services are scarcely available. Mashup services only provide limited capabilities of detecting and aggregating such events. Thus, in this section we propose a generic Complex Event Processing (CEP) service for the Web. This service should be available to multiple tenants and be able to deal with a high throughput of events in an elastic manner, utilizing computing resources in a cloud. This section collects challenges in the current state of the art.

Evaluating event patterns in a single computing node is not enough to keep up with growing bandwidth of events and growing complexity of patterns. As applications become more and more decentralized and Internet-based, it is straightforward to consider a large-scale architecture for complex event processing.

Current solutions for distributed event processing are oriented toward static, topology-driven distributed solutions, which perform correlations close to the data sources. Although this solution reduces the amount of data to be propagated to remote nodes of the network, it has many disadvantages caused by its locality in computing. Due its large-scale remote computing nature cloud computing seems to be a very suitable structure for distributed complex event processing. The main advantage is the flexibility in the number and the processing power of cloud nodes, which enables the optimization of the realization of a complex event processing task. Indeed, depending on the current event flow capacity and the complexity of the event processing task, the processing can be split between several cloud nodes, a new node can be added, or even deleted. By using the publish/subscribe mechanism all these activities can be automated.

Distributed complex event processing approaches circumvent resource limitations by taking advantage of pre-existing (shared) or dedicated network infrastructure. The Padres system (Fidler et al. 2005) is a distributed approach relying on a pre-existing network of brokers. A broker may be an end-point for publishers (event sources) and subscribers (event sinks) to access the network. More importantly it is the task of the brokers to match events to the available subscriptions. The set of all interconnected brokers forms an overlay network across the underlying infrastructure. The overlay network is used to route events from publishers to subscribers along its links. In Padres complex subscriptions are possible which entail the combination of events from several streams. Depending on the subscription expression, these subscriptions typically match less than the cross product of the participating streams. Therefore, to save bandwidth on the network, these subscriptions are computed as close as possible to the event sources minimizing unnecessary network traffic (Li & Jacobsen 2005). Additionally, in Padres the overlay network may be dynamically reconfigured to remove network congestion and subscribers may be relocated to other brokers in order to mediate broker overload. A drawback of Padres is the event pattern language used in its subscriptions. It is limited to handle only key-value maps for events and the set of event operators is very limited, for example, temporal relationships between events cannot be expressed declaratively but must be expressed on a timestamp-arithmetical level. We propose a richer event format (e.g. based on RDF) and an extensible set of operators to provide a more expressive pattern language.

4.3 Adaptive Query Processing

Another distributed approach is presented by (Xing, Zdonik & Hwang 2005). The approach is described as an application of dynamic task allocation incorporating dynamic migration. Many ideas are valid for our work. Compared to Padres, event operators are allowed to be explicitly moved between nodes as a first-class operation. Like Padres, compared to a cloud approach, there is a fixed number of pre-existing computing nodes. Task allocation from Xing et al. is also termed operator placement in works on adaptive query processing surveyed in (Babu & Bizarro 2005) which is valid for our work.

Adaptive query processing (Babu & Bizarro 2005) (Deshpande, Ives & Raman 2007) is concerned with dynamically optimizing a query during its execution. Historically, database queries were optimized once before their execution using heuristics and statistics about the data. However, long-running queries on the one hand and scenarios with little or no statistics on the other paved the way for optimization of queries at runtime. For CEP, adaptive query processing yields relevant algorithms where queries are long-running and where statistics about input streams might change dynamically and require re-optimization.

Another approach to a self-tuning event processing network is STAR (Jain et al. 2007). Like the approaches previously mentioned STAR assumes a fixed number of participating computing nodes. However, STAR is built to compute approximate results allowing it to spend an error budget when calculating aggregates in a distributed network. Distributing the error budget optimally across nodes is STAR's approach to load shedding. Unlike this and other approaches incorporating lossy techniques we view CEP as the processing of discrete events where each event has a potential impact which cannot be derived from an incomplete sampling of the stream.

4.4 Stream Reasoning

Combining Semantic Web Technologies with real-time updates is often referred to as Stream Reasoning e.g. in (Valle et al. 2009). Semantic Web Technologies such as description logics have produced good tools to reason about knowledge e.g., infer implicit knowledge from given facts, and provide methods to publish data widely on the Web. Such tools work very well on static facts. However, the semantics of streaming facts and rapid, real-time updates has not gained much attention in the Semantic Web community until recently.

Modern description logic reasoners can classify large and complex ontologies in reasonable, but noticeable, time. Researchers are trying to cope with the complexity of logical inference by excluding time-consuming features from the language.

Current efforts on OWL2 profiles are related to the two classical inference strategies: forward chaining and backward chaining. OWL2-RL (OWL2-RL 2009) focuses on optimizing forward chaining (materialization), by designing a subset of OWL2 that can be processed in polynomial time with respect to the size of the ontology. The expressive power of this OWL2-QL (OWL2-QL 2009) profile, based on the DL-Lite family of description logics (Calvanese et al. 2005), is instead quite limited as it contains the intersection of RDFS and OWL 2 DL. It is designed so that data that is stored in a standard relational database system can be queried by rewriting the query into an SQL query that is then answered by the RDBMS system.

In the context of sufficiently large datasets, another trend is represented by query optimization. In the LarKc project, (Quesada et al. 2010) and (Peikov et al. 2010) describe different approaches to the task of selecting ("priming") relevant nodes with respect to a query, with the aim of providing approximate solutions in a short time and as a way to reduce response time.

Existing works assume the knowledge base as being either static or slowly evolving. This can be quite problematic when both data and models evolve during time. In such cases, classical forward chaining reasoners require repeating the whole inference process discarding all the already derived data. The possibility to perform an incremental reasoning can then be considered a key feature when dealing with live data.

Current research efforts towards incremental reasoning are aiming at investigating incremental maintenance of materialized views in logic (see (Volz et al. 2005) and (Barbieri et al. 2010)), object-oriented (Kuno et al. 1998) and graph databases (Zhuge et al. 1998). Extensions of the well-known Rete algorithm for incremental rule-based reasoning have been proposed in (Fabret et al.1993) and (Berster et al. 2002), while (Cuenca-Grau et al. 2007) and (Parsia et al.2006) described ideas about how to support incremental reasoning in description logics. All of these methods operate incrementally and are thus suited for dealing with non-static information, but none of them is explicitly dedicated to process an (almost) continuous information flow with the recent information being more relevant.

C-SPARQL (Barbieri et al. 2010), Streaming SPARQL (Bolles et al. 2008) and Time-Annotated SPARQL (TA-SPARQL) (Rodriguez et al. 2009) are extensions of SPARQL designed to express temporal issues in SPARQL queries. The work in (Rodriguez et al. 2009) motivates the need for a semantic management of streaming data. Streaming data are represented in RDF format with the purpose of its exploitation in Semantic Web applications (semantically annotated data and reasoning services). For this purpose, they propose a Time-Annotated RDF model and syntax. However, the authors explicitly mention that continuous queries, as one typical requirement of streaming data management systems, are not considered in that work.

Continuous SPARQL (C-SPARQL) (Barbieri et al. 2010), on the other hand, is a language for continuous query processing and stream reasoning. It extends the SPARQL language by adding support for window and aggregation operations. C-SPARQL, however, does not provide event processing capabilities: after determining the set of currently valid RDF statements, classical reasoning on that RDF set is performed as if it were static. In particular, C-SPARQL offers no way of detecting occurrences of RDF triples in a specific temporal order. We strongly believe that additionally temporal relatedness between events (e.g., an event happened before another event) as defined in streaming database systems is required to capture more complex patterns over RDF streaming data. Additionally, in C-SPARQL queries are divided into static and dynamic parts. The static part is evaluated by a RDF triple storage, while a stream processing engine evaluates the dynamic part of the query. In such settings, these two parts act as "black boxes" and C-SPARQL cannot take advantage of a query pre-processing and optimizations over the unified (static and dynamic) data space.

Streaming Knowledge Bases (Walavalkar et al. 2008) is a reasoner dealing with streaming RDF triples and computation of RDFS closures with respect to an ontology. For instance, the reasoner can identify a triple from a stream having a subject that is an instance of a certain class (or any of its subclasses, defined in an ontology). The approach is based on TelegraphCQ (Chandrasekaran et al. 2003) that is an efficient data stream system. In order to speed up stream reasoning, the authors propose to pre-compute all inferences in advance, and to store them in a database. Although this is an interesting approach, we believe that stream reasoning demands both: on-the fly-inference capabilities, and scalability.

The work in (Bolles et al. 2008) introduces Streaming SPARQL. The approach is built on temporal relational algebra, and the authors provide an algorithm to transform SPARQL queries to that algebra. Similarly as in (Barbieri et al. 2010), the approach is lacking event processing capabilities i.e., detecting RDF triple sequences occurring in a specific order. The same holds for (Barbieri et al. 2010b) where the authors propose stream reasoning based on incremental maintenance of materializations. Streaming RDF triples (as they occur) trigger an inference procedure that maintains materializations. Although promising, it is not clear how this approach works for multiple queries with different time window definitions (currently, materializations are assumed to be maintained only for one query).

EP-SPARQL is proposed in (Anicic et al. 2011) as a SPARQL extension and execution model to remedy many of the abovementioned drawbacks. The execution model of EP-SPARQL is based on the logic-based CEP engine ETALIS (Anicic et al. 2009) and thus grounded on logic rules and has both effective event processing and inference capabilities over temporal and static knowledge. We plan to build on and extend EP-SPARQL in the context of PLAY.

4.5 CEP for Proactive Application Monitoring

Complex event processing (CEP) has been used in industrial process control and in business activities and is now finding its way into IT management solutions[8]. For example, network management rule engines received events from the infrastructure: router memory low, packets dropped, server CPU usage too high, database server performance alerts, etc. With no idea about what caused this sudden increase, IT operations might have been tempted to increase the Web site infrastructure capacity. If IT operations were using predictive analysis, it would detect an abnormal pattern and predict the impending crash of the Web site in time for IT operations to do something about it. But it would have fared no better in analyzing the true root cause of the problem. Using complex event processing adds the business event dimension, for example, that the company made its intention public at some point in time. Processing this business event with the infrastructure events tells IT operations why the traffic increased, infers that it is most probably a transient phenomenon, and therefore recommends a temporary increase in capacity. If a private cloud is implemented, it could even trigger the provisioning of this extra and temporary capacity.

This is the driving concept behind Business Transaction Performance (BTP): end-to-end monitoring of applications and processes that uses a complex event processing (CEP) engine capable of processing millions of rules per second. With a CEP in place, you can reduce your mean time to know (MTTK) that a problem has occurred, relying on this strategy to alert you when problems first occur and, when possible, to automatically fix these problems by adjusting conditions on the basis of pre-defined rules.

The power of process automation in combination with CEP enables customers to fully leverage the elasticity the cloud and virtualization offers. The intelligence to interpret (we call it "context-awareness") business indicators and IT events is needed, as well as the end-to-end view of the automated process to know which process-steps are coming next. The combination gives the patterns and ability to predict what action is required/needed (essentially, to do the right thing at the right time.) There is already a broad range of use cases for CEP and automation across multiple industries…e-commerce, financial services, utilities and more.

[8] http://blogs.forrester.com/jean_pierre_garbani/10-08-24-complex_event_processing_ and_it_automation

4.6 Cloud Computing for Real-Time Processing

Cloud computing is a paradigm for large-scale computing. The "cloud" takes a service-oriented approach in support of "everything-as-a-service" (XaaS) (Baran 2008). Virtualized physical resources, virtualized infrastructure, as well as virtualized middleware platforms and business applications are being provided and consumed as services (Lenk et al. 2009).

Cloud computing providers guarantee quality of service to various degrees. Amazon, for example, provides terms with an "Annual Uptime Percentage of at least 99.95% during the Service Year" and a penalty model based on service credits (Amazon SLA, n.d.). Other providers have similar terms, often only covering availability. For most current cloud offerings there is little or no mention of latency. Generally, cloud computing seems to focus more on data-intensive (throughput-oriented) computing and neglecting basic requirements for low latency type applications.

The authors of (Boniface et al 2010) focus on providing real-time guarantees in the cloud. Their goal is to enable interactive real-time applications on top of virtualized infrastructure in the cloud. To that end they extended a cloud hypervisor and the Linux-RT kernel with real-time features to enforce such guarantees.

Although approaches like this are not yet available in production, such extensions could further enhance the value of the cloud for event processing.

4.7 Role of Complex Event Processing in PLAY

The role of CEP is to detect complex events using events from various distributed sources such as federated SOAs.

The distributed Complex Event Processing engine will use virtualized computing resources to meet changing demand for event processing. Thereby, complex event-driven communication is provided for large, highly distributed and heterogeneous service systems.

The (D)CEP component will be comprised of a network of so-called event processing nodes (EPN). Each of these nodes provides a specific function, which is applied to incoming events. These functions include the whole spectrum of CEP functionality, such as filtering, accumulation, transformation, enrichment and pattern detection. To provide this rich functionality the (D)CEP component is built upon pre-existing work carried out at FZI. This work comprises the standalone CEP engine ETALIS which will be extended to be applicable to a distributed environment.

In such a distributed environment ETALIS is seen just as one reactive autonomous node capable to communicate with such other nodes. The whole network of reactive nodes then becomes a distributed event-driven inference system efficient to execute some actions independently, or in collaboration.

4.8 Discussion

The above mentioned solutions and many others for distributed CEP are oriented toward a static topology of the underlying physical network. Cloud computing strives to abstract from this. Due to the observance of pay-as-you-go resource pricing (Armbrust et al. 2009), cloud applications must try to adapt their underlying topology to the current workload to optimize cost.

Since this is early work in progress, we tried to derive design decisions involved in bringing Complex Event Processing (CEP) to the cloud. Combining cloud computing and CEP raises a number of new challenges presented in this chapter but also allows advancements over the state of the art in distributed CEP which would not be possible without the benefits provided by existing cloud infrastructure. Future work is on the one hand to further enhance the RDF capabilities of our event processor, on the other hand to start implementing a first distributed prototype to gain experimental performance data about the quality of different operator placement strategies in a real cloud environment.

5 State of the Art on Interactions between People, Services and Events

5.1 Introduction

WP4 aims to enhance the user's experience when interacting with a service-based application or system using contextual information and adapt the system behaviour to it. Within PLAY, events can be used as a source of context, since they are snippets of the past activities; therefore event processing results will be transferred to WP4 tools, injecting context related information to the service-based application or system. To achieve the aforementioned goal, an event-based context model is needed along with context monitoring mechanisms for being able to recommend and perform the appropriate service adaptations as means for successfully reacting to dynamic environmental changes.

Adaptation as one of the basic phenomena of biology is the evolutionary process whereby a population becomes better suited to its habitat (Martin, 2010), (Williams, 1966). The notion of adaptation has been extensively used, especially nowadays, in the computer science domain. It is considered as one of the most desired functionalities of today's highly dynamic, distributed and ubiquitous environments in the service-oriented setting. The need for highly flexible services that can be orchestrated in order to provide certain behaviours and at the same are able to react and adapt inside their "habitat" (i.e. change based on the context and events that formulate the imprinting of a highly dynamic service environment), is considered to be a desired but difficult to achieve fact.

S-Cube as the most prominent Network of Excellence[9] in service adaptation, points out the evolution and adaptation methods and tools as keys to enable service-based applications (SBAs). Following the S-Cube's terminology, the term evolution is considered to be the more traditional modification of a system's requirements, specification, models and execution during design time (e.g. re-design and/or the re-engineering of the application modifying it permanently), while adaptation refers to the modification of a specific instance of a system during run-time (e.g. re-execution of a unavailable service or a substitution of a unsuitable service). In (S-Cube, 2009) an adaptation taxonomy was introduced for defining Why, Who, What, and How software adaptation takes place. The first dimension of this taxonomy defines the usage of the adaptation process and corresponds to: Perfective Adaptation (i.e. improve the application even it runs correctly), Corrective Adaptation (i.e. remove the faulty behaviour of a SBA by replacing it by a new version that provides the same functionality), Adaptive Adaptation (i.e. modify the application in response to changes affecting its environment), Preventive Adaptation (i.e. prevent future faults or extra-functional issues before they occur) and Extending Adaptation (i.e. extend the application by adding new needed functionalities). The Who dimension characterizes the adaptation problem from the view of the different actors (software or human) involved in the adaptation process (i.e. Adaptation Requestor, Adaptation Designer, Adaptation Initiator and Adaptation Executor) by considering What the Subject of adaptation is (i.e. SBA instance, class, context and mechanisms), the Adaptation Aspect (e.g. quality model, functionality, HCI aspects etc.), and the adaptation scope (i.e. temporary vs. permanent adaptation). Finally, the How dimension of the adaptation can be achieved by defining specific adaptation strategies. According to Bucchiarone et al. (2009) these strategies are: Service substitution, Re-execution, (Re-)negotiation, (Re-)composition, Compensation, Trigger evolution (i.e. insertion of workflow exception able to activate the application evolution), Log/updated adaptation information, Fail. In addition, they refer to the diversity of specific adaptation needs and of factors the adaptation strategies depend on (Bucchiarone et al., 2009). According to this distinction, the following design approaches have been defined in the work of (Bucchiarone et al., 2009):

- Built-in adaptation: adaptation needs and possible adaptation configurations are fixed and known a priori (i.e. design time).
- Abstraction-based adaptation: adaptation needs are fixed, but the possible configurations in which adaptation is triggered, are not known a priori. In such a case, a typical pattern according to (Bucchiarone et al., 2009), (S-Cube, 2008) is to define an abstract model of an SBA and a generic adaptation strategy, which are then made concrete at deployment/run-time.
- Dynamic adaptation: adaptation needs that may occur at runtime are not known at design time. In such a case, it is necessary to provide specific mechanisms that select and instantiate adaptation strategies depending on a specific trigger and situation.

In the following sections of PLAY's state of the art, we give an overview of well-known research efforts that present techniques, models and integrated solutions that cover the two aspects of service adaptation.

[9] http://www.s-cube-network.eu/

These two aspects are: (i) under which circumstances an adaptation should take place (i.e. triggers of adaptation) and (ii) how such an adaptation should happen (i.e. implementation of service adaptation).

5.2 Triggers of Adaptations in Service-Based Applications

5.2.1 Context-awareness for Triggering Adaptations

Gartner research analysts define "context-aware computing" as the concept of leveraging information about the end user to improve the quality of the interaction. They even make claims such as that context will be as influential to future consumer services and relationships as search is to the Web; see e.g. Alvarez and Clark (2010). Moreover, they predict that contextual systems of the future will deliver highly personalized relevant information and services with minimum effort from the user; see Jones (2010). Achieving this goal will require increasingly smart and automated context, which implies that context will evolve from reactive to proactive interactions, it will use personal knowledge-based on both deductions and information volunteered by the user and that contextual services will become more socially aware and more tightly integrated with social networking tools.

What is particularly relevant to PLAY – due to its implications for complex event processing – is what Gartner analysts call Context Delivery Architecture (CoDA), which they consider as the next step in the evolution of Service-Oriented Architecture (SOA); see Natis et al (2007), Clark and Lapkin (2008) and Clark (2010). In CoDA the functioning of software elements (services or event handlers) is determined not only by the input to the element, but also by the secondary sources of information – the context; two invocations of the same service with the same parameters may yield different results in different circumstances, i.e. within different contexts. Like SOA, CoDA introduces the concept of creating composite applications through reusable services, and in addition aims to enhance user's experience using the knowledge of context and adapting the application behaviour to it.

Another helpful background for CoDA is Event Driven Architecture (EDA). Events can be viewed as a source of context, since they are snippets of the past activities; therefore, event processing may in some cases be viewed as a context detecting technology. Event processing results may be transferred to other applications, injecting context related information into services and processes; see Etzion et al (2010) for an overview of the utilization of context in event-based systems.

Context-awareness in service-oriented systems, refers to the capability of a service or service-based application to be aware of its physical environment or situation and to respond proactively and intelligently based on such awareness; see e.g. Abowd et al (2002). Through the use of context, a new generation of service-based applications is expected to arise for the benefit of coping with the dynamic nature of the Internet; see e.g. Sheng et al (2008) and Sheng, Yu and Dustdar (2010). The multiplicity of applications and the surge of research activities in context-aware service systems are also evident in recent survey research; see e.g. Truong and Dustdar (2009), Hong et al (2009) and Kapitsaki et al (2009).

We believe that there three research areas of interest for PLAY. The first concerns the definition and modelling of context. The second refers to the provision of context information and the detection of context changes, while the third concerns the mechanisms that can be used to adapt an application's behaviour based on context information without explicit user intervention, i.e. how to use context information to achieve context-awareness of services.

In the following subsections we briefly review the literature concerning the first two topics, while we discuss the third one in section 5.3.2.

5.2.1.1 Context Definition and Modelling

The most widely used definition of context in computer science is the one by Dey and Abowd (2000) and Dey (2001) who define context as:

> "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves".

Abowd and Mynatt (2000) have further pointed out that context should include the 'five W': Who, What, Where, When, and Why. For example, by 'Who', they mean that it is not enough to identify a person as a customer; the person's past actions and service related background should also be identified for better service provision. 'What' refers to the activities conducted by the people involved in the context and interactions between them. 'Where' represents location data. 'When' is related to time. 'Why' specifies the reason for 'Who' did 'What'. 'Why' represents a complicated notion and acts as the driving force for context sensitive information systems.

Furthermore, as Bucchiarone et al (2010) point out, in the case of service-based applications, context has various different facets as it includes information ranging from the situation in which users exploit a service-based application to the conditions under which the component services can be exploited. The identification of these facets depends on the specific application and should be performed very early. Actually, context modelling should start from the requirement analysis phase where, in parallel to the precise definition of requirements, a proper context model has to be defined.

This context model will then form the basis for the definition of those situations that may trigger the adaptation or evolution of a service-based application, and, at runtime, it will enable the identification and the collection of the appropriate context information.

In the literature there is a plethora of context models. For example Strang and Linnhoff-Popien (2004) and Bettini et al (2010) review models of context that range from key-value models, to mark-up schemes, graphical models, object-oriented models, logic-based models and ontology-based models.

These model categories will be examined in detail within WP4 of PLAY in order to derive directions for development work. An interesting context model is the one proposed by Tanca et al (2009), which was initially developed for mobile devices and later extended for use in service-based applications in Bucchiarone et al (2010). Another example is the one by Truong et al (2009) who develop an ontological model of the W4H classification for context. The W4H ontology provides a set of general classes, properties, and relations exploiting the five semantic dimensions: identity (who), location (where), time (when), activity (what) and device profiles (how); see de Neto et al. (2005). Truong et al (2009) exploit the concepts of the W4H ontology by including domain-independent common context concepts from existing work; e.g. FOAF, vCard, the OWL-Time Ontology, etc.

Another interesting issue for the design of context-aware applications concerns modelling languages, which take context explicitly into account. The first such effort was ContextUMLa UML-based modeling language which was specifically designed for context-aware Web service development and applies model-driven development principles; see Sheng and Benatallah (2005). In a Web-service-based environment ContextUML considers that context contains any information that can be used by a Web service to adjust its execution and output. Examples of contexts in ContextUML are: i) contexts related to a service requester (mostly it is the client who invokes a service), including the requester's identification information, personal preferences, current situation (e.g., location), and other information (e.g., friends list, calendar);ii) contexts related to a Web service, such as service location, service status (e.g., available, busy), and its QoS attributes (e.g., price, reliability); and iii) other contexts like time and weather information. ContextUML has been adopted for the development of a model-driven platform, called ContextServ, which is used to develop context-aware Web applications; see Sheng et al (2010).

Finally, it is interesting to note here two modelling efforts that attempt to treat service logic and context handling as separate concerns: the first if the work of Prezerakos et al (2007) who modified ContextUML using Aspect-Oriented Programming (AOP) principles the second is by Grassi and Sindico (2007) who leveraged ideas from model driven development (MDD) and AOP in order to define a conceptual context model and then mapped these ideas to a UML framework.

5.2.1.2 Detecting Context

Modelling and monitoring the events that trigger adaptation is a critical task once the context model and its relation with the service-based application are defined. Specifically, it is important to define the contextual changes which are critical for the functioning of the application and what should be done when these changes take place; Bucchiarone et al (2010).

Context monitors may take application-specific forms and in some cases the adaptation trigger may not correspond to some value of a particular single context element, but may be characterized by the complex combination of different context dimensions. Such situations may be directly encoded using the capabilities provided by the existing monitoring frameworks, such as Dynamo; see Baresi and Guinea (2005) and Baresi et al (2007). Dynamo augments BPEL technology with self-healing capabilities with the use of run-time monitoring, to check whether the system behaves correctly and whether functional and non-functional expectations are met, and reaction strategies, to recover from erroneous situations. This research effort develops and uses WSCoL, a constraint language for the supervision of BPEL processes; monitoring information is specified as assertions on the BPEL code and the overall approach uses Aspect-Oriented Programming (AOP) to check the assertions at runtime; see Baresi and Guinea (2010).

Another approach is by Wu et al. (2008), who propose an AOP-based approach for identifying patterns in BPEL processes and use a stateful aspect extension allowing the dentition of behaviour patterns that when identified they can trigger different actions. It also allows the monitoring of certain patterns by using history-based pointcuts.

Our intention within PLAY (in the work of Task 4.1 entitled "Event-based Context Detection") is to experiment with context changes which correspond to either atomic or complex events and use Complex Event Processing approaches (CEP) to model and identify them. Hence, recent work on applying event-driven monitoring is quite relevant.

5.2.2 Monitoring Services for Triggering Adaptation

As organizations increasingly rely on the flexibility offered by service-based applications (SBAs), in order to operate and survive in a highly dynamic world, the level and quality of service provisioning, as well as requirements keep on changing and evolving. In order to detect events and situations that necessitate an adaptation (to keep high quality) of a service-based application, the majority of adaptation approaches from the service-oriented computing field resorts to exploiting monitoring techniques. But the problem is that orchestration (e.g. BPEL) standards do not provide any means for monitoring running processes and it is up to each orchestration engine to provide such monitoring interfaces. The monitoring of QoS is a necessary foundation to decide whether or not a replacement of a service should be performed and it cannot be considered as an easy task, especially in cases where the orchestration of highly distributed services is needed. Monitoring provides a way to collect and report relevant information about the execution and evolution of a service-based application. Depending on the goal of a particular adaptation approach, different kinds of events are monitored and different techniques are used for this purpose.

In (Moser et al., 2008), authors present the VieDAME system, which allows monitoring of BPEL processes according to Quality of Service (QoS) attributes and replacement of existing partner services based on various (pluggable) replacement strategies. Details of their work regarding how they implement the service adaptation are discussed in section 5.3.1. We focus in this section on their proposed monitoring functionalities. They use statistical monitoring data about previous service invocations, calculating the overall execution time of each BPEL process. The VieDAME system provides this information for optimizing the business process in terms of partner service invocations. They have developed a dedicated monitoring component which is responsible for storing these partner service invocation events (i.e., SOAP calls that result from <invoke> activities in the BPEL process), that in turn are aggregated to calculate performance related QoS values such as average response time, accuracy or availability of some particular operation.

In many others approaches (Baresi et al., 2004; 2007; Erradi et al., 2006) the events that trigger the adaptation are failures. These failures include typical problems such as application exceptions, network problems and service unavailability (Baresi et al., 2004), as well as the violation of expected properties and requirements. Baresi et al. (Baresi et al., 2007) present a solution to self-healing BPEL compositions called Dynamo, which is an assertion-based solution that provides special purpose languages (WSCoL and WSReL) for defining monitoring and recovery activities. The assertions used (pre-conditions, post-conditions, and invariants), define constraints on the functional and quality of service (QoS) parameters of the service composition and its context. Spanoudakis et al. (2005) use properties in the form of complex behavioural requirements expressed in event calculus. The central component of the architecture is the SCS (service centric system) manager, which is responsible for overseeing the monitoring of functional, and quality of service requirements and making decisions on whether to issue requests for discovering candidate services that should replace unavailable or malfunctioning services. The specifications of the violated requirements detected by the monitoring component are used to generate queries for discovering services that could substitute the failed services.

Erradi at al. (2006) express expected properties as policies on the QoS parameters in the form of event-condition-action (ECA) rules. When a deviation from the expected QoS parameters is detected, the adaptation is initiated and the application is modified. A dedicated monitoring service is used, that continuously monitors interactions with the participating services to verify that the configured monitoring policies are being satisfied and to detect any condition changes such as faults. The monitoring policies specify the desired behaviour of the system in terms of pre-conditions/post-conditions that express constraints over exchanged messages and thresholds over QoS guarantees (e.g. service response time) as stipulated in pre-established Service Level Agreements (SLAs). This monitoring service listens to fault messages returned by invoked services as specified in their WSDL interface. In a similar approach Siljee et al. (2005) propose the DySOA (Dynamic Service-Oriented Architecture), an architecture that extends service-centric applications to make them self-adaptive. They use monitoring to track and collect the information regarding a set of predefined QoS parameters (response time, failure rates, availability) infrastructure characteristics (load, bandwidth) and even context. The collected information is checked against expected values defined as functions of the above parameters, and in case of a deviation, the reconfiguration of the application is triggered.

It is true that all these monitoring approaches follow the reactive approach to adaptation, i.e., the modification of the application takes place after the critical event happened or a problem occurred. We can understand that once this adaptation happens, many process instances might have already been executed in a "wrong" mode. This is something that can be prevented by using online testing techniques that are presented in the next section.

5.2.3 Online and Regression Testing for Triggering Adaptation

The dynamic nature of service-oriented architectures poses new challenges to system validation and testing. At the present time, managing and services over large distributed systems together with the growing complexity of underlying technologies has become a challenge. Faults can occur anywhere in a distributed system. Techniques that address the evaluation of the services' behaviour over time are considered critical. Online testing has been used as an important tool that uncovers failures and deviations of the quality of services from the expected one, by systematically executing services or service-based applications (service compositions- SBA's). Existing approaches for testing service-based applications mostly focus on testing during design time, which is similar to testing of traditional software systems (Gehlert et al., 2011).

In Wang et al. (2004) the authors stress the importance of online testing of Web-based applications, argue on its special needs in comparison to offline testing at design time and propose methods to reduce the interference to the normal services of system and improve the efficiency of online testing. The authors, furthermore, see monitoring information as a basis for online testing. Deussen et al. (2003) propose an online validation platform with an online testing component. They use TTCN-3 (Testing and Test Control Notation Version 3[10]) in order to implement their system in Active Network environments.

A metamorphic online testing is proposed by Chan et al. (2007), which uses oracles created during offline testing for online testing. Specifically, they determine, during the offline testing, a set of successful test cases to construct their corresponding follow-up test cases for the online testing. These test cases are executed by metamorphic services that encapsulate the services under test as well as the implementations of metamorphic relations. Thus, any failure revealed by the metamorphic testing approach will be due to the failures in the online testing mode.

Bai et al. (2007) propose an adaptive testing framework in, which can continuously learn and improve the built-in test strategies, using a windowing mechanism. Tests are executed during the operation of the service-based application and can be adapted to changes of the application's environment or of the application itself.

Another approach related to online testing is the regression testing, which aims at checking whether changes of a system negatively affect the existing functionality of that system. The typical process is to re-run previously executed test cases. Ruth et al. (2007) reports a framework that makes it possible to carry out safe regression test selection for verification of Web services in an end-to-end manner, by using the Safe RTS technique. Di Penta et al. (2007) also propose a regression test technique for Web services to discover if functional and non-functional expectations are maintained over time. However, none of the techniques addresses how to use test results for the adaptation of service-based applications.

The combination of monitoring and testing for validating service-based applications was examined in (Canfora and di Penta, 2006). However, all these approaches do not exploit testing results for (self-) adaptation in comparison to (Gehlert et al., 2011). According to Gehlert et al. (2011), Service-based Applications (SBAs) can be dynamically adapted to address various goals, which may include: (1) aiming to better achieve the users' requirements named as "perfective adaptation", and (2) repairing and preventing failures named as "corrective adaptation". This is the first work that takes into account the interplay of both of these different adaptation goals and tries to avoid conflicting situations. They introduce a framework to integrate and align perfective and corrective adaptations, and use requirements engineering techniques to trigger the former adaptation and online testing techniques to trigger the latter one. In terms of requirements engineering for perfective adaptation, they try to ensure that the service provides the functionality needed by the SBA and if new services appear to be better than the services already used in the SBA to adapt the workflow by replacing them. In order to accomplish that, they use the Tropos approach (Bresciani et al., 2004; Castro et al., 2002), which has a formalisation that allows to reason about goal satisfaction in a goal model, and detect in such way the desired adaptation. In terms of online testing for corrective adaptation, they propose and use the PROSA framework (PRO-active Self-Adaptation) (Hielscher et al., 2008). PROSA exploits online testing solutions (at run-time) to pro-actively trigger adaptations. The major activities that PROSA performs are: test initiation, test case generation/selection, test execution and adaptation detection.

A disadvantage of their work is that they constrain it by considering the exchange of individual services as the only mechanism for performing adaptations of SBAs. So, SBAs adaptations take place by changing the bindings of services (or end points) to the workflow, using dynamic binding. The modification of the control or data flow structure is not addressed in their work, but they consider whether all or some running instances should benefit from these modifications (instance migration) or whether only future instances should use the described modifications.

5.3 Event-based Service Adaptation

The modern dynamic business environments create the need for continuously adapting business processes, by keeping a competitive level and adequate quality of service (QoS). The desired and optimal

[10] http://www.ttcn-3.org/

changes in business processes are considered to be the automatic context –aware adaptations that do not require the redeployment of processes. Critical challenges were defined in (Hermosillo et al., 2010b), regarding adaptation, as the use of traditional orchestration engines (e.g. BPEL engines) lead to static process definitions incapable for adaptations without their costly redeployment that generates downtime for systems and loss of information about on-going transactions. The only changes possible at runtime are the bindings to partner links, but they have to be previously defined at deploy-time (Juric, 2006). Also, in the context of composite Web services several kinds of changes and faults may arise (partner services may go down, services may be updated to require new policies, etc.) after deploying such a service. Most available Web service orchestration engines do not provide automated support for detecting and reacting to such situations and handling them can only be done through manual human intervention. Such an approach is considered inappropriate because the operation of the composite service is discontinued, certain processes may be interrupted in the middle of a business transaction, and also because of the huge administrative overhead.

5.3.1 Using Aspect-Oriented Techniques for Service Adaptation

Nowadays, several efforts that try to cope exactly with this issue point to Aspect-Oriented Programming (AOP), as a novel way to weave alternative actions in business processes at run-time. AOP has been proposed as a technique for improving the separation of concerns in software systems and for adding crosscutting functionalities without changing the business logic of the software. The domain of AOP appeared in 1997 (Kiczales et al., 1997). It was pioneered by Gregor Kiczales (Kiczales et al., 2001), and his team at the Xerox Palo Alto Research Center. While original and innovative, the domain of AOP inherits results from other programming approaches, such as reflection, open implementations, meta-object protocols, and generative programming. AOP, as a new programming paradigm, introduces notions such as aspect, join point, pointcut and advice code. However, these notions do not replace but complement existing ones, such as class, object, procedure or method. Furthermore, these notions are not specific to a programming style (e.g., object-oriented or procedural) or a given syntax (Java, C#, Ada, COBOL, etc.). Today, aspect-oriented extensions exist for many languages, object-oriented or procedural.

One of the most recognizable approaches for service adaptations using AOP is the work in (Charfi and Mezini, 2007) where the AO4BPEL is introduced. AO4BPEL is an XML-based language that creates a wrapper around the BPEL and has the ability to weave aspects at runtime to business processes. Aspects consist of one or several pointcuts and advices. AO4BPEL is based on XPath[11], which is used to select activity join points (i.e., points corresponding to the execution of activities) and internal join points (i.e., points inside the execution of activities such as the point where the outgoing message of an invoke activity is generated). An advice is the new behaviour to be included at a join point and contains the new code to be executed. Special constructs may be used inside the advice to access the input/output data of the join points, the respective SOAP messages, as well as reflective information. AO4BPEL supports only the manual activation and deactivation of aspects via the administrator interface.

Nevertheless, the advantage of using AO4BPEL is that the business process specifications can change at runtime without the need to redeploy them and lose all the on-going transactions. AO4BPEL handles two types of aspect deployments: *process-level* deployment and *instance-level* deployment (Charfi and Mezini, 2007). The former is used when the aspect is needed in all the instances of the business process, while the latter is used when only specific kind of instances are targeted. The process can be adapted using three kinds of advices: before advice, after advice and around advice. The first one intercepts the call before the join point, executes its task and then lets the process continue the normal flow, including the execution of the join point activity. The after advice gets executed just after the join point activity is completed and the process continues normally afterwards. Finally, the around advice intercepts the call just like the before advice, but it adds functionality before and after the join point activity. Moreover, this advice can be used without executing the join point activity.

Charfi et al. (2009) use AO4BPEL and present a plug-in-based architecture for self-adaptive processes. They propose different plug-ins (implicit/explicit) with a well-defined objective, which realize the sense-think-act paradigm. Each plug-in will have two types of aspects: the monitoring aspects that will check the system to observe when an adaptation is needed and the adaptation aspects that will handle the situations detected by the monitoring aspects. These monitoring aspects can be hot-deployed to their BPEL engine and whenever their conditions are met, it uses AO4BPEL to weave the adaptation aspects into the process at runtime. (Charfi et al., 2009) have extended AO4BPEL with special support for dynamic plug-ins through generating, activating, and deactivating aspects at runtime. Since, initially AO4BPEL (Charfi et al., 2009) supported only the manual activation and deactivation of aspects via the administrator interface, they tried to extend AO4BPEL with new activities for activating and deactivating aspects from an advice of another aspect. Their proposed overall architecture presents an (AO4BPEL) orchestration engine, extended by special self-adaptation plug-ins using two kinds of extension points: implicit and explicit. Every process

[11] http://www.w3.org/TR/xpath/

activity is an implicit extension point where the plug-in can execute adaption logic. At the explicit extension points, the architecture may be extended with Web services that are provided by the self-adaptation plug-in (e.g., for monitoring or for diagnosis so that the plug-in can decide if adaptation is needed). Each plug-in follows a well-defined objective (e.g., selfhealing). It consists of several aspects and infrastructural services and is developed by domain experts, e.g., an administrator and deployed to the orchestration engine at runtime through an administrator console. Inside the plug-ins, two types of aspects are used: monitoring aspects, which collect information and decide based on it whether adaptation is needed and adaptation aspects, which handle the erroneous situations and events detected by the monitoring aspects. The monitoring aspects are able to activate and deactivate the adaptation aspects at runtime.

The development process of self-adaptation plug-ins, consists of three steps according to the sense-think-and-act paradigm. A developer, who analyses the application for concrete problem scenarios, implements the sense part using a pointcut. The advice implementing the think part identifies critical situations by following one of three possible strategies: 1) pre-examination, 2) post-examination, and 3) wrapping. Pre-examination proactively ensures that certain process activities will not fail or will be replaced by alternative activities. Post-examination is performed after the monitored activities are executed. Wrapping ensures that a process activity will not fail by encapsulating it into a wrapping activity (e.g., a scope with a fault handler) that captures problematic situations when the original activity is executed. Wrapping is used when we cannot decide via pre-examination if the activity will fail or not. Finally, the developer implements the act part in form of an adaptation aspect, which uses pointcut-and-advice to change running processes.

Another approach, which considers aspects in BPEL but doesn't try to extend this orchestration language, is the work of Karastoyanova and Leymann, (2009). The BPEL'n'Aspects approach is not restricted to only BPEL code for the advice implementations, but rather allows for the use of any Web services (WS). Additionally, they avoid extending BPEL in order to claim reuse of legacy BPEL processes. In their approach, they combine standard BPEL, with the publish/subscribe paradigm and WS-Policy so that WS operations play the role of aspects with respect to BPEL processes. They presented the syntax for such aspects as an extension of the WS-Policy framework and introduced the architecture of the supporting infrastructure. In other words, they try to introduce flexibility and adaptation in a non-intrusive way with respect to existing technologies and infrastructure in order to gain acceptance. Their basic idea is to surface events occurring during navigation through a BPEL process model. These events signal that the BPEL engine has reached an event of interest (or a join point) and so, operations of services may be registered as subscribers to such events. Whenever an event happens, the operation of the registered service will be invoked - this is in fact the weaving.

The infrastructure for the BPEL'n'Aspects approach comprises four major components: a *BPEL engine*, a *Service Bus* (Leymann, 2005) for delegating the invocation of services that implement process activities, a *Broker* with "weaving" role of aspects into BPEL processes and an *Aspect Management Tool* for creating, editing, deleting, deploying and undeploying aspects. To enable the communication between the broker and the orchestration engine, they have augmented the open-source ActiveBPEL [12] engine (Khalaf, et al., 2007) with a component, called controller, which is responsible for notifying life cycle events about process instance constructs. They have used the aspect-oriented programming language, AspectJ[13] for developing aspects and the ActiveMQ [14] for the pub/sub middleware implementation.

Recently, in (Hermosillo et al., 2010b) another AOP approach was proposed as a framework named CEVICHE (Complex Event processIng for Context-adaptive processes in pervasive and Heterogeneous Environments). The purpose of CEVICHE is to create context-aware business processes that are able to adapt dynamically in order to respond to different scenarios. Their main focus is to provide a mechanism for automatic adaptations in order to maintain high QoS (in terms of service performance and service availability) for business processes. CEVICHE relies on the BPEL extension AO4BPEL (Charfi and Mezini, 2007) and on the use of CEP engines for detecting situations that need adaptation. For that, they defined a language called the Standard Business Process Language (SBPL), which gathers in an XML file, all the information about the processes, contextual environment, business rules, and adaptation conditions.

CEVICHE is composed of three main parts: a user interface to create SBPL files, a translation framework to manage the plug-ins for the chosen CEP engine, and an aspect manager to deal with the process adaptation. The information in the SBPL file is sent to the translation framework, which separates the data in three parts: the business process (BPEL), the adaptation situations (CEP rules) and the aspects to adapt the process. In such a way they define the part of the business process, it needs to perform the adaptation (the pointcut). Using the pointcuts and the AOP framework, they define what special behaviour (the advice code) must be applied in that part of the business process at run-time. Once the advice code is

[12] www.activebpel.org

[13] http://www.eclipse.org/aspectj/

[14] http://activemq.apache.org/

woven, the process is adapted. Through CEP they analyse events and their context information, in order to automatically decide when and how to adapt the system.

Morin et al. (2008) proposed an approach that leverages Aspect-Oriented Modelling (AOM) and Model Driven Engineering (MDE) in order to manage variability and adaptation on the architecture of running systems. Their work relies on the notion of aspect models that can be woven into an explicit model of the runtime configuration seating on top of the running system. They argue that the high-variability of features in Dynamic Adaptive Systems (DAS) introduces an explosion of possible runtime system configurations (often called modes) and mode transitions. Actual mode transitions between runtime configurations are based on the differences between the initial model and the newly woven one. Morin et al. (2009) extended their initial work, in order to show how aspects can help designers determine interactions between dynamic variants and how runtime models can be used to validate new configurations on the fly, before actually adapting the running system. They focus on variation points and variants (represented by aspects) instead of focusing on whole configurations. A variability dimension is a particular concern that may be realized in different ways. They use SmartAdapters (Lahire et al., 2007) an AOM tool for weaving aspects at a model level. However, the approach presented in their paper is not dependent from SmartAdapters and other AOM tools like MATA (Jayaraman et al., 2007) can also be used. They (optionally) extend each aspect with a context describing when to trigger the weaving of aspects. A context is a slice of the environment describing when the aspect is useful and its impact on QoS properties. Aspects with no context can be manually triggered by the user. Once a target model (representing the system they want to reach) is created and validated, it is compared with the source model (representing the actual architecture of the running system). They used EMF Compare[15] in order to compare models and obtain the relevant changes between the source model and the target model e.g., addition/removal of components/bindings, changes of attribute values, etc.

As in (Morin et al., 2009), they focus on the validation of target configurations, they define some invariants on the metamodel that they use to represent runtime architecture and check these invariants for every constructed (by aspect weaving) target configuration. These invariants are expressed as Kermeta (Muller et al., 2005) meta-aspects that are woven into the metamodel. Kermeta meta-aspects can be used to refine existing meta-classes by integrating contracts (pre/post-conditions, invariants), attributes and references, operations and super-classes. All the models (meta-model, configurations and aspects) can be serialized in XML and transmitted to other systems, or can be used for simulations.

The authors in (Sánchez and Villalobos, 2008) use an aspect-oriented approach introducing concern separation and instrumentation. Their goal is to build extensible workflow applications using synchronized executable models. They use open objects, which are representations of the state of the elements in the model, to monitor the invocation of services and adapt the process by weaving the interaction with other models before (activation) and after (deactivation) the call to the service.

Another aspect-oriented implementation, can be found in the work of (Rahman et al., 2008), where they propose an adaptable ECA centric architecture and implementation mechanism based on service-oriented computing and aspect-oriented programming for rule-based enterprise information systems. They use contracts to assign Web services to instances of execution calls. They achieve adaptation by changing the contract at runtime and assigning new Web services to a call.

In (Baresi et al., 2007), a design process model was introduced, for the definition of supervised BPEL processes, in which supervision rules are automatically generated starting from the policies that characterize external services. These policies exploit WSCoL as a language for describing constraints on the messages exchanged with the business process. WS-Policy policies are specified in a deployment descriptor and transformed to supervision rules that are automatically enforced by an AOP-based framework implemented with AspectJ on top of ActiveBPEL. Adaptation at runtime is not supported as AspectJ uses static weaving.

In (Moser et al., 2008), Moser et al. present the VieDAME environment, which is an extension to the ActiveBPEL engine that allows the monitoring of BPEL processes (as already mentioned in section 5.2.2) according to Quality of Service (QoS) attributes and replacement of existing partner services based on various (pluggable) replacement strategies. The chosen replacement services can be syntactically or semantically equivalent to the BPEL interface. Services can be automatically replaced at runtime without any downtime of the overall system, using an aspect-oriented approach by intercepting SOAP messages and allowing services to be exchanged during runtime with little performance penalty costs. The main idea for VieDAME was to achieve non-intrusive behaviour with regard to dynamic service adaptation, which enables the runtime exchange of partner links within a BPEL process, without any changes to the BPEL

[15] http://www.eclipse.org/modeling/emft/

process or the involved partner services. Currently, VieDAME environment supports ActiveBPEL 3.0[16] and Apache ODE[17].

In (Kongdenfha et al.,2006) a framework for service adaptation was introduced that consists of a taxonomy of the different possible types of mismatch between external specification and service implementation, a repository of aspect-based templates that automate the task of handling mismatches, and a tool that supports template instantiation and their execution together with the service implementation. This approach requires BPEL process modification when aspects are woven into the base system code, something which implies that services have to be statically bound to invokers.

Finally David and Ledoux (2006) present SAFRAN (Self- Adaptive FRActal compoNents) an Aspect-Oriented approach and extension of the Fractal component model (Bruneton et al., 2006) for the development of the adaptation aspect as reactive adaptation policies. These policies detect the evolutions of the execution context and adapt the base program by reconfiguring it.

5.3.2 Context-driven Service Adaptation

Context-driven service adaptation is the topic of several research works in the literature which address both service design and implementation issues.

Santos et al (2007) proposed a service-oriented middleware for context-aware applications which integrates two components in order to provide context-aware capability: the Context Management Service (containing a set of well-defined component interfaces and a data model for representing context), and the Awareness and Notification Service (offering a rule-based facility which contains the context-based conditions and receives notification when the specified context holds).

More recently, Dai and Liu (2010) build on the modeling work of Grassi and Sindico (2007) mentioned previously in order to develop an adaptive and responsive service framework that integrates context-aware techniques with OSGi-based open source service-oriented platform.

In a different direction, Moltchanov et al (2009) report their work in the C-Cast IST project towards providing an end-to-end system design for context-aware services and content provisioning. They present a framework for modular multi domain context detection and show how required context can be obtained by acquiring a variety of source data and applying reasoning mechanisms for aggregation. The raw data can originate from virtual, logical or physical sensors, covering the physical environment of the user, profile information and other databases (e.g. calendar, address book). Based on the derived context, content or service provisioning can be adapted in order to reduce the need for explicit feedback and to increase overall user experience.

Catarci et al (2008) developed within the European IST project Workpad an adaptive process management system that uses the W4H context ontology mentioned above. Their adaptive process system employs process schemas in which each task has an associated set of conditions that must be true to perform the task. A context service monitors internal and external conditions and adapts the process to deal with discrepancies.

In order to allow programmers to reason at a higher semantic level a number of domain-specific languages have emerged; e.g. Munnelly et al (2007) propose a language that provides higher-level constructs for expressing the adaptation of application behaviour due to a change in context. In a middleware-based approach David and Ledoux (2005) present a the Wildcat framework for self-adaptive components that follows the Event-Condition-Action (ECA) pattern and makes use of events emitted by a context-aware service that provides information about the execution context of the application.

Besides its extensive use in context-aware mobile and pervasive systems, AOP has also been applied in research for service adaptation and for service compositions.

An interesting approach is the one by Boukadi et al (2008 and 2009) who propose an architecture for a high-level structure called Service Domain which orchestrates a set a of related IT services based on a BPEL specification. The Service Domain concept was developed to enhance the use of Web services for e-business collaborations and is context-aware (context-awareness is guaranteed by enhancing BPEL execution using AOP). The Service Domain is implemented as a node consisting of an Entry Module, a Context Manager Module, a Service Orchestration Module and an Aspect Activator Module. Concerning the context of the Service Domain Boukadi et al (2008) propose a context ontology modeled in OWL. Their adaptation approach is a three-step process consisting of: (a) context detection, i.e. checking the runtime context information in order to detect possible context changes (performed by the Context Manager Service which is developed as a Web service in the BPEL process); (b) aspect activation, which is

[16] http://www.activebpel.org/

[17] http://ode.apache.org

responsible for the plug-in and the removal of predefined Aspects into the BPEL process using the Aspect Activator Module, an extension to the BPEL engine similar to AO4BPEL; and (c) updating the original BPEL process by activating the appropriate aspect which is executed in the BPEL process to create a contextualized process.

Finally, Li, Liu and Bouguettaya (2009) introduce semantics for context-aware service composition. Specifically they introduce semantic weaving (in contrast to the syntactic weaving of other AOP-based approaches like AO4BPEL), they describe context as a concept of an ontology and model services by automata.

5.3.3 Service Discovery, Adapters and Rules for Adaptations

Another group of research efforts address the issue of service adaptation, by using service discovery, service composition, adapters and rule-based techniques. Such an effort is the work of Küster et al. (2005) that recognize the need for automatic service composition through different cases where the solutions proposed are: i) chaining of services through graph search, forward chaining, backward chaining or estimation regression planning, ii) behaviour-based composition and component-based composition and iii) knowledge gathering while searching and conditional plans.

Van der Aalst et al. (2009) aim to provide some foundational notions related to service interactions. More specifically they present an approach for addressing the service exposure, replacement, refinement and generation of service adapters' issues. They include Open Nets as a basic tool to explain and formalize services, which is a refinement of Petri Nets and can be also translated into BPEL. The actual service composition is based on a number of service interaction patterns that establish relationships between service instances and messages exchanged. Using such patterns van der Aalst et al. (2009) present a service refinement approach using transformation rules and service adapters.

Lanese et al. (2010) propose an approach for dynamic adaptation based on the combination of adaptation hooks and adaptation rules. Adaptation hooks are provided by the adaptable application and specify where adaptation can happen. Adaptation rules are external and specify when and how adaptation can be performed. The rules are provided by an adaptation manager and include a description of the activity to be adapted, a condition for adaptation and the new code of the activity as well as a set of required variables and some other non-functional information. At run-time, the rules are matched against the corresponding activities because they are either invoked by the running application or the adaptation manager. The adaptation manager will also make decisions on the order of application of the multiple rules according to their effect on the application.

Spanoudakis et al. (2005) present an approach for static and dynamic discovery and composition of services into service centric systems (SCS). They combine components for monitoring the compliance of service centric systems with requirements at run-time and components for discovering services. The specifications of the violated requirements detected by the monitoring component are used to generate queries for discovering services that could substitute the failed services. The framework assumes that a service centric system is built as a collection of Web services, which are coordinated by a composition process specified in BPEL4WS. The central component of the architecture is the SCS manager, which creates queries that specify both the interface of the required service, specified in WSDL and its behaviour constructed from the BPEL4WS compilation process by using transformation rules.

In a later work, Spanoudakis and Zisman (2010) present a service discovery framework that supports the design of service-based systems based on existing services and their adaptation during execution. The need for adaptation may be due to unavailability or malfunctioning of the services that systems deploy, changes in the services' context or the service-based system environment and/or emergence of new and better services. The main components of the framework are: the service requestor, the query processor and the service registry intermediary. The service requestor receives service requests from client applications and context information about the services participating in a service-based system. The query processor parses the different parts of a query and evaluates them against service specifications in the various service registries. The service registry intermediary supports the use of different service registries and the discovery of services stored in different types of registries. The design process supported by the framework is iterative and uses structural and behavioural design models of service-based systems to support discovery of services through queries that can fulfil the models. These services are used to reformulate the design models and trigger new service discovery iterations. The execution time adaptation allows services to be identified based on pull and push modes of query executions.

Hu et al. (2008) present a distributed approach to automatically discover a composition of services. The composition is based on the desired input and output of a target process. Candidate services are found through a search in a service repository by using user defined criteria and a publish/subscribe model was proposed for the process search.

Lemahieu et al. (2003) propose a way to improve Web service technology in terms of service discovery, service invocation and automated service composition by introducing the concept of business events. They propose that event notifications are broadcasted in parallel to all services that have an interest in an event of the corresponding type. To represent the information system in terms of business events, their effect on enterprise objects and the related business rules, the MERODE object-oriented analysis and design methodology was used. Although MERODE follows an object-oriented approach, business events are identified as independent concepts and their relation with the object types is defined in an object-event table (OET). The event concept is included in a richer Web service description compatible with the Web service standards such as SOAP, WSDL and UDDI.

Aydın et al. (2008) use a logical action-effect definition language, the Event Calculus in order to automate the preparation and execution of Web service compositions. Furthermore, these plans (if more than one) can be compiled into a graph so that the best one can be chosen by the execution engine. They also present an OWL-S to Event Calculus translation scheme so that planning with generic process definitions is possible.

Juric (2010) proposes a solution for support of business events and event driven architecture (EDA) concepts in service-oriented architecture (SOA). This is achieved by proposing specific extensions to WSDL and BPEL. WSDL extensions are XML constructs that enable a service to produce and receive events while assuring backward compatibility. BPEL extensions add support for event-based service orchestration with new activities for triggering and listening events and with extensions to the event and fault handlers, to variables and to property aliases.

Ferrari et al. (2006) argues on the need for adaptability in SOA and provide us with an attempt to explore the SOA features within computing environments, without a publicly addressing schema where visibility of service addresses is not always guaranteed. More specifically, they extend a middleware for service choreography called JSCL (Java Signal Core Layer) in order for it to cope with partial visibility of services through intermediate entities called gateways. To achieve this, an alternative SOAP binding for HTTP 1.1 is proposed based on the x-mixed-replace mimetype.

K. Lee et al. (2006) propose the development of an event-condition-action (ECA) XML-based rule description language named WS-ECA to support autonomous interactions among service-oriented devices in ubiquitous computing networks. The rules are embedded in distributed devices, which invoke appropriate services in the network if the rules are triggered by some internal or external events. Such rules that were also proposed by others (e.g. Jung et al., 2007; J. Park et al., 2008) can introduce the desired adaptation functionality in loosely coupled service orchestrations.

5.3.4 Recommendations in Services-based Systems

The consideration of user preferences and service properties may assist service selection. Recommendation techniques have been used in research projects to enhance Web service selection. Especially relevant for service recommendation are non-functional service properties such as response time, price, reputation, correctness, etc., often known as properties of Quality of Service. One of the first works in this domain is the work of Sreenath and Singh (2004) who first reformulated content-based filtering and Collaborative Filtering (CF), the two traditional recommender approaches, for service selection and proposed an agent-based approach in which agents cooperate to evaluate service providers. In this approach, agents rate each other and autonomously decide how much to weigh each other's recommendations. The underlying algorithm with which the agents reason is developed in the context of a concept lattice, which enables finding relevant agents and providing service recommendations. Mehta et al. (2004) find that semantics and syntax are insufficient to discover a service that meets users' needs. They add two more dimensions: quality and usage pattern. Blake and Nowlan (2007) propose to compute a Web service recommendation score by matching strings collected from a user's operational sessions and the description of Web services. Based on this score, their approach judges whether a user is interested in the service. Maamar et al. (2005) propose a model for the context of Web service interactions and highlight that the resources on which the Web services are performed have an impact on web services personalisation.

There exist several approaches applying CF to Web service recommendation. In these approaches, QoS values are predicted for an active user based on the QoS records provided by users who have similar historical QoS experiences on some Web services.

Shao et al. (2007) propose a user-based CF algorithm to predict QoS values. Zheng et al. (2009) propose a hybrid user-based and item-based CF algorithm to recommend Web services. Rong et al. (2009) propose a Web service ranking framework in which a set of users with similar interest are firstly identified. Afterwards, association rule mining is applied to all Web service composition transactions related to that set of users. By combining user group and association rule mined from that group, a personalized Web service ranking mechanism is achieved. Toma et al. (2009) explore the idea of using social annotations available in del.icio.us for ranking Web services. Chen et al. (2010) propose a hybrid CF approach based on the observation that QoS, regarded as a set of user-perceived properties, highly relates to users' physical locations. They first cluster users into several regions based on their physical locations and historical QoS similarities. Then region-sensitive services are identified. Next, a modified nearest neighbour approach is used to predict the QoS of the candidate Web services for an active user by leveraging historical QoS information gathered from users of highly correlated regions.

Beyond service-based systems, recommendations can be also useful for enabling flexibility in process management and workflow systems. Guidelines are explored as a means of providing recommendations. Adams et al. (2003) define each process step within such a guideline as a simple placeholder task, which is dynamically replaced by a context sensitive choice from an extensible catalogue of suitable workflow definitions during run-time. In addition, recommendations can be derived based on a rough task structure (Eichholz et al. 2004). In contrast to these guideline approaches that are mainly based on predefined process models and might not be instantiated at all, recommendations can be also based on best-practices shared by users within a company (Stoitsev et al. 2007). Pesic et al. (2006 and 2007) provide recommendations based on past experiences and additionally on a specific process goal. This is achieved by comparing the current process instance with past executions (logs), while preferring those executions that satisfy the specified goal. A similar approach is proposed by Schonenberg et al. (2008) where recommendations are generated based on similar past process executions by considering the specific optimization goals. Another approach is followed by Almeida et al. (2004) where recommendations are based on an ontology and semantic rules that generate possible process alternatives or suitable process steps if the execution of a workflow instance fails to proceed. Vanderfeesten et al. (2008) follow an approach in which, based on the information available for a case, the next step to be performed is determined using a strategy of e.g. lowest cost or shortest processing time. Dorn et al. (2010) present a self-adjusting approach for providing context-sensitive process recommendations based on the analysis of user behaviour, crowd processes, and continuous application of process detection. Specifically they provide recommendations learned from previous processes executed by 'that' user and couple them with process decisions taken from all users involved in that particular process type.

The importance of contextual information in service and process recommendations has been recognized by service engineering researchers. For example, in the approach of Chen et al. (2010), the location of users is taken into account in order to provide service recommendations based on QoS. Further, in some process recommendation approaches, the goal of the current process and other contextual information is taken into account for providing recommendations. Different approaches to using contextual information in the recommendation process can be broadly categorized in two groups: (1) recommendation via context-driven query and search, and (2) recommendation via contextual preference elicitation and estimation (Adomavicius and Tuzhilin, 2010). Systems using the first approach typically use contextual information (obtained either directly from the user, e.g., by specifying current expectations or interests, or from the environment, e.g., by obtaining current location or current process goal) to query or search a certain repository of resources (e.g., services) and present the best matching resources (e.g., services that best match goals) to the user. The second approach is using contextual information via the contextual preference elicitation and estimation. It attempts to model and learn user preferences, e.g., by observing the interaction of this and other users with the service-based application or by obtaining preference feedback from the user on various previous recommendations. To model users' context-sensitive preferences and generate recommendations, these techniques typically either adopt existing collaborative filtering, content-based, or hybrid recommendation methods to context-aware recommendation settings or apply various intelligent data analysis techniques from data mining or machine learning (such as Bayesian classifiers or support vector machines).

5.4 Role of Event-Based Service Adaptation in PLAY

In WP4 we will work to enhance the user's experience when interacting with a service-based application or system using contextual information and adapt the system behaviour to it. Within PLAY, events can be used as a source of context, since they are snippets of the past activities; therefore event processing results will be transferred to WP4 tools, injecting contextual information to the service-based application or system.

To achieve the aforementioned goal, we will first develop an event-based context model. Our model will include the general classes, properties, and relations of de Neto et al. (2005) - identity (who), location (where), time (when), activity (what) and (service) profiles (how) – and will be enriched with domain-specific concepts. More importantly, the model will link classes, properties and relations with related concepts and attributes of the (complex) event model.

Work in WP4 will also focus on context monitoring. Our intention within PLAY is to deal with context changes, which correspond to either atomic or complex events and use Complex Event Processing approaches (CEP) to model and identify them.

Context detected with the context monitors will be used towards two goals by developing the appropriate software components. The first goal is to recommend dynamic event subscriptions (see section3.2.1.3) that will enable how a service in the distributed EDA will "decide" which complex event to subscribe to. The second goal is to recommend service adaptations in order to define ways for successfully reacting to dynamic environmental changes. Such reactions are needed in the dynamically changing service setting of PLAY, in order to optimize service executions, correct system abnormal behaviour or even completely avoid problematic situations.

5.5 Discussion

Applying context detection and service recommendation and adaptation techniques in a cloud infrastructure such as the PLAY envisaged one raises several challenges. With respect to event-based systems, the short survey (see section 3.2.1.3) revealed novel ways for event subscriptions in which users do not explicitly define in advance their interests in specific events. Advances in publish/subscribe systems are considered valuable for dynamic and distributed systems that include many publishers creating new events that were not predefined at design time. The efforts that we reviewed reveal a tendency towards assessing user related context changes, combining them with the appropriate semantics and producing implicit subscriptions that should be interesting for the users. Differentiations appear in the way that each effort proposes to detect the user's context and its changes and reason about them. In tasks T4.1 and T4.2 of PLAY, we plan to continue this work towards the direction of capturing context changing situations and translating them into implicit event subscriptions, like in Berkovsky and Eytani(2005), by focusing on complex event subscriptions and extending semantically recommendation functionalities such in the work of Brenna et al. (2006).

With respect to service adaptation, we plan to combine and possibly extend prominent aspect-oriented approaches like the CHEVICHE framework (Hermosillo et al., 2010b) or BPEL'n'Aspects approach (Leymann, 2005) by integrating them to the distributed CEP of PLAY. We will also focus on extending aspects with context by using or extending the work of Morin at al. (2008) in order to enable context-sensitive recommendations for enabling adaptations of services to different circumstances. We should try to enable AOP-based adaptations triggered by the (D)CEP capabilities of PLAY. Moreover, we should enable context-sensitive recommendations for adaptations. Although there exist various approaches for applying recommendation techniques in service-based, systems, there is limited work in systematically incorporating context in the recommendation techniques. Further, the preferences of users interacting with the service-based systems are also not taken into account in most cases. To systematically take into account both context and user preferences we can build on the analysis provided by Adomavicius and Tuzhilin (2010). They note that, in its general form, context-aware recommender systems are built based on the knowledge of partial contextual user preferences and typically deal with data records of the form < user; item; context; rating >, where each specific record includes not only how much a given user liked a specific item, but also the contextual information in which the item was consumed by this user (e.g., Context = Saturday). Based on the presence of this additional contextual data, several important questions arise: How contextual information should be reflected when modelling service properties as well as user preferences? How contextual information can be used in the recommendation process? How can we enhance recommendations by considering the most recent user activities reflected in event streams (e.g., views, clicks of elements of service-based system)? How can we interpret distributed event streams to infer the current context? We will explore these indicative research questions in tasks T4.3 and T4.4 within PLAY.

6 Related Projects

6.1 Overview of the SOA4All Project

6.1.1 Project Description

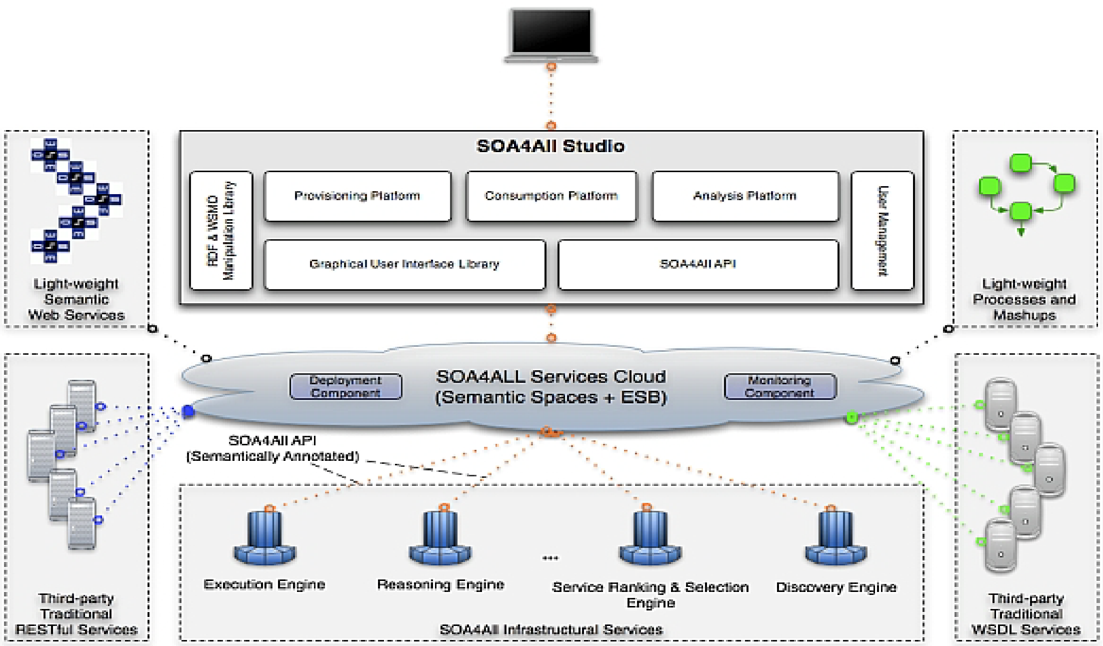
Service-Oriented Architectures for All (SOA4All[18]) is a Large-Scale Integrating Project funded by the European Seventh Framework Programme, under the Service and Software Architectures, Infrastructures and Engineering research area.

SOA4All aims at realizing a world where billions of parties are exposing and consuming services via advanced Web technology: the main objective of the project is to provide a comprehensive framework that integrates complementary and evolutionary technical advances (i.e., SOA, context management, Web principles, Web 2.0 and semantic technologies) into a coherent and domain-independent service delivery platform.

The SOA4All framework supports a world where a massive number of parties expose and consume services by realizing a coherent and domain independent platform. The overall architecture of SOA4All can be structured into four main parts:

1. SOA4All Studio: A rich Web platform that provides users with a unified view covering the whole lifecycle of services, including design-time, run-time and post-mortem analysis.
2. SOA4All Distributed Service Bus: The infrastructural backbone around which all the SOA4All components communicate and collaborate by combining Semantic Spaces and an extended Enterprise Service Bus.
3. SOA4All Platform Services: The group of services that provide the basic SOA4All functionality and activities, such as Service Ranking and Selection, Service Discovery, Service Adaptation, Service Composition, Service Execution, and the Reasoning Engine.
4. Business Services (3rd party Web services and light-weight processes): The actual services provided by final users. The SOA4All framework will be as technology agnostic and less intrusive as possible.

The following figure shows a high-level depiction of the overall SOA4All architecture, which is described in the next sections.



[18] http://www.soa4all.eu/

6.1.1.1 Service Bus

In the very centre, there is the SOA4All Distributed Service Bus. The Distributed Service Bus (DSB) serves as infrastructure service and core integration platform and is a direct evolution of the open-source PEtALS Enterprise Service Bus (Petals ESB Website, 2010) that is promoted by the OW2 Consortium (OW2 Website, 2010). The DSB delivers the necessary extensions and developments of the project to augment PEtALS towards large scale, open, distributed and hence Web-scale computing environments, as the one targeted by SOA4All. These extensions and consequently novelties of service bus infrastructures include ease of deployment, openness, and scalability in terms of internal repositories, amongst others. To realize these extensions, the DSB will benefit from the internal use of the ProActive distributed programming language also promoted by the OW2 Consortium.

Problems to be solved to augment existing ESB technology as required are of similar nature as those raised when programming, deploying, securing, monitoring, adapting a distributed application on a computing grid infrastructure.

Furthermore, the DSB is enhanced to incorporate a scalable Semantic Space infrastructure. The Semantic Space infrastructure is used as a shared memory to build repositories, as cooperative access to monitoring data, and as communication infrastructure to enhance the traditionally message-oriented bus towards a publication infrastructure for anonymous and asynchronous service communication with a notion of event-driven architecture; something currently neither available from openly-accessible ESB technology. In addition, the Deployment Facility provides a uniform, declarative, user-friendly, and automatic support for the management and distributed deployment of all software composing the whole SOA4All service computing environment (i.e., DSB, Studio, and platform services). Additionally, the Monitoring Platform of the SOA4All DSB collects monitoring data about the usage of SOA4All Platform Services and traditional 3rd party Web services. Finally, the DSB has been leveraged to an upper level by using federation mechanisms, which allow massive deployments and collaboration over the Internet connecting private Service Parks together, and providing the so-called Federated Distributed Service Bus (fDSB).

6.1.1.2 Studio and Platform Services

Around the integration platform, the SOA4All Distributed Service Bus, there are the SOA4All Studio and the SOA4All Platform Services. These are the components that are delivered by other research and development work packages in the project. The SOA4All Studio is defined in WP2 and delivers a fully Web-based user front-end that enables the creation, provisioning, consumption and analysis of the platform services and various 3rd party business services that are published to SOA4All. The studio supports different types of users at different times of interaction.

SOA4All knows two major groups of users: i) a large group of service consumers that use SOA4All for the consumption of functionality that is provided by either platform services or by business services, and ii) a significantly smaller group of (administrative) users that exploits the capabilities of the platform services (via the SOA4All Studio) to annotate, select and compose Web services. The platform services are products of the SOA4All project and deliver service discovery, ranking and selection, composition and invocation functionality, respectively. These components are exposed to the SOA4All Distributed Service Bus as Web services and hence consumable as any other published service. Their functionalities are used by the SOA4All Studio to offer clients the best possible functionality, while their combined activities (i.e., discovery, selection, composition and invocation) are coordinated via the DSB. The ensemble of DSB, SOA4All Studio and platform services delivers the innovative, fully Web-based and Web-enabled service experience of the SOA4All project: global service delivery at the level of the bus, Web-style service access via studio, and advanced state of the art service processing, management and maintenance via platform services.

6.1.1.3 Business Services (Web Services) and Processes

The figure also shows the semantic service descriptions and processes (composed services) that are created and processed by means of the SOA4All infrastructure. First, there are available Web services that are exposed either as traditional RESTful services (Representational state transfer), or as traditional WSDL-based services (lower parts of the figure). These are invokable third-party business services that SOA4All enhances in terms of automation, composition and invocation. Second, the top-left of the figure depicts the semantic annotations of the business services, so-called Semantic Web services. The semantic descriptions are published in the Service Registry, and used for reasoning with service capabilities (functionality), interfaces and non-functional properties, as well as context data. These semantic descriptions are the main enablers of the automation processes related to Semantic Web services. Third, in the top-right corner, light-weight processes and mash-ups are shown that are the basis for the definition and execution of composed services. Both, mash-ups and semantic descriptions of service compositions are published to the globally distributed and shared Semantic Spaces infrastructure, and become a public good for automated large-scale service computing.

6.1.2 Similarities / Dissimilarities to PLAY

As defined in its name, the SOA4All project is really a pure SOA-oriented approach while Play aims to be an EDA one. As defined several times, an EDA must be built on top of an effective SOA and results from the SOA4All infrastructure project will be used as the basis of the PLAY infrastructure:

- The SOA4All Semantic Space uses advanced techniques to provide a massively distributed semantic data store (Data is stored in RDF format).
- The SOA4All Federated Distributed Service Bus will be used and extended to move from SOA to EDA. This evolution will be possible with a better integration between the Semantic Space and the Distributed Service Bus.

6.1.3 Strengths / Weaknesses Compared to PLAY Approach

While the addressed architecture is not the same, the big strength of the SOA4All project is the work achieved around the so-called SOA4All Studio (the Web interface). This central tool provides all the features to manage and use the whole platform in an efficient way. Another important point is the potential to potentially address a large number of services among many organisations over the Internet and to move to the Cloud.

6.2 Overview of the STREAM Project

The STREAM[19] project aims at producing a highly scalable middleware platform able to process in real time massive data streams such as the IP traffic of an organization, the output of a large sensor network, the e-mail processed by an ISP, the market feeds from stock exchange and financial markets, the calls in a telco operator, credit card payments, etc.

6.2.1 Similarities / Dissimilarities to PLAY

The project has not yet listed any publications, but according to Christof Fetzer's keynote (Fetzer 2010) on the DEBS 2010 conference the project uses the StreamMine event processing system developed at the University of Dresden. As opposed to PLAY, StreamMine uses a variation of MapReduce, which is modified to deal with real-time results and their incremental computation. A similar adaptation of MapReduce was also published by (Condie et al. 2009).

To deal with incremental results, the MapReduce algorithm is adapted to avoid global synchronization or locking. Original implementations synchronised all operations before the Reduce phase. This was not suitable for event processing where events happen asynchronously and results e.g., complex events, must be obtained as fast as possible.

Nevertheless, there are other limitations of MapReduce, which might cause limitations in its applicability to generic CEP tasks. For example, in MapReduce there are a fixed number of mappers. Based on this, it is unclear how StreamMine will react to changing runtime dynamics to achieve elasticity in the cloud. Also it is unclear how StreamMine handles a large variety in event patterns e.g. in an extreme case where most patterns do not pertain to the same event processing network. Will MapReduce still provide a useful abstraction in this case? Lastly, it is unclear whether MapReduce is suitable for dynamic environments where event patterns change. Traditionally, MapReduce is a *single instruction, multiple data* (SIMD) operation. The operation must be stopped and restarted if any instruction changes.

6.3 Overview of the SRT-15 Project

The objective of the SRT-15[20] research project is to bridge the gap between cloud infrastructures and enterprise services by building a distributed service platform. For that purpose, SRT-15 relies on four key enabling technologies: content-based routing, complex event processing, dependability and data privacy.

6.3.1 Similarities / Dissimilarities to PLAY

Not many technical details are publicly available on project SRT-15 as of this writing. Like in project STREAM, Christof Fetzer of University of Dresden is a member of the consortium. We therefore assume that a similar technical foundation is used in both projects. Thus, our description of StreamMine and its MapReduce algorithm apply here, cf.6.3.

---

[19] http://www.streamproject.eu/

[20] http://www.srt-15.eu/

6.4 Overview of the SpoVNet Project

SpoVNet[21] (Spontaneous Virtual Networks) is a research project funded by the German "Baden-Württemberg Stiftung GmbH". It was launched in 2006 and is now in its second funding period.

In SpoVNet, Cordies (Koch, Koldehofe, & Rothermel 2010) is the CEP component of the project. Cordies claims to be designed for the operation in large-scale, heterogeneous networks and to adapt dynamically to changing network conditions. It is used as a vehicle to research various aspects of CEP, such as distribution, QoS or security.

6.4.1 Similarities / Dissimilarities to PLAY

Although SpoVNet has different use cases than PLAY, there are a number of technical similarities which make SpoVNet research relevant for PLAY.

According to (Koch, Koldehofe, & Rothermel 2010) SpoVNet performs distributed CEP on dedicated CEP nodes which form a special overlay network purely for the purpose of detecting complex events. Other types of projects reuse overlays dictated by broker networks which also serve other purposes than CEP. This might constrain elasticity and introduce constraints not relevant for an efficient, elastic CEP service as envisioned by PLAY. SpoVNet shares this idea.

Likewise, SpoVNet and PLAY share a separation of middleware and CEP systems. The middleware (event notification service) in SpoVNet is called EONSON, whereas the CEP component is the aforementioned Cordies.

Although architecturally and technically there are a number of similarities, SpoVNet and PLAY differ significantly in higher-level goals: SpoVNet targets a low-level CEP pattern language to target high expressivity at the cost of brevity. PLAY on the other hand wants to offer useful operators to allow quicker results for the use cases. Also PLAY offers RDF as an event format to integrate well with knowledge on the Web such as Linked Open Data. SpoVNet offers a simpler format of tuples with less built-in semantics.

6.5 Overview of the Pachube commercial Webplatform

Pachube[22], pronounced "PATCH-bay", is an online service allowing developers to connect sensor data to the Web. In the same way, event sources can be searched and used by others.

6.5.1 Similarities / Dissimilarities to PLAY

Pachube offers a platform to subscribe to and publish events, thus creating a marketplace for real-time data. For the PLAY platform this is a necessary requirement to perform CEP. Unlike PLAY, Pachube does not offer any computation on events. Pachube is simply a broker and a directory for events and event sources. PLAY, on the other hand, aims to process events and historical events together in one powerful platform using semantic technologies. Therefore, PLAY's planned features can be seen as a superset of the features currently offered by Pachube.

6.6 Overview of the Digital Graffiti project

Digital Graffiti[23] is a research project developed in cooperation between the University of Linz, Siemens Corporate Technology in Munich and the Ars Electronica Futurelab in Linz, which allows users to place smart messages in public and private places using mobile devices such as cell phones, PDAs, notebooks, etc. (Digital Graffiti Website 2010). A reference implementation of Digital Graffiti is a new location-based information- and communication system at the campus of the University of Linz. Students as well as lectors and administrative staff should be able to perceive location-based information on the university campus and to place information for other users.

6.6.1 Similarities / Dissimilarities to PLAY

Similarly to PLAY, Digital Graffiti is concerned with notions from the area of publish/subscribe. Digital Graffiti is viewed as *one-to-many communication* through a system (i.e. broker), which decouples senders and receivers e.g., in time. Digital Graffiti further defines methods of *addressing*, location-based, group-based, and considering indirection through the broker. Also, Digital Graffiti takes *context* into account when

[21] http://www.spovnet.de/

[22] http://www.pachube.com/

[23] Digital Graffiti Website: http://dg.jku.at/

delivering messages e.g. messages are only relevant on certain conditions or the content of messages is presented in a different manner to the user.

6.7 Overview of the ALIVE project

ALIVE [24] is an EU funded research project that finished in 2010 (31st of October). ALIVE aimed to combine Coordination and Organisation mechanisms (providing flexible, high-level means to model the structure of inter-actions between services in the environment) and Model Driven Design (providing for automated transformations from models into multiple target platforms) to create a framework for software and services engineering for distributed systems.

They presented a formal conceptual model and architecture for service-oriented systems which complements existing SOA approaches and adds new elements based on organisation, coordination and autonomy (such as goals, commitments, roles, rights, responsibilities, authority, power, norms, violations and sanctions). In addition, a set of metamodels, offline and on-line tools, which implement architectural elements necessary for the three levels of the ALIVE Theoretical Framework, were developed.

6.7.1 Similarities / Dissimilarities to PLAY

ALIVE project aimed to cover not only the deployment and execution of a given design system but also on-line monitoring mechanisms to detect failures or deviations from expected function and the mechanisms to correct, recover or adapt to them. This fact along with their crisis management scenario used to validate their approach (an emergency simulation tool for the Dutch emergency forces for coping with flooding disasters), constitute the specific project as highly relevant to PLAY.

We consider as one of their main strengths the use of Service Templates as intermediary descriptions linking (higher-level) goals or tasks with specific service interactions. These templates include parameterized process models that are dynamically bound at execution time into concrete ontological term instances. They use a workflow enactment module that receives an abstract workflow from the Coordination Module, with some (or all) tasks pointing to abstract services rather than to a concrete one (i.e. "mapping services" instead of a "Google maps service") and via the Matchmaker component look for services that can fulfil this abstract task, binding them to the task. With the integration of such components they succeed in defining flexible and dynamic process flows that start as abstract constructs at design at and become concrete at run-time.

The service adaptors, that they use, allow external services to be adapted without modification for given organisational tasks. This adaptation refers to type translation of services for changing the given service interface to the terms or data types used by the organisation. They distinguish three levels of adaptation:

- Service: changes in system functionalities. Automatic discovery of new services when old ones become unavailable
- Coordination: changes in environmental conditions. Generation o ne plans of actions for a specific objective when symptoms are detected that can lead to potential failures.
- Organisation: changes in stakeholders needs. Selection of new objectives when changes appear in laws and norms that regiment particular organizational protocols.

6.8 Overview of the WORKPAD project

The WORKPAD [25] project (finished in 2009) designed and developed an innovative software infrastructure for supporting collaborative work of human operators in emergency/disaster scenarios. This project tried to support different teams, belonging to different organizations that needed to collaborate with one other to reach a common goal; each team member was equipped with handheld devices (PDAs) and communication technologies, to carry on specific tasks. WORKPAD proposed a 2-level framework for such scenarios: a back-end peer-to-peer community, providing advanced services, data & knowledge & content integration, and a set of front-end peer-to-peer communities, that provide services to human workers, mainly by adaptively enacting processes on mobile ad-hoc networks.

6.8.1 Similarities / Dissimilarities to PLAY

The objective of the WORKPAD project was to investigate how to create communities of Public Safety Systems (PSSs), and how to enable mobile teams to exploit such back-end PSSs through the interplay of MANET (mobile ad hoc networking) technologies, process management and geo-collaboration. The 2-

[24] Project ALIVE Website: http://www.ist-alive.eu/

[25] http://www.workpad-project.eu/home.jsp

level peer-to-peer architecture, implemented in this project along with the crisis management scenario used to validate their findings, constitute WORKPAD as a related to PLAY project. They also present interesting work in adaptive process management, by trying to exploit context-awareness and process mining, in order to manage coordination of team members. They developed an adaptive process-management system (APMS) as the core element of the front-end middleware for controlling emergency management processes based on contextual information retrieved by a context monitor module. This contextual information is associated with devices, networks, team members and activities. Their process miner detects workflow patterns, individual member and team social behaviour along with possible correlations. Regarding the adaptive functionality they mainly focus on recovering disconnecting nodes (user's PDA) through specific tasks (e.g. assign the "follow disconnected node" task to another node to guarantee the connection).

6.9 Overview of the S-CUBE project

S-Cube[26] is a European Network of Excellence in Software Services and Systems (ends in 2012) and aims to establish an integrated and multidisciplinary research community, in order to help and shape the software-service-based Internet of the future. Among S-Cube's main objectives are the support of software services revolution and the advancement of agile, holistic service engineering and adaptation principles and techniques. Specifically, they try to provide methods and tools for building service-based systems in such a way that they can self-adapt while guaranteeing the expected level of service quality. Such an adaptation can be required due to changes in a system's environment or in response to predicted and unpredicted problems.

6.9.1 Similarities / Dissimilarities to PLAY

S-Cube as the most prominent Network of Excellence in service adaptation, points out the evolution and adaptation methods and tools as keys to enable service-based applications (SBAs). This fact constitutes S-Cube as highly relevant to PLAY project that intends to be aligned with the directions of S-Cube research work. Their research is organised around 6 interconnected fields in the areas of "Service Principles, Techniques and Methods" and "Service Technologies". The business process management mechanisms for expressing and managing enterprises organised in large, dispersed networks of value-added services along with the adaptation and monitoring techniques that aim to support the prediction and management of the activities of distributed systems that may lead to adaptations, will be the starting point of our PLAY research work in terms of service adaptation.

6.10 Overview of the TRIDEC project

TRIDEC[27] is an European project (start : 09/2010 ; end : 08/2013). It focuses on new technologies for real-time intelligent information management in collaborative, complex critical decision processes. Key challenge of the project is the construction of a communication infrastructure of interoperable services through which management of dynamically increasing volumes and dimensionality of information and data is efficiently supported; where groups of decision makers collaborate and respond quickly in a decision-support environment.

6.10.1 Similarities / Dissimilarities to PLAY

One of the two use cases of TRIDEC project relates to a crisis management scenario. This is about a large group of experts working collaboratively in crisis centres and government agencies using sensor networks. Their goal is to make critical decisions and save lives, infrastructural and industrial facilities in evolving tsunami crises.

Unfortunately, there are no technical details publicly available on TRIDEC at the moment.

6.11 Overview of the EMILI project

The EMILI[28] project (Emergency Management in Large Infrastructures) is a European funded project started in 2010.

It aims at providing a new generation of control systems for large Critical Infrastructures (CIs) like power grids and telecommunication systems, airports, railway / metro systems, oil and gas pipelines, in order to

[26] http://www.s-cube-network.eu/

[27] http://www.tridec-online.eu/

[28] http://www.emili-project.eu

improve safety and security of large Infrastructures, especially in the case of emergencies and crisis situations.

6.11.1 Similarities / Dissimilarities to PLAY

EMILI will use existing platform to implement their CEP, and according to their SOTA on the Analysis of Event Management Technologies, they choose to work with MonetDB/DataCell (open source column-oriented database system) and (according to their survey on complex event processing) to develop an Event Query Language (EQL) based on Xchange^{EQ}. EMILI will use ontologies and Web services technologies to manage the complex events, the risks and the impacts. EMILI platform will be able to collect, analyse and provide events to the involved stakeholders, and to help the collaboration actors to make a decision. This project will also take in account the QoS (detection of noise, missing events) : one of its goal is to implement an event stream processing system capable of handling discrete and continuous events in (near) real time. EMILI use cases are related to crisis and emergency situations in large infrastructures. We can see that a lot of characteristics of EMILI are close to PLAY ones, that makes EMILI highly relevant. But one topic is not clearly mentioned in EMILI description: what about the agility of the crisis process response?

7 Standardization

7.1 Standards relevant to PLAY

The current section goal is to list all the standards, which are relevant to the project. Main standards and most used ones will compose this non-exhaustive list.

7.1.1 SOA Standards

The following SOA standards relates to the ability to specify QoS contracts (WS-Agreement) and to manage QoS requirements (WSDM).

• OASIS – WSDM (WSDM specification, 2006): The Web Service Distributed Management specification defines the use of Web service architecture and technology to manage distributed resources. This specification is composed of two parts: MOWS stands for Management Of Web Services and MUWS stands for Management Using Web Services; defining message formats for management purposes.

• OGF – WS-Agreement (WS-Agreement specification, 2007) from OGF (Open Grid Forum): A Web Services protocol for establishing agreement between two parties, such as between a service provider and consumer, using an extensible XML language for specifying the nature of the agreement, and agreement templates to facilitate discovery of compatible agreement parties. The specification consists of three parts which may be used in a composable manner: a schema for specifying an agreement, a schema for specifying an agreement template, and a set of port types and operations for managing agreement life cycle, including creation, expiration, and monitoring of agreement states.

• OASIS – bSLA (committee specification – November 2010) is an on-going effort from the OASIS SOA-EERP (Service-Oriented Architecture End-to-End Resource Planning) TC aiming to define XML vocabulary for business service level agreement (bSLA). Business service level agreement describes the agreement between two parties, service requester and service provider, on business-related characteristics or attributes of a service. The SLA is network/system-oriented agreement that deals with network performance and system availability. The bSLA is a business-oriented agreement that deals with price, time to deliver, and the quality/rating of the service.

Petals ESB and Master provide already an implementation of both WSDM and WS-Agreement standards. The last one is integrated into the Petals Master open-source governance platform. The bSLA standard will be investigated in order to assess pros and cons versus WS-Agreement.

7.1.2 Event Standards

Following standards are relevant to Event Driven Architecture. Some of them deal with the standardization of Event's format, which is a value-added feature for interoperability; the others deal with the standardization of event-based middleware.

7.1.2.1 Event Format

- EPCGlobal (EPCGlobal Website, 2010): The EPCGlobal consortium provides the Electronic Product Code Information Services (EPCIS) specification that defines a standard way to share EPC observations within and across enterprises. An EPC observation is an event generated at a location, such as an assembly area, shipping and receiving dock door, or retail store, during the movement of an asset. EPCIS is a standard that defines the format for this event as well the interfaces to be implemented by software for capturing the event. Once the event is captured, EPCIS also defines an interface for saving the event into a repository and an interface for applications to query and obtain the event from the repository.

- OASIS – CAP (CAP specification v1.1, 2005): The CAP (Common Alerting Protocol) specification is being developed within the OASIS Emergency Management Technical Committee. It is a simple but general format for exchanging all-hazard emergency alerts and public warnings over all kinds of networks. CAP allows a consistent warning message to be disseminated simultaneously over many different warning systems, thus increasing warning effectiveness while simplifying the warning task. CAP also facilitates the detection of emerging patterns in local warnings of various kinds, such as might indicate an undetected hazard or hostile act. And CAP provides a template for effective warning messages based on best practices identified in academic research and real-world experience.

  This format seems very relevant to the Crisis management use case, consequently the project will study and implement this standard in the corresponding use case.

- OASIS – WEF (WSDM specification, 2006): The WSDM specification family provides an extensible XML event format called WEF that carries management event information. The format defines a set of basic, consistent data elements that allow different types of management event information to be carried in a consistent manner. The WSDM Event Format provides a basis for programmatic processing, correlation and interpretation of events from different products, platforms, and management technologies.

This last standard is relevant for managing QoS: main idea is to carry service's monitoring information thanks to events in the frame of the WSDM framework.

7.1.2.2 Event-Based Middleware

This part introduces event-based middleware standards. A lot of work has already been done in this area; main relevant specifications are listed below.

Many of those standards are relative to MOM (Message Oriented Middleware). Message-oriented middleware is software infrastructure focused on sending and receiving messages between distributed systems. MOM allows application modules to be distributed over heterogeneous platforms, and reduces the complexity of developing applications that span multiple operating systems and network protocols by insulating the application developer from the details of the various operating system and network interfaces. The lack of standards governing the use of message-oriented middleware has caused problems. All the major vendors have their own implementations, each with its own application programming interface (API) and management tools.

Consequently some standardization bodies have been started to work on this since some years and produced the following results.

- OMG – CORBA Event Service (CORBA Event Service 1.2 specification, 2004): The Event Service decouples the communication between objects. The Event Service defines two roles for objects: the supplier role and the consumer role. Suppliers produce event data and consumers process event data. Event data are communicated between suppliers and consumers by issuing standard CORBA requests.. The two approaches to initiating event communication are called the push model and the pull model. The push model allows a supplier of events to initiate the transfer of the event data to consumers. The pull model allows a consumer of events to request the event data from a supplier. In the push model, the supplier is taking the initiative; in the pull model, the consumer is taking the initiative. An event channel is an intervening object that allows multiple suppliers to communicate with multiple consumers asynchronously. An event channel is both a consumer and a supplier of events. Event channels are standard CORBA objects and communication with an event channel is accomplished using standard CORBA requests.
- OMG – DDS (DDS specification 1.2, 2007): The Object Management Group provides a collection of specification called Data Distribution Service covering real-time, CCM, and Interoperability. One event-related part of these specifications is the Data-Centric Publish-Subscribe layer. This layer provides the functionality required for an application to publish and/or subscribe to the values of data objects. DDS introduces a virtual Global Data Space where applications can share information by simply reading and writing data-objects addressed by means of an application-defined name (Topic) and a key. DDS features fine and extensive control of QoS parameters, including reliability, bandwidth, delivery deadlines, and resource limits.
- Java JMS (JEE/JMS specification 1.1, 2002): The Java Message Service (JMS) API is a Java Message Oriented Middleware (MOM) API for sending messages between two or more clients. JMS is a part of the Java Platform, Enterprise Edition, and is defined by a specification developed under the Java Community Process as JSR 914. It is a messaging standard that allows application components based on the Java 2 Platform, Enterprise Edition (J2EE) to create, send, receive, and read messages. It allows the communication between different components of a distributed application to be loosely coupled, reliable, and asynchronous. The JMS API supports two communication models: (i) point-to-point or queuing model; (ii) publish and subscribe model. JMS API is implemented by most MOM vendors and aims to hide the particular MOM API implementations; however, JMS does not define the format of the messages that are exchanged, so JMS systems are not interoperable.
- AMQP (2006): The Advanced Message Queuing Protocol (AMQP) is an industrial standard that defines the protocol and formats used in the messaging server and client, so implementations are interoperable. AMQP has been defined to provide flexible routing, including common messaging paradigms like point-to-point, fanout, publish/subscribe, and request-response.

The following specifications are related to SOA middleware.

- OASIS – WS-Notification (WS-Notification specification, 2006): This collection of specification provides a standardized way for a Web service, or other entity, to disseminate information to a set of other Web services, without having to have prior knowledge of these other Web Services. They can be thought of as defining Publish/Subscribe for Web services. This family is composed of three specifications: WS-BaseNotification dealing with standard mechanisms and description of publish/subscribe message payloads; WS-BrokeredNotification describing message exchanges involved in publish/subscribe of a message broker; and WS-Topics defining mechanisms to organize and categorize items of interest for subscriptions known as topics.

- W3C – WS-Eventing (WS-Eventing specification, 2006): This specification describes a protocol that allows Web services to subscribe or to accept subscriptions for notifications messages and relies on other Web service specifications to provide secure, reliable, and/or transacted message delivery and to express Web service and client policy.

Petals ESB provides an implementation of the WS-Notification standard. It is implemented as a JBI service engine. The main drawback of this implementation is its centralized architecture: the WS-Notification service engine is deploy as an end-point on one Petals ESB node and all subscribers or publisher have to use this endpoint. This creates a SPOF (Single Point Of Failure) and this is not a scalable architecture pattern. The project will investigate possibilities to setup a distributed architecture.

7.1.3 CEP Standards

- W3C – RDF (RDF specification, 2004): RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a "triple"). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications. This linking structure forms a directed, labelled graph, where the edges represent the named link between two resources, represented by the graph nodes.

8 Positioning PLAY in current State of the Art

Since the vision of the project is very challenging, especially from the technological point of view, there are many intersections between the current research and technological trends and our work.

In the previous sections we gave a comprehensive overview of the related research and shortly discussed our possible contribution to advancing state of the art.

In this section we present the impact that our results can have on the major research, technological and application trends. The goal is to create the awareness of the importance of our work in the broader context. This can help us in improving the visibility of our work and a better dissemination of results.

The following table illustrates the main relations between trends and our main R&D areas.

R&D areas / **Trends**	Federated middleware	Distributed CEP	Service adaptivity/	Contextualization	Platform as a whole
Future Internet	X	X			X
Dynamic Business		X	X		X
Context-aware systems	X			X	X
Event-driven systems		X		X	X
Real-time Business Intelligence/Monitoring		X		X	
Personalization		X	X	X	
Mobile computing		X	X	X	
Cloud Computing		X			
Sensor Networks	X	X			
Event Marketplace	X	X			X

Table 1: General trends and our main R&D areas

The plan is to update this table as well as the list of relevant projects on a regular basis. The goal is to enable so called continual monitoring of the state of the art that will ensure that the results will go beyond current state of the art.

9 References

Aberer, K., Cudré-Mauroux, P., Datta, A., Despotovic, Z., Hauswirth, M., Punceva, M. & Schmidt, R. (2003), 'P-Grid: a self-organizing structured P2P system', ACM SIGMOD Record 32(3), 33.

Aberer, K., Cudré-Mauroux, P., Hauswirth, M. & Pelt, T. V. 2004, 'GridVine: Building Internet-Scale Semantic Overlay Networks', in International Semantic Web Conference.

Abowd, GD & Mynatt E.D. 2000, 'Charting past, present, and future research in ubiquitous computing', ACM Transactions on Computer-Human Interaction, Vol. 7, Issue 1, pp. 29–58.

Abowd, GD, Ebling, M, Hunt, G, Lei, H & Gellersen, HW 2002, 'Context-Aware Computing', IEEE Pervasive Computing, Vol 1, Issue 3, pp. 22–23.

Adams, M., Edmond, D., ter Hofstede, A.H.M.2003, 'The application of activity theory to dynamic workflow adaptation issues', In Proceedings of the 7th Pacific Asia Conference on Information Systems, pp. 1836–1852.

Adomavicius, G., Tuzhilin, A. 2010, 'Context-Aware Recommender Systems', in P. Kantor, F. Ricci, L. Rokach, B. Shapira, Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners, Springer, 2010.

Afshar, M., Cincinatus, M., Hynes, D., Clugage, K., Patwardhan V., 2007, 'SOA Governance – Framework and Best Practices', White Paper, Oracle Corporation.

Agrawal, J.; Diao, Y.; Gyllstrom, D. & Immerman, N. 2008, 'Efficient pattern matching over event streams', *in* 'SIGMOD'.

Agrawal, R & Srikant, R 1994, 'Fast algorithms for mining association rules', *Proceedings of the 20th International Conference for Very Large Data Bases, VLDB,* pp. 487–499.

Aguilera, MK, Strom, RE, Sturman, DC, Astley, M & Chandra TD 1999, 'Matching events in a content-based subscription system', *Symposium on Principles of Distributed Computing,* Atlanta.

Alex, P., Chirita, R., Idreos, S., Koubarakis, M. & Nejdl, W. 2004, 'Designing Semantic Publish/subscribe Networks using Super-Peers', in Semantic Web and Peer-To-Peer.

Almeida, T., Vieira, S.C., Casanova, M.A. 2004, 'Flexible workflow execution through an ontology-based approach', In Proceedings of the Workshop on Ontologies as Software Engineering Artifacts, OOPSLA 2004.

Alvarez, G & Clark W, 2010, 'Context-Aware Computing and E-Commerce', Gartner Research, ID Number: G00200943.

Amazon SLA, http://aws.amazon.com/ec2-sla/, no date

Anicic, D.; Fodor, P.; Rudolph, S. & Stojanovic, N. (2011), EP-SPARQL: A Unified Language for Event Processing and Stream Reasoning, *in* 'WWW 2011', ACM, .

Anicic, D.; Fodor, P.; Stühmer, R. & Stojanovic, N. (2009), Event-driven Approach for Logic-based Complex Event Processing, *in* 'CSE '09: Proceedings of the 2009 12th IEEE International Conference on Computational Science and Engineering', IEEE Computer Society, Washington, DC, USA.

Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D. & Rabkin, A. 2010, 'A view of cloud computing', Communications of the ACM 53(4), 50.

Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A. D.; Katz, R. H.; Konwinski, A.; Lee, G.; Patterson, D. A.; Rabkin, A.; Stoica, I. & Zaharia, M. 2009, 'Above the Clouds: A Berkeley View of Cloud Computing'(UCB/EECS-2009-28), Technical report, EECS Department, University of California, Berkeley.

Atre, M., Srinivasan, J. & Hendler, J. 2008, BitMat, 'A Main-memory Bit Matrix of RDF Triples for Conjunctive Triple Pattern Queries', in 7th International Semantic Web Conference (ISWC).

Aydın, O, Kesim Cicekli, N & Cicekli, I 2008, 'Automated web services composition with the event calculus.' Engineering Societies in the Agents World VIII, pp. 142–157.

Babu, S. & Bizarro, P. 2005, 'Adaptive Query Processing in the Looking Glass', *in* CIDR, pp. 238-249.

Bai, X, Chen, Y & Shao, Z 2007,'Adaptive web services testing'*, In the Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC)*, pp. 233–236.

Baldoni, R. & Shvartsman, A. 2010, 'Theoretical aspects of dynamic distributed systems: report on the workshop, Elche, Spain, September 26, 2009', Special Interest Group on Algorithms and Computation Theory (SIGACT) News 40(4), 87–89.

Baldoni, R., Bertier, M., Raynal, M. & Tucci-Piergiovanni, S. 2007, 'Looking for a definition of dynamic distributed systems', in Parallel Computing Technologies, Springer, pp. 1–14.

Baran, D. (2008) Cloud computing basics. http://www.webguild.org/2008/07/cloud-computing-basics.php, July 2008.

Barbieri, D. F.; Braga, D.; Ceri, S.; Valle, E. D. & Grossniklaus, M. (2010), Incremental Reasoning on Streams and Rich Background Knowledge, *in* 'ESWC 2010b

Barbieri, D.F., Braga, D., Ceri, S., Grossniklaus, M.: An Execution Environment for C-SPARQL Queries. In: Proc. Intl. Conf. on Extending Database Technology (EDBT). 2010a

Baresi, L & Guinea, S 2005, 'Dynamo: Dynamic Monitoring of WS-BPEL Processes', In Proceedings of the Service-Oriented Computing - ICSOC 2005, Lecture Notes in Computer Science, Vol. 3826, pp. 478-483.

Baresi, L & Guinea, S 2010, 'Self-supervising BPEL Processes', IEEE Transactions on Software Engineering, (forthcoming)

Baresi, L, Ghezzi, C & Guinea, S 2004, 'Towards Self-healing Service Compositions', In the Proceedings of the First Conference on the PRInciples of Software Engineering (PRISE'04), pp. 11–20 (2004)

Baresi, L, Guinea S & Plebani, P 2007, 'Policies and Aspects for the Supervision of BPEL Processes', Advanced Information Systems Engineering, Lecture Notes in Computer Science, Vol. 4495, pp. 340-354.

Baresi, L, Guinea, S & Pasquale, L 2007, 'Self-healing BPEL processes with Dynamo and the JBoss rule engine', *In the Proceedings of the International workshop on Engineering of software services for pervasive environments*, pp. 11–20.

Battre, D. 2008, 'Caching of intermediate results in DHT based RDF stores', International Journal on Metadata Semantics and Ontologies 3(1), 84–93.

Battré, D., Heine, F., Höing, A. & Kao, O. 2006, 'On Triple Dissemination, Forward-Chaining, and Load Balancing in DHT Based RDF Stores', in DBISP2P, pp. 343–354.

Baude, F., Filali, I., Huet, F., Legrand, V., Mathias, E., Merle, P., Ruz, C., Krummenacher, R., Simperl, E., Hamerling, C. & Lorré, J.-P. 2010, 'ESB Federation for Large-Scale SOA', in Proceedings of the ACM Symposium on Applied Computing (SAC), pp. 2459–2466.

Belokosztolszki, A., Eyers, D., Pietzuch, P., Bacon, J. & Moody, K. (2003), Role-based access control for publish/subscribe middleware architectures, in `Proceedings of the 2nd international workshop on Distributed event-based systems', ACM, pp. 1-8.

Ben-Asher, Y & Berkovsky, S 2004, 'UNSO: Unspecified ontologies for peer-to-peer e-commerce applications', *Proceedings of the International Conference on Informatics,* Turkey.

Berkovsky, S & Eytani, Y 2005, 'Semantic Platform for Context-Aware Publish/Subscribe M-Commerce', Proceedings of the SAINT Workshops, Trento Italy, pp. 188-191.

Bernstein, A., Kiefer, C. & Stocker, M. 2007, 'OptARQ: A SPARQL Optimization Approach based on Triple Pattern Selectivity Estimation', Technical report, University of Zurich.

Bettini, C, Brdiczka, O, Henricksen, K, Indulska, J, Nicklas, D, Ranganathan, A & Riboni, D 2010, 'A survey of context modelling and reasoning techniques', Pervasive and Mobile Computing, Vol. 6, Issue 2, pp.161–180.

BizTalk Website, n.d. http://www.microsoft.com/biztalk/.

Blake, M. B. & Nowlan, M. F. 2007, 'A web service recommender system using enhanced syntactical matching', Proceedings of the 5th International Conference on Web Services (ICWS 2007).

Bolles, A., Grawunder, M., Jacobi, J.: Streaming SPARQL – Extending SPARQL to Process Data Streams. In: Proc. Europ. Semantic Web Conf. (ESWC). (2008) 448–462

Boniface, M.; Nasser, B.; Papay, J.; Phillips, S.; Servin, A.; Yang, X.; Zlatev, Z.; Gogouvitis, S.; Katsaros, G.; Konstanteli, K.; Kousiouris, G.; Menychtas, A. & Kyriazis, D. (2010), Platform-as-a-Service Architecture for Real-Time Quality of Service Management in Clouds, *in* 'Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference on', pp. 155 -160. (2010)

Boukadi, K, Ghedira, C & Vincent, L, 2008, 'An Aspect Oriented Approach for Context-Aware Service Domain Adapted to E-Business', In Z. Bellahsène and M. Léonard (Eds.): CAiSE 2008, LNCS, Vol. 5074, pp. 64–78.

Boukadi, K, Ghedira, C, Maamar, Z, Benslimane, D & Vincent L 2009, 'Context-Aware Data and IT Services Collaboration in E-Business', In Proceedings of the Transactions on Large-Scale Data-and Knowledge-Centered Systems I, pp.91–115.

Brenna, L, Gurrin, C, Johansen, D & Zagorodnov, D 2006, 'Automatic subscriptions in publish-subscribe systems', *Proceedings of the 26th IEEE Intl Conf. on Distributed Computing Systems (ICDCS) Workshops: Distributed Event-Based Systems (DEBS),* pp. 23-23.

Bresciani, P, Perini, A, Giorgini, P, Giunchiglia, F & Mylopoulos, J 2004 'Tropos: An agentoriented software development methodology', *In Autonomous Agents and Multi-Agent Systems*, Vol. 8, Issue 3, pp. 203–236.

Broekstra, J, Ehrig, M, Haase, P, van Harmelen, F, Menken, M, Mika, P, Schnizler, B & Siebes R 2004, 'Bibster - A Semantics-Based Bibliographic Peer-to-Peer System', *in The Second Workshop on Semantics in Peer-to-Peer and Grid Computing (SEMPGRID)*, New York.

Brown W., Moore G., Tegan W., 2006, 'SOA governance—IBM's approach., Application Innovation IBM Global Services Effective governance through the IBM SOA Governance Management Method approach' White paper.

Bruneton, E, Coupaye, T, Leclercq, M, Quema, V & Stefani, J 2006, 'The FRACTAL Component Model and its Support in Java', Software Practice and Experience, Special Issue on Experiences with Auto-adaptive and Reconfigurable Systems, Vol. 36, Issue 11-12, pp. 1257–1284.

Bry, F. & Eckert, M. 2007, 'Rule-Based Composite Event Queries: The Language XChangeEQ and Its Semantics', *in* RR, Springer, .

Bucchiarone, A, Cappiello, C, di Nitto, E, Kazhamiakin, R, Mazza, V and Pistore, M 2009, 'Design for Adaptation of Service-Based Applications: Main Issues and Requirements', In Proceeding of the ICSOC/ServiceWave Workshops 2009, pp. 467-476

Bucchiarone, A, Cappiello, C, di Nitto, E, Kazhamiakin, R, Mazza, V & Pistore, M 2010, 'A context-driven adaptation process for service-based applications', In Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems, pp. 50–56.

Buyya, R. 2009, 'Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility', in CCGRID '09: Proc. of the 2009 9th IEEE/ACM Int. Symposium on Cluster Computing and the Grid, IEEE Computer Society, Washington, DC, USA, p. 1.

Cai, M. & Frank, M. R. 2004, 'RDFPeers: a scalable distributed RDF Repository Based on a Structured Peer-to-Peer Network', in WWW, pp. 650–657.

Cai, M., Frank, M. R., Yan, B. & MacGregor, R. M. 2004, 'A subscribable Peer-to-Peer RDF Repository for Distributed Metadata Management', J. Web Sem. 2(2), 109–130.

Cai, M., Frank, M., Chen, J. & Szekely, P. 2003, 'MAAN: A multi-attribute Addressable Network for Grid Information Services', in Journal of Grid Computing, Vol. 2.

Callaway, RD, Viniotis, Y, Rodriguez, A, Brown, KG & Robinson, R 2008, 'Enabling federations of enterprise service buses using a distributed service registry'. Proceedings of the Second international workshop and Summer School on Service Science, Management and Engineering, SSME2008, Palermo Italy, 2-6 June 2008.

Calvanese D., De Giacomo G., Lembo D., Lenzerini M., and Rosati R.. DL-Lite: Tractable description logics for ontologies. In Manuela M. Veloso and Subbarao Kambhampati, editors, Proc. of the 20th Nat. Conf. on Arti?cial Intelligence (AAAI-05), pages 602–607. AAAI Press/The MIT Press, 2005.

Canfora, G & di Penta, M2006 'SOA: Testing and Self-checking', *In the Proceedings of the International Workshop on Web Services - Modeling and Testing - WS-MaTE*, pp. 3 – 12.

Castro, J, Kolp, M & Mylopoulos, J 2002, 'Towards requirements-driven information systems engineering: The tropos project', *In Information Systems*, Vol. 27, Issue 6, pp. 365–389.

Castro, M., Costa, M. & Rowstron, A. 2005, 'Debunking Some Myths about Structured and Unstructured Overlays', in Proceedings of the 2nd conference on Symposium on Networked Systems Design and Implementation (NSDI), USENIX Association, pp. 85–98.

Castro, M., Druschel, P., Kermarrec, A. & Rowstron, A. 2002, 'SCRIBE: A large-scale and decentralized application-level multicast infrastructure', IEEE Journal on Selected Areas in Communications 20(8), 1489–1499.

Catarci, T, de Leoni, M, Marrella, A, Mecella, M, Salvatore, B, Vetere, G, Dustar, S, Juszczyk, L, Manzoor, A, 2008, 'Pervasive software environments for supporting disaster responses', IEEE Internet Computing, pp.26–37.

Centrasite Website, 2010. http://www.centrasite.com/.

Chan, W, Cheung, S & Leung, K 2007, 'A metamorphic testing approach for online testing of service-oriented software applications', In International Journal of Web Services Research, Vol. 4, issue 2, pp. 61–81.

Chandrasekaran, S.; Cooper, O. & et al. (2003), TelegraphCQ: Continuous Dataflow Processing for an Uncertan World, *in* 'CIDR'.

Chapell, 2004, 'Entreprise Service Bus', O'Reilly Media.

Charfi, A & Mezini, M 2007, 'Ao4bpel: An aspect-oriented extension to bpel', *World Wide Web*, vol. 10, no. 3, pp. 309–344.

Charfi, A, Dinkelaker, T. & Mezini, M 2009, 'A plug-in architecture for self-adaptive web service compositions', *Proceedings of the 2009 IEEE International Conference on Web Services (ICWS '09)*, IEEE Computer Society, pp. 35–42.

Chen, X. Liu, X., Huang, Z. & Sun, H. 2010, 'RegionKNN: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation', In Proceedings of the 2010 IEEE International Conference on Web Services, pp.9-16.

Clark, W & Lapkin, A 2008, 'Fundamentals of Context Delivery Architecture: Introduction and Definitions', Gartner Research ID Number: G00161876

Clark, W 2010, 'Fundamentals of Context Delivery Architecture: Provisioning Context-Enriched Services', Gartner Research, ID Number: G00200649

Condie, T.; Conway, N.; Alvaro, P.; Hellerstein, J. M.; Elmeleegy, K. & Sears, R. (2009), 'MapReduce Online'(UCB/EECS-2009-136), Technical report, EECS Department, University of California, Berkeley.

CORBA Event Service specification (n.d.), http://www.omg.org/spec/EVNT/

Corsaro, A, Querzoni, L, Scipioni, S, Tucci, S & Virgillito, A 2006, 'Global Data Management', *Chapter Quality of Service in Publish/Subscribe Middleware*, IOS Press, pp. 1-19.

Cosminexus Website, 2010. http://www.hitachi.co.jp/Prod/comp/soft1/global/prod/cosminexus/.

Cudré-Mauroux, P., Agarwal, S. & Aberer, K. 2007, 'Gridvine: An infrastructure for peer information management', IEEE Internet Computing 11, 36–44.

Cuenca-Grau, B., Halaschek-Wiener, C., Kazakov, Y.: History matters: Incremental ontology reasoning using modules. In: Proceedings of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007. Volume 4825 of Lecture Notes in Computer Science. (2007) 183–196

Cugola, G & Jacobsen, HA 2002, 'Using publish/subscribe middleware for mobile systems', *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 4, pp. 25–33.

Dabek, F., Li, J., Sit, E., Robertson, J., Kaashoek, M. F. & Morris, R. 2004, 'Designing a DHT for low latency and high throughput', in Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation-Volume 1, p. 7.

Dai, W & Liu, JJ 2010, 'Context aware adaptive services framework', In Proceedings of the 7th International Conference on Service Systems and Service Management (ICSSSM), pp. 1–6.

Dantas, F, Batista, T & Cacho, N 2007, 'Towards aspect-oriented programming for context-aware systems: A comparative study' In Proceedings of the 1st International Workshop on Software Engineering for Pervasive Computing Applications, Systems, and Environments. pp. 4.

David P & Ledoux, T 2006, 'An Aspect-Oriented Approach for Developing Self-Adaptive Fractal Components', In Proceedings of the 5th Int. Symposium on Software Composition, LNCS, Vol. 4089, Vienna, Austria, pp. 82-99.

David, PC & Ledoux, T 2005, 'Wildcat: a generic framework for context-aware applications' In Proceedings of the MPAC, pp. 1–7.

DDS specifcation (n.d.), http://www.omg.org/technology/documents/dds_spec_catalog.htm

Della Valle, E., Turati, A. & Ghioni, A. 2006, 'PAGE: A distributed infrastructure for fostering RDF-based interoperability', in Distributed Applications and Interoperable Systems (DAIS), Springer, pp. 347–353.

Derler P., Rainer W., 2007, 'Models and Tools for SOA Governance', in Proceedings of Trends in Enterprise Application Architecture Lecture Notes in Computer Science 4473, 2007, Springer Berlin / Heidelberg, pp: 112-126.

Deshpande, A.; Ives, Z. & Raman, V. (2007), 'Adaptive query processing', *Found. Trends databases* **1**(1), 1--140.

Deussen, P, Din, G & Schieferdecker, I 2003, 'A TTCN-3 based online test and validation platform for Internet services', In the Proceedings of the 6th International Symposium on Autonomous Decentralized Systems (ISADS), pp. 177–184.

Dey, AK & Abowd, GD 2000, 'Towards a Better Understanding of Context and Context-Awareness', In Proceedings of the PrCHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness, pp. 304-307

Dey, AK, 2001, 'Understanding and using context', Personal and ubiquitous computing, Vol. 5, Issue 1, pp.4–7.

di Penta, M, Bruno, M, Esposito, G, et al. 2007, 'Web Services Regression Testing', In L. Baresi, E. Di Nitto (eds.) Test and Analysis of Web Services, pp. 205 – 234. Springer.

Digital Graffiti Website 2010, http://dg.jku.at/

Dorn, C. Burkhart, T., Werth, D. & Dustdar, S. 2010, 'Self-adjusting Recommendations for People-Driven Ad-Hoc Processes', in BPM 2010, pp. 327–342.

Eichholz, C., Dittmar, A., Forbrig, P. 2004, 'Using task modelling concepts for achieving adaptive workflows', In Proceedings of the EHCI DS-VIS 2004, pp. 96–111.

El-Ansary, S., Alima, L., Brand, P. & Haridi, S. 2003, 'Efficient Broadcast in Structured P2P Networks', in Peer-to-Peer Systems II, Springer, pp. 304–314.

EPCGlobal Website, 2010. http://www.epcglobalinc.org/.

Erradi, A, Maheshwari, P & Tosic, V 2006, 'Policy-Driven Middleware for Self-adaptation of Web Services Compositions', In the Proceedings of the 7th International ACM Middleware Conference, pp. 62–80.

Etzion, O, Skarbovsky, I, Magid, Y, Zolotorevsky, N, & Rabinovich, E 2010, 'Context Aware Computing and its utilization in event-based systems', Tutorial presented in DEBS'10, Cambridge, UK

Eucalyptus Systems Website 2010, Website. http://open.eucalyptus.com/.

Eugster, PT, Felber, PA & Kermarrec, AM 2003, 'The many faces of publish/subscribe', *ACM Computing Surveys*, vol. 35, pp. 114-131, 2003.

Ferrari, G, Guanciale, R & Strollo, D 2006, 'Event based service coordination over dynamic and heterogeneous networks.' Proceedings in the Service-Oriented Computing–ICSOC 2006, Chicago, USA, pp. 453–458.

Fetzer, C. (2010), StreamMine: a scalable and dependable event processing platform [Keynote], *in* 'Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems', ACM, New York, NY, USA, pp. 222--222.

Fidler, E.; Jacobsen, H.-A.; Li, G. & Mankovski, S. 2005, 'The padres distributed publish/subscribe system', *in* 8th International Conference on Feature Interactions in Telecommunications and Software Systems, pp. 12--30.

Filali, I., Bongiovanni, F., Huet, F. & Baude, F. (n.d.), RDF Data Indexing and Retrieval: A survey of Peer-to-Peer based solutions, Technical report. URL: http://hal.archives-ouvertes.fr/inria-00540314/en/

Fiorano Website 2010. http://www.fiorano.com/products/ESB-enterprise-service-bus/.

FlexiScale Website (n.d.). http://www.flexiant.com/products/flexiscale/.

Fuse ESB Website, 2010. http://fusesource.com/products/enterprise-servicemix/.

Garces-Erice, L, Biersack, EW, & Urvoy-Keller, G. 2003, 'Hierarchical Peer-to-peer Systems', in Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par), pp. 643–657.

Gehlert, A, Metzger, A, Karastoyanova, D, Kazhamiakin, R, Pohl, K, Leymann, F & Pistore, M 2011, 'Chapter 6: Integrating Perfective and Corrective Adaptation of Service-based Applications', *In Service Engineering: European Research Results Book by Dustdar S & Li F*, Springer ISBN: 978-3-7091-0414-9, pp. 137-169.

Georgakopoulos, K. & Margaritis, K. 2008, 'Distributed and Parallel Systems',Springer, chapter Integrating Condor Desktop Clusters with Grid, pp. 37–42.

Ghodsi, A. 2006, 'Distributed k-ary System: Algorithms for Distributed Hash Tables', Thesis, KTH - Royal Institute of Technology.

GIF Website, 2010. https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-130-27^2804_4000_100__/.

Gilbert, S. & Lynch, N. 2002, 'Brewer's Conjecture and the Feasibility of Consistent Available Partition-Tolerant Web Services', in ACM SIGACT News, p. 2002.

Gnutella RFC 2003, http://rfc-gnutella.sourceforge.net/.

GoGrid Website (n.d.). http://www.gogrid.com/.

Goldchleger, A., Kon, F., Goldman, A. & Finger, M. 2004, 'Integrade: Ob ject-oriented grid middleware leveraging idle computing power of desktop machines', 16, 16–449.

Graham,S et al., 2004, 'Publish-subscribe notification for web services v. 1.0', *Joint specification by BEA Systems, IBM, and Microsoft*, available online at: http://www-128.ibm.com/developerworks/library/ws-pubsub/WSPubSub.pdf.

Grassi, V. & Sindico, A., 2007. Towards model driven design of service-based context-aware applications. In International workshop on Engineering of software services for pervasive environments: in conjunction with the 6th ESEC/FSE joint meeting. pp. 69–74.

Gregory Todd Williams, Jesse Weaver, M. A. & Hendler, J. A. 2009, 'Scalable Reduction of Large Datasets to Interesting Subsets', in 8th International Semantic Web Conference.

Gu, T., Pung, H. K. & Zhang, D. 2007, 'Information Retrieval in Schema-based P2P Systems Using One-dimensional Semantic Space', Computer Networks 51(16), 4543–4560.

Guerraoui, R, Levy, RR, Pochon, B & QuéÃľma, V 2010, 'Throughput optimal total order broadcast for cluster environments', ACM Trans. Comput. Syst. 28(2), 1–32.

Harth, A. & Decker, S. 2005, 'Optimized Index Structures for Querying RDF from the Web', in Proceedings of the Third Latin American Web Congress (LA-WEB), IEEE Computer Society, Washington, DC, USA, p. 71.

Henning M & Spruiell M 2009, 'Distributed programming with ICE, revision 3.3.1' available online at: http://www.zeroc.com/doc/Ice-3.3.1/manual/toc.html

Hermosillo, G, Seinturier L & Duchien L 2010a, 'Creating Context-Adaptive Business Processes, Service-Oriented Computing', Lecture Notes in Computer Science, Vol. 6470, pp. 228-242

Hermosillo, G, Seinturier L & Duchien L 2010b, 'Using Complex Event Processing for Dynamic Business Process Adaptation', In Proceedings of the IEEE SCC 2010, pp 466-473

Hielscher, J, Kazhamiakin, R, Metzger, A & Pistore, M 2008, 'A framework for proactive selfadaptation of service-based applications based on online testing', *In the Proceedings of the ServiceWave 2008*, pp. 122-133

Hinze, A, Michel, Y & Eschner, L 2009, 'Event-based communication for location-based service collaboration', *Proceedings of the Australasian Database Conference (ADC 2009)*, Wellington, New Zealand, pp. 127-136.

Hogenboom, A., Milea, V., Frasincar, F. & Kaymak, U. 2009, 'RCQ-GA: RDF Chain Query Optimization Using Genetic Algorithms', in Proceedings of the 10th International Conference on E-Commerce and Web Technologies, Springer-Verlag, pp. 181–192.

Hong, J, Suh, E & Kim, SJ, 2009, 'Context-aware systems: A literature review and classification', Expert Systems with Applications, Vol.36, Issue 4, pp.8509–8522.

Hu, S, Muthusamy, V, Li, G & Jacobsen, HA 2008, 'Distributed automatic service composition in large-scale systems,' in Proceedings of the 2nd International Conference on Distributed Event-based Systems, Rome, Italy, pp. 233–244.

Jain, N, Kit, D, Mahajan, P, Yalagandula, P, Dahlin, M & Zhang, Y 2007, 'STAR: self-tuning aggregation for scalable monitoring', *in* VLDB '07: Proceedings of the 33rd international conference on Very large data bases, VLDB Endowment, pp. 962--973.

Jayaraman, P, Whittle, J, Elkhodary, A & Gomaa, H 2007, 'Model Composition in Product Lines and Feature Interaction Detection Using Critical Pair Analysis', Proceedings of the 10th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, Nashville USA, pp. 151-165.

JBI Specification, 2005. http://jcp.org/en/jsr/detail?id=208

Jha, S., Katz, D., Parashar, M., Rana, O. & Weissman, J. 2009, 'Critical perspectives on large-scale distributed applications and production grids', in Grid Computing, 2009 10th IEEE/ACM International Conference on, pp. 1 –8.

JMS specification, 2003, http://jcp.org/en/jsr/detail?id=914

Jones, N 2010, 'Context-Enriched Services: From Reactive Location to Rich Anticipation', Gartner Research, ID Number: G00175559

Jung, JY, Park, J, Han, SK & Lee, K 2007, 'An ECA-based framework for decentralized coordination of ubiquitous web services.' Information and Software Technology, vol. 49, no. 11-12, pp. 1141–1161.

Juric, MB 2006, 'Business Process Execution Language for Web Services BPEL and BPEL4WS 2nd Edition', Packt Publishing.

Juric, MB 2010, 'WSDL and BPEL extensions for Event Driven Architecture.' Information and Software Technology, vol. 52, no. 10, pp. 1023-1043.

Kapitsaki, GM, Prezerakos, GN, Tselikas, ND & Venieris, IS 2009. 'Context-aware service engineering: A survey', Journal of Systems and Software, Vol. 82, Issue 8, pp.1285–1297.

Karastoyanova, D & Leymann, F 2009, 'BPEL'n'Aspects: Adapting Service Orchestration Logic', *Proceedings of the IEEE International Conference on Web Services, ICWS 2009*, pp. 222 – 229.

Karnstedt, M., Sattler, K.-U., Hauswirth, M. & Schmidt, R. 2008, 'A DHT-based Infrastructure for Ad-hoc Integration and Querying of Semantic Data', in Proceedings of the 2008 international symposium on Database engineering and applications (IDEAS), ACM, New York, NY, USA, pp. 19–28.

Keith Jeffery, B. N.-L. (n.d.), 'The future of cloud computing: Opportunities for european cloud computing beyond 2010', http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf.

Khalaf, R, Karastoyanova, D & Leymann, F 2007, 'Pluggable Framework for Enabling the Execution of Extended BPEL Behavior', *Proceedings of WESOA workshop at ICSOC'07*, Springer, pp. 376-387.

Kiczales, G, Hilsdale, E, Hugunin, J, Kersten, M, Palm, J, & Griswold, WG 2001, 'An overview of AspectJ', *Proceedings of the 15th European Conference on Object-Oriented Programming (ECOOP '01),* Springer-Verlag, pp. 327–353.

Kiczales, G, Lamping, J, Mendheka, A, Maeda, C, Lopes, CV, Loingtier, JM & Irwin, J 1997, 'Aspect-Oriented Programming'*, Proceedings of the European Conference on Object-Oriented Programming (ECOOP)*, Lecture Notes in Computer Science, S. Gjessing and K. Nygaard, Eds., no. 1241. Springer, pp. 220-242.

Knežević, P., Wombacher, A. & Risse, T. 2009, 'DHT-Based Self-adapting Replication Protocol for Achieving High Data Availability', Advanced Internet Based Systems and Applications pp. 201–210.

Koch, G., Koldehofe, B. & Rothermel, K. (2010), Cordies: expressive event correlation in distributed systems, *in* 'Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems', ACM, New York, NY, USA, pp. 26--37.

Kongdenfha, W, Saint-Paul, R, Benatallah, B & Casati, F 2006, 'An Aspect-Oriented Framework for Service Adaptation', In Proceedings of the 4th International Conference in Service-Oriented Computing - ICSOC 2006, Chicago, IL, USA, Springer, pp. 15–26.

Kossmann, D. 2000, 'The State of The Art in Distributed Query Processing', ACM Computing Surveys 32(4), 422–469.

Koubarakis, M., Miliaraki, I., Kaoudi, Z., Magiridou, M. & Papadakis-Pesaresi, A. 2006, 'Semantic Grid Resource Discovery using DHTs in Atlas', in Proceedings of 3rd GGF Semantic Grid Workshop, Athens, Greece.

Ktari, S, Zoubert, M, Hecker, A & Labiod, H 2007, 'Performance Evaluation of Replication Strategies in DHTs Under Churn', in Proceedings of the 6th International Conference on Mobile and Ubiquitous Multimedia (MUM), ACM, New York, NY, USA, pp. 90–97.

Kuno, H., Rundensteiner, E.: Incremental maintenance of materialized object-oriented views in multiview: Strategies and performance evaluation. IEEE Transactions on Data and Knowledge Engineering 10(5) (september 1998) 768–792

Küster, U, Stern, M & König-Ries, B 2005, 'A classification of issues and approaches in automatic service composition,' In Proceedings of the 1st International Workshop on Engineering Service Compositions (WESC05), Amsterdam, Netherlands.

Lahire, P, Morin, B, Vanwormhoudt, G, Gaignard, A, Barais, O & J´ez´equel, JM 2007, 'Introducing Variability into Aspect-Oriented Modeling Approaches', Proceedings of the 10th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, Nashville USA, pp. 498-513.

Laliwala, Z, Sorathia, V & Chaudhary, S 2006, 'Semantics based event-driven publish/subscribe service-oriented architecture', *Proceedings of the workshop on Emerging Services delivery platforms and software models for Next Generation Network Services (SOFTPLATFORM2006),* New Delhi, India.

Lanese, I, Bucchiarone, A & Montesi, F 2010, 'A framework for rule-based dynamic adaptation.' Proccedings of the 5th International Symposium on Trustworthy Global Computing (TGC 2010), Munchen, Germany, pp. 284–300.

Lausen, G.; Ludäscher, B. & May, W. 1998, 'On Active Deductive Databases: The Statelog Approach', *in* ILPS'97.

Lee, K, Lee, W, Jeon, J, Lee, S & Park, J 2006, 'Event-driven Coordination Rule of Web Services enabled Devices in Uniquitous environments', Proceedings of the W3C Ubiquitous Web Workshop, Tokyo, Japan.

Lemahieu, W, Snoeck, M, Michiels, C, Goethals, F, Dedene G, Vandenbulcke, J 2003, 'Event based web service description and coordination.' Proceedings of the Web Services, e-Business, and the Semantic Web Workshop of the CaiSE'03 Conference, Klagenfurt, Austria, Lecture Notes in Computer Science Vol. 3095, pp. 120–133.

Lenk, A, Klems, M, Nimis, J, Tai, S & Sandholm, T 2009, 'What's inside the Cloud? An architectural map of the Cloud landscape', *in* CLOUD '09: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, IEEE Computer Society, Washington, DC, USA, pp. 23--31.

Lenk, A.; Klems, M.; Nimis, J.; Tai, S. & Sandholm, T. (2009), What's inside the Cloud? An architectural map of the Cloud landscape, *in* 'CLOUD '09: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing', IEEE Computer Society, Washington, DC, USA, pp. 23--31.

Leymann, F 2005, 'The (Service) Bus: Services Penetrate Everyday Life', Proceedings of the 3rd International Conference in Service-Oriented Computing, (ICSOC'2005), Amsterdam, the Netherlands, pp. 12-20.

Li, G. & Jacobsen, H. 2005, 'Composite subscriptions in content-based publish/subscribe systems', *Lecture notes in computer science* **3790/2005**, 249--269.

Li, L, Liu, D & Bouguettaya, A 2009, 'Semantic Weaving for Context-Aware Web Service Composition', In Proceedings of the 10th International Conference on Web Information Systems Engineering, pp. 101-114.

Liarou, E., Idreos, S. & Koubarakis, M. (2005), Publish/Subscribe with RDF Data over Large Structured Overlay Networks, in 'DBISP2P', pp. 135–146.

Liarou, E., Idreos, S. & Koubarakis, M. 2006, 'Evaluating Conjunctive Triple Pattern Queries over Large Structured Overlay Networks', in Proceedings of 5th International Semantic Web Conference (ISWC).

Liarou, E., Idreos, S. & Koubarakis, M. 2008, 'Continuous RDF Query Processing over DHTs', Lecture Notes in Computer Science 4825, 324–339.

Liu, Y & Plale, B 2003, 'Survey of publish subscribe event systems', *Technical Report TR574,* Indiana University, Bloomington.

Luckham, D. C. 2001, '*The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems'*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Maamar, Z., Mostefaoui, S.K. & Mahmoud, Q.H. 2005, 'Context for personalized web services', in Proc. The 38th Annual Hawaii International Conference.

Mahambre, S. P., S.D, M. K. & Bellur, U. 2007, 'A Taxonomy of QoS- Aware, Adaptive Event-Dissemination Middleware', IEEE Internet Computing 11(4), 35–44.

Marks, E., 2008, 'Service-Oriented Architecture Governance for the Services Driven Enterprise', John Wiley & Sons, Inc. Editions.

Martin, EA 2010, 'The Oxford Dictionary of Science, 6th edition', Oxford University Press, ISBN-13: 9780199561469

Mathias E., 'Hierarchical Multi-domain computing based upon a component-oriented approach', *PhD Thesis, Univ. of Nice Sophia-Antipolis*, December 2010.

Matono, A., Pahlevi, S. & Ko jima, I. 2007, 'RDFCube: A P2P-Based Three-Dimensional Index for Structural Joins on Distributed Triple Stores', Databases, Information Systems, and Peer-to-Peer Computing pp. 323–330.

Mehta, B., Niederée, C., Stewart, A., Muscogiuri, C. & Neuhold, E. J. 2004, 'An Architecture for Recommendation Based Service Mediation', in M. Bouzeghoub et al. (Eds.): Semantics of a Networked World (ICSNW 2004), LNCS 3226, pp. 250–262.

Miklós, Z. (2002), Towards an access control mechanism for wide-area publish/subscribe systems, in `Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on', IEEE, pp. 516-521.

Moltchanov, B, Knappmeyer, M, Fuchs, O & Paschetta, E, 2009, 'Context management and reasoning for adaptive service provisioning', In Proceedings of the International Conference on Ultra Modern Telecommunications & Workshops,ICUMT'09, pp. 1–6.

Morin, B, Barais, O, Nain, G & Jézequel, JM 2009 'Taming dynamically adaptive systems using models and aspects', *In Proceedings of the 31st International Conference on Software Engineering*, Vancouver, Canada, pp. 122–132.

Morin, B, Fleurey, F, Bencomo, N, Jé´ezé´equel, JM, Solberg, A, Dehlen, V & Blair, G 2008, 'An Aspect-Oriented and Model-Driven Approach for Managing Dynamic Variability', *Proceedings of the 11th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MoDELS'08),* Toulouse, France, pp. 782-796.

Moser, O, Rosenberg, F & Dustdar, S 2008, 'Non-Intrusive Monitoring and Service Adaptation for WS-BPEL', *In Proceedings of the International Conference on World Wide Web*, pp 815-824.

Moser, O, Rosenberg, F & Dustdar, S 2010, 'Event Driven Monitoring for Service Composition Infrastructures', *In Proceedings of the 11th International Conference on Web Information System Engineering (WISE'10*), Hong Kong, China

Mu, Y., Yu, C., Ma, T., Zhang, C., Zheng, W. & Zhang, X. 2009, 'Dynamic Load Balancing With Multiple Hash Functions in Structured P2P Systems', *In Proceedings of the 5th International Conference on Wireless communications, networking and mobile computing (WiCOM),* IEEE Press, Piscataway, NJ, USA, pp. 5364–5367.

Mule Galaxy Website, 2009. http://www.mulesoft.org/documentation/display/GALAXY/Home.

Muller, P, Fleurey, F & J´ez´equel, JM 2005, 'Weaving Executability into Object-Oriented Meta-languages', *In Proceedings of the 8th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, Montego Bay, Jamaica, pp. 264-278.

Munnelly, J, Fritsch, S & Clarke, S 2007, 'An aspect-oriented approach to the modularisation of context' In Proceedings of the PerCom, pp. 114–124.

Muse Website, 2010. http://ws.apache.org/muse/.

Napper, J. & Bientinesi, P. 2009, 'Can cloud computing reach the top500?', in UCHPC-MAW '09: *In Proceedings of the combined workshops on UnConventional high performance computing workshop plus memory access workshop*, ACM, New York, NY, USA, pp. 17–20.

Natis, YV, Clark, W, Valdes, R 2007, 'Context Delivery Architecture: Putting SOA in Context', Gartner Research, ID Number: G00152306, 2007.

Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I. & Löser, A. 2003, Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks, in 'Proceedings of the 12th international conference on World Wide Web', ACM, pp. 536–543.

Neto, RFB, da Graca M & Pimentel C 2005, 'Toward a domain-independent semantic model for context-aware computing', In Proceedings of the Third Latin American Web Congress(LA-Web'05), Washington DC, USA, pp. 61.

Nicoara, A & Alonso G 2005, 'Dynamic AOP with prose', In Proceedings of the International Workshop on Adaptive and Self-Managing Enterprise Applications (ASMEA), pp. 125–138.

Oracle Registry Website, 2010. http://www.oracle.com/us/technologies/soa/service-registry-066462.html.

Oracle Repository Website, 2010. http://www.oracle.com/us/technologies/soa/enterprise-repository-066456.html.

OWL2-QL 2009 http://www.w3.org/TR/owl2-profiles/#OWL_2_QL

OWL2-RL 2009 http://www.w3.org/TR/owl2-profiles/#OWL_2_RL

Pallickara, S., Pierce, M., Fox, G., Yan, Y. & Huang, Y. (2003), `A Security Framework for Distributed Brokering Systems', http ://www.naradabrokering.org.

Papazoglou P., Traverso P., Dustdar S., Leymann F., 2007 'Service-Oriented Computing: State of the Art and Research Challenges', in proceedings of IEEE Computer Journal.

Park, J, Lee, H, Park, P & Shin, D 2008, 'Development of a WS-ECA Rule Management System for a Healthcare Robot', Proceedings of the Computer-Human Interaction 8th Asia-Pacific Conference - APCHI, Seoul, Korea. pp. 149-156.

Parra, C., Blanc, X. & Duchien, L., 2009, 'Context awareness for dynamic service-oriented product lines' In Proceedings of the 13th International Software Product Line Conference. pp. 131–140.

Parsia, B., Halaschek-Wiener, C., Sirin, E.: Towards incremental reasoning through updates in owl-dl. In: Proceedings of the Reasoning on the Web Workshop at WWW2006, Edinburgh, UK. (2006)

Paschke, A.; Kozlenkov, A. & Boley, H. 2007, 'A Homogenous Reaction Rules Language for Complex Event Processing', *in* EDA-PS, ACM, .

Patchube Service Website 2010, http://www.pachube.com/

Peikov I., Grinberg M., Haltakov V., Stefanov H., Kiryakov A., Ognyanoff D., Velkov R. - LarKc Project Deliverable 2.4.2, 2010

Pesic, M., Schonenberg, M.H., Sidorova, N., van der Aalst, W.M.P. 2007, 'Constraint-based workflow models: Change made easy', In Proceedings of the OTM Conferences 2007, vol. (1), pp. 77–94.

Pesic, M., van der Aalst, W.M.P. 2006, 'A declarative approach for flexible business processes management', In: BPM 2006, pp. 169–180.

Petals ESB Website, 2010. http://petals.ow2.org/.

Petals Master Website, 2010. http://petalsmaster.ow2.org

Prezerakos, GN, Tselikas, ND & Cortese, G 2007, 'Model-driven composition of context-aware web services using ContextUML and aspects', In Proceedings of the IEEE International Conference on Web Services, 2007. ICWS 2007, pp. 320–329.

Quesada J., Brandao R., Schooler L., Otte S., Zeng Y., Wang Y., Huang Z., Zhong N. - LarKc Project Deliverable 2.3.2, 2010

Rahman, SS, Aoumeur, N & Saake, G 2008, 'An adaptive eca-centric architecture for agile service-based business processes with compliant aspectual .net environment', In Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services, ACM, pp. 240–247.

Raiciu, C. & Rosenblum., D. S. (2006), Enabling condentiality in content-based publish/subscribe infrastructures, in `Securecomm and Workshops', p. 1-11.

Ranger, D. & Cloutier, J. F. 2005, 'Scalable Peer-to-Peer RDF Query Algorithm', in Proceedings of Web information systems engineering International Workshops (WISE), p. 266.

Ratnasamy, S., Francis, P., Handley, M., Karp, R. & Shenker, S. 2001, 'A Scalable Content-Addressable Network', in Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), ACM, pp. 161–172.

Resource Description Framework (1999). http://www.w3.org/RDF/

RightScale Website (n.d.). www.rightscale.com.

Rodrigues, R. & Druschel, P. 2010, 'Peer-to-peer systems', Communications of the ACM 53(10), 72–82.

Rodriguez, A., McGrath, R., Liu, Y., Myers, J.: Semantic Management of Streaming Data. In: Proc. Intl. Workshop on Semantic Sensor Networks (SSN). (2009)

Rodriguez-Dominguez, C, Benghazi, K & Noguera M 2010, 'Redefinable events for dynamic reconfiguration of communications in ubiquitous computing', *Proceedings of the First International Workshop on Data Dissemination for Large Scale Complex Critical Infrastructures (DD4LCCI '10),* Valencia, Spain, pp. 17-22.

Rong, W., Liu, K. & Liang, L. 2009, 'Personalized Web Service Ranking via User Group combining Association Rule', In Proceedings of the IEEE International Conference on Web Services, pp. 445-452.

Ruth, M, Oh, S, Loup, A, Horton, B, Gallet, O, Mata, M & Tu, S 2007, 'Towards automatic regression test selection for web services', *In the Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC)*, pp. 729–734.

Sánchez M & Villalobos, J 2008, 'A flexible architecture to build workflows using aspect-oriented concepts', In Proceedings of the 2008 AOSD workshop on Aspect-oriented modeling, ACM, pp. 25–30.

Santos, L, Wijnen, R, Vink, P. 2007, 'A service-oriented middleware for context-aware applications' In Proceedings of the 5th international workshop on Middleware for pervasive and ad-hoc computing, pp 37-42.

Schepers, Iacob, Van Eck, 2008, 'A lifecycle approach to SOA governance', in Proceedings of the 25th Symposium On Applied Computing, SAC'08, 2008, March 16-20, pp 1055-1061.

Schmitt M.-T. Schmidt, B. Hutchison, P. Lambros, R. Phippen. 2005, 'The Enterprise Service Bus: Making service-oriented architecture real'. IBM SYSTEMS JOURNAL VOL 44, NO 4, 2005.

Schonenberg, H., Weber, B., Dongen, B., Aalst, W. 2008, 'Supporting flexible processes through recommendations based on history', In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 51–66.

S-Cube, 2008, 'Deliverable # PO-JRA-1.2.1', pp. 38-59, available online at: http://www.s-cube-network.eu/

S-Cube, 2009, 'Deliverable # CD-JRA-1.2.2', pp. 39-56, available online at: http://www.s-cube-network.eu/

ServiceMix Website, 2010. http://servicemix.apache.org.

SETI@home (n.d.). http://setiathome.ssl.berkeley.edu/.

Shao, L. Zhang, J., Wei, Y., Zhao, J., Xie, B., & Mei, H. 2007, 'Personalised QoS prediction for web services via collaborative filtering', in Proc. 5th International Conference on Web Services (ICWS 2007), pp. 439-446.

Sheng, Q & Benatallah, B 2005, 'ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services Development', In Proceedings of the International Conference on Mobile Business (ICMB'05), Sydney, Australia, pp. 206-212.

Sheng, QZ, Nambiar, U, Sheth, AP, Srivastava, B, Maamar, Z & Elnaffar S 2008, 'WS3: international workshop on context-enabled source and service selection, integration and adaptation', In Proceedings of the International Workshop on Context enabled Source and Service Selection, Integration and Adaptation (CSSSIA'08), Beijing, China, pp 1263-1264.

Sheng, QZ, Yu, J & Dustdar, S 2010, 'Enabling Context-Aware Web Services: Methods, Architectures, and Technologies', 1st ed., Chapman and Hall/CRC.

Sheng, QZ, Yu, J,Segev, A & Liao, K 2010, 'Techniques on developing context-aware web services', In Proceedings of the International Journal of Web Information Systems, Vol. 6, Issue 3, pp.185–202.

Shikfa, A., Önen, M. & Molva, R. (2009), `Privacy-Preserving Content-Based Publish/Subscribe Networks', Emerging Challenges for Security, Privacy and Trust pp. 270-282.

Siljee, J, Bosloper, I, Nijhuis, J & Hammer, D 2005, 'DySOA: Making Service Systems Selfadaptive', In the Proceedings of the 3rd International Conference Service-Oriented Computing – ICSOC, pp. 255–268.

SOA4All D1.4.2B deliverable, Hamerling, C., Mathias, E., 2010, to appear.

SOA4All D2.3.1 deliverable, Mos, A., Gonzales-C., R., Pedrinaci, C., Gomez-P., J.M., Ruiz, C., Alvaro Rey, G., Hamerling, C. 2008, 'Service Monitoring and Management Tool Suite Design'. http://soa4all.eu/file-upload.html?func=startdown&id=17.

SOA4All Website, 2010. http://soa4all.eu.

Song, D., Wagner, D. & Perrig, A. (2002), Practical techniques for searches on encrypted data, in `Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on', IEEE, pp. 44-55.

Sonic ESB Website, 2010. http://web.progress.com/en/sonic/sonic-esb.html

Spanoudakis, G & Zisman, A 2010, 'Designing and Adapting Service-based Systems: A Service Discovery Framework', Service Engineering: European Research Results, Springer, pp. 261.

Spanoudakis, G, Zisman, A & Kozlenkov, A 2005, 'A service discovery framework for service centric systems,' *Proceedings of the IEEE International Conference on Service Computing*, vol. 1, pp. 251 - 259.

SpoVNet Project Website 2010, http://www.spovnet.de/

Sreenath, R. M., Singh, M. P. 2004, 'Agent-Based Service Selection', Web Semantics: Science, Services and Agents on the World Wide Web, Vol 1, Issue 3, pp. 261–279.

Srivatsa, M. & Liu, L. (2007), Secure event dissemination in publish-subscribe networks, in `Distributed Computing Systems, 2007. ICDCS'07. 27th International Conference on', IEEE, p. 22.

SRT-15 Project Website 2010, http://www.srt-15.eu/

Staab, S. & Stuckenschmidt, H. 2006, Semantic Web and Peer-to-Peer, Springer.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F. & Balakrishnan, H. 2001, 'Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications', in Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), ACM, New York, NY, USA, pp. 149–160.

Stoitsev, T., Scheidl, S., Spahn, M.2007, 'A framework for light-weight composition and management of ad-hoc business processes', In: Winckler, M., Johnson, H., Palanque, P. (eds.) TAMODIA 2007. LNCS, vol. 4849, pp. 213–226

Strang, T & Linnhoff-Popien, C 2004, 'A Context Modeling Survey', In Proceedings of the Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England.

Stream Project Website 2010, http://www.streamproject.eu/

Tanca, T, Miele, A, & Quintarelli, E 2009, 'A methodology for preference-based personalization of contextual data', In Proceedings of the ACM EDBT 2009,pp. 287-298.

Taylor H., Yochem A., Phillips L. Martinez F. 2009, 'Event Driven Architecture', chapter 2.

Tivoli Website, 2010. http://www-01.ibm.com/software/tivoli/.

Toma, I., Ding, Y., Chalersmook K., Simperl, E., Fensel, D. 2009, 'Utilizing Web2.0 in Web Service Ranking', In Proceedings of the Third International Conference on Digital Society, pp. 174-179.

TRIDEC Project Website 2010. http://www.tridec-online.eu/home

Truong, HL & Dustdar, S, 2009, 'A survey on context-aware web service systems', International Journal of Web Information Systems, Vol. 5, Issue 1, pp.5–31.

Truong, HL, Manzoor, A & Dustdar, S, 2009, 'On modeling, collecting and utilizing context information for disaster responses in pervasive environments', In Proceedings of the first international workshop on Context-aware software technology and applications, pp. 25–28.

Valle, E. D.; Ceri, S.; Harmelen, F. v. & Fensel, D. (2009), 'It's a Streaming World! Reasoning upon Rapidly Changing Information', *IEEE Intelligent Systems* **24**(6), 83--89.

van der Aalst, W, Mooij, A, Stahl, C & Wolf, K 2009, 'Service interaction: Patterns, formalization, and analysis.' Lecture Notes in Computer Science, 2009, vol. 5569/2009, pp. 42–88.

Vanderfeesten, I.T.P., Reijers, H.A., van der Aalst,W.M.P. 2008, 'Product based workflow support: Dynamic workflow execution', In: Bellahsene, Z., Leonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 571–574.

Volz, R., Staab, S., Motik, B.: Incrementally maintaining materializations of ontologies stored in logic databases. J. Data Semantics 2 (2005) 1–34

Voulgaris, S., Rivière, E., Kermarrec, A.-M. & Steen, M. V. 2006, 'Sub-2- sub: Self-organizing content-based publish subscribe for dynamic large scale collaborative networks', in Proceedings of fifthe International Workshop on Peer-to-Peer Systems (IPTPS).

Vukoli, M. 2010, 'The byzantine empire in the intercloud', ACM SIGACT News 41(3), 105–111.

Walavalkar, O.; Joshi, A.; Finin, T. & Yesha, Y. (2008), Streaming Knowledge Bases, *in* 'International Workshop on Scalable Semantic Web Knowledge Base Systems'.

Wang, C., Carzaniga, A., Evans, D. & Wolf, A. (2002), Security issues and requirements for Internet-scale publish-subscribe systems, in `System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on', IEEE, pp. 3940-3947.

Wang, Q, Quan, L & Ying, F 2004, 'Online testing of Web-based applications'*, In the Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC)*, pp. 166–169.

Websphere Website, 2010. http://www-01.ibm.com/software/fr/websphere/.

Williams, GC 1966, 'Adaptation and natural selection: a critique of some current evolutionary thought', Princeton, NJ: Princeton Univ. Press.

WS-Agreement specification (n.d.). http:// www.ogf.org/documents/GFD.107.pdf

WS-BaseNotification specification (n.d.), http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf

WS-BrokeredNotification specification (n.d.), http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf

WSDM specification (n.d.), http://www.oasis-open.org/committees/download.php/20571/wsdm-1.1-os-01.zip

WS-EventDescriptions specification (n.d.), http://www.w3.org/TR/ws-event-descriptions/.

WS-Eventing specification (n.d.). http://www.w3.org/Submission/WS-Eventing/.

WS-Notification specification (n.d.). http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn.

WSO2 Registry Website, 2010. http://wso2.com/products/governance-registry/.

WS-Topics specification (n.d.), http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf

Wu, G, Wei, J & Huang, T 2008, 'Flexible Pattern Monitoring for WS-BPEL through Stateful Aspect Extension', In Proceedings of the IEEE Intl. Conf. on Web Services (ICWS'08), pp. 577- 584.

Xing, Y.; Zdonik, S. & Hwang, J.-H. 2005, 'Dynamic Load Distribution in the Borealis Stream Processor', *in* ICDE '05: Proceedings of the 21st International Conference on Data Engineering, IEEE Computer Society, Washington, DC, USA, pp. 791--802.

Yan, Y., Huang, Y., Fox, G., Pallickara, S., Pierce, M., Kaplan, A. & Topcu, A. (2003), Implementing a prototype of the security framework for distributed brokering systems, in `Proceedings of the International Conference on Security and Management (SAM'03)', Citeseer, pp. 212-218.

Yang, B. & Garcia-Molina, H. 2003, 'Designing a Super-Peer Network, in Proceedings of the 19th International Conference on Data Engineering (ICDE), Vol. 1063, pp. 17–00.

Zhao, Q, Huang, G, Huang, J, Liu, X, Mei, H, Li, Y, Chen, Y 2008, 'An On-the-Fly Approach to Web-Based Service Composition', *Proceedings of Congress on Services Part II, SERVICES-2*, IEEE, pp. 208–209.

Zheng, Z., Ma, H., Lyu, M.R & King, I. 2009, 'WSRec: a collaborative filtering based web service recommendation system', in Proc. 7th International Conference on Web Services (ICWS 2009), pp. 437-444.

Zhou, J, Hall, W & Roure, DD 2009, 'Building a Distributed Infrastructure for Scalable Triple Stores', *Journal of Computer Science and Technology,* vol. 24, no. 3, pp. 447–462.

Zhuge, Y., Garcia-Molina, H.: Graph structured views and their incremental maintenance. In: Proceedings of the Fourteenth International Conference on Data Engineering. (1998) 116 – 125

Zimory Website (n.d.). www.zimory.com.

Annex A. PLAY relevant Projects

Title	Link	Short Description	Topics/ Relevant WPs*	Relation to Use Cases	Strengths	Weaknesses	Tracking Method**	Relevance Degree***	Responsible Partner
STREAM	http://www.streamproject.eu/	Provides scalable streaming using some sort of Online Map Reduce (and some partners based on the Borealis project)	Complex Event Processing, WP3	little (Telco)	Probably very efficient for a small number of fixed event patterns. PLAY can also be used for the "long tail" of CEP needs.	Probably not well suited for large number of patterns with small number of events each.	web-page, meet at EC events	high	FZI
SRT-15	http://www.srt-15.eu/	Building a distributed service platform relying on technologies: content-based routing, complex event processing, dependability and data privacy.	Complex Event Processing, WP3	none	Similar motivation as PLAY: combine federated applications using event infrastructure in the cloud.	-	web-page, meet at EC events	high	FZI
FastFix	http://www.fastfixproject.eu/	Monitoring Control for Remote Software Maintenance. Gathering semantic information on application and user behaviour. This information is sent in real time to a support centre. Identify behaviour patterns using correlation techniques.	Complex Event Processing, WP3	none	unknown	-	web-page	medium	FZI
ESS	http://www.ess-project.eu/	Seems to be quite relevant for our Crisis Management use case	QoS (scalability), CEP, WP3	Little (Crisis management)	Integrate data from various sources into a common information management and communication platform	Limited to the analysis of abnormal events	web-page	medium	AR MINES
RESERVOIR	http://62.149.240.97/	The RESERVOIR project aims at providing an infrastructure as a service cloud layer including massive scale deployment and management features, relying upon federating in a transparent manner using nodes originating from multiple, heterogeneous clouds (e.g., private and public clouds). RESERVOIR wants to take care of setting up and	WP2 / WP5 for PLAY platform hosting and testing	none	Contrary to Reservoir, PLAY is not an IaaS cloud computing research project, even if PLAY aims to be deployable and runnable on such multi cloud infrastructures. Reservoir makes it explicit that these	The Reservoir solution applies at the network and infrastructure layers only, whereas PLAY seeks to be more portable, configurable. Consequently, PLAY aims to address the sought non-functional properties at the	web-page (has ended)	medium	INRIA

		managing the inter-clouds operability (e.g. firewalls crossing) and the heterogeneous QoS of these various clouds (e.g. in the privacy level of the outsourced data, in the level of fault-tolerance, robustness). Research and proposed solutions are said to focus on the infrastructure: how to set up virtual subnetworks, efficiently deploying virtual images, allowing virtual images migration, adjust resource allocation on-demand and in a transparent way. The project goal is to create the basis for future service products in the Future Internet.			clouds offer heterogeneous levels of QoS, in particular, regarding faults, that the PLAY event cloud concept seeks to tolerate as much as possible but from an application level perspective (i.e., within the P2P layer).	application level, without preventing PLAY to cooperatively make use of existing solutions offered by the hosting infrastructure level whenever relevant.			
VISION	http://www.visioncloud.eu/	VISION (Virtualised Storage Services Foundation for the Future Internet) is a specific application of RESERVOIR developed technology, with the aim to provide a scalable, flexible and dependable framework for optimised delivery of data-intensive storage services, in particular bringing computing to storage and overcoming the problem of data lock-in.	WP2	none	Emphasize the importance of managing huge amount of multi-tenant data, spread onto multi-organizational clouds. Contrary to PLAY, data is queried in most traditional database-oriented ways, and not by relying upon a pub/sub event driven way.	Not applicable yet as the VISION project has just started	web page, meet at EC events	low	INRIA
IRMOS	http://www.irmosproject.eu/	Cloud Solutions for interactive real-time applications, and especially multimedia applications	Cloud computing, real-time, WP3	none	They offer real-time guarantees in cloud infrastructure. Amazon and other state of the art cloud providers don't do this.	A specialized virtualization hypervisor is needed to hsot cloud nodes. This hypervisor is not available at current cloud offerings.	web-page	medium	FZI
SOA4All	http://www.soa4all.eu/	Aims to provide a comprehensive framework that integrates complementary and evolutionary technical advances (i.e., SOA, context management, Web principles, Web	SOA and federated Infrastructures, integratio	none	The SOA4All project provides a federated SOA architecture which communicates over the Internet and which can		EBM and INRIA are involved in this project.	high	EBM

		2.0 and semantic technologies) into a coherent and domain-independent service delivery platform. Interesting part - context-based service adaptation.	n of services, WP2, WP5		be extended to Cloud and to events.				
PANDORA	http://www.pandora-project.eu/	Advanced training environment for crisis scenarios.	None	None	-	-	web-page	low	ARMINES
PRONTO	http://www.ict-pronto.org/index.php?id=286	Event recognition for intelligent resource management. Draws methods and expertise from the fields of data fusion, information extraction, temporal representation and reasoning, machine learning and knowledge management systems.	Complex Event Processing, WP3	emergency rescue operations, a fire department is in the consortium	-	-	web-page	medium	FZI
WSAN4CIP	http://www.wsan4cip.eu/	Wireless sensor networks for the protection of critical infrastructures. Increase the dependability of critical infrastructures security, by providing self-healing and dependability modules for the WSAN	None (Project related to sensors (hardware improvement))	None	-	-	web-page	low	ARMINES
SENSEI	http://www.sensei-project.eu/	Common framework of global scale for wireless sensor and actuator networks, made available to services and applications via universal service interfaces. Internet of Things.	Internet of Things, Scalability, Trust and security mechanisms, WP3	Crisis management, transport, smart places Telco companies are involved in the project (Telefóni	Capture and pre-analysis (nature, priority) of numerous events	Privacy issues (e.g. the capture/use of personal data) No use of CEP (the project focuses on the hardware part and the capture of trustable events)	web-page, blog, newsletter	medium (this project focuses on the hardware part of a federated network of sensors and the related aims	ARMINES

				ca I+D, Telenor R+I)				(scalability, trust…) for event capture and their pre-analysis (before sending the events information to the services and applications)	
SAMURAI	http://www.samurai-eu.org/	SAMURAI is to develop a real-time adaptive behaviour profiling and abnormality detection system for alarm event alert and prediction with much reduced false operators and mobile sensory input for patrolling security staff for a hybrid context-aware based abnormal behaviour recognition.	Detecting behaviour in video, real-time, WP3	Maybe useful to monitor crowds in a crisis situation?	Create events from video feeds	Privacy issues because of monitoring public infrastructure?	web-page	low	FZI
SpoVNet	http://www.spovnet.de/	The Spontaneous Virtual Network (SpoVNet) research project develops overlay-based tools and techniques for easy development and spontaneous deployment of distributed network applications and services. SpoVNet follows the approach of providing spontaneous communication by composing algorithms and protocols that allow self-organization in distributed systems. Such self-organizing systems are able to adapt to the given requirements and network loads flexibly and without further	Distributed Complex Event Processing, WP3, Self-organizing overlay network, QoS-aware Publish/subscribe	none	They are writing their own distributed CEP engine. They have their own overlay simulation tool (OverSim).	No focus on event throughput, fault tolerance regarding the P2P layer and group communication primitives.	web-page, possibly personal meeting	high	INRIA/ FZI

		involvement of administrative expenditure.	layer, Group Communication primitives, WP2						
Diadem	http://www.pdc.dk/diadem/	The Prime Objective of the Diadem project is to create an ICT system supporting collaborative situation assessment and decision making for effective protection of the population and the environment against chemical hazards in industrial areas.	Crisis management, WP6	Little (prevent CRBNE crisis)	-	-	web-page	low	ARMINES
TRIDEC	http://www.tridec-online.eu/	TRIDEC focuses on new technologies for real-time intelligent information management in collaborative, complex critical decision processes in earth management. Key challenge is the construction of a communication infrastructure of interoperable services through which intelligent management of dynamically increasing volumes and dimensionality of information and data is efficiently supported; where groups of decision makers collaborate and respond quickly in a decision-support environment.	Crisis management, WP6	Medium (the studied crisis is a tsunami crisis)	Scalability (management of dynamically increasing volume of information and data) Real-time information management	-	web-page	high	ARMINES
weknowit	http://www.weknowit.eu/	The main objective of WeKnowIt is to develop novel techniques for exploiting multiple layers of intelligence from user-contributed content, which together constitute Collective Intelligence, a form of intelligence that emerges from the collaboration and competition among many individuals, and that seemingly has a mind of its own	Development of Event-Model-F, WP3	Emergency Response case study where users provide intelligence about large scale	Experience in reasoning, maybe interesting event format	-	web-page	medium	FZI

				emergencies					
COMPASS	http://www.compas-ict.eu/	The COMPAS project will design and implement novel models, languages, and an architectural framework to ensure dynamic and on-going compliance of software services to business regulations and stated user service-requirements. COMPAS will use model-driven techniques, domain-specific languages, and service-oriented infrastructure software to enable organizations developing business compliance solutions easier and faster	Complex Event Processing, WP3	None, CEP-based compliance monitoring	Tools for compliance monitoring	No specific expertise in CEP	web-page	low	FZI
ALIVE	http://www.ist-alive.eu	ALIVE aimed to combine Coordination and Organisation mechanisms (providing flexible, high-level means to model the structure of inter-actions between services in the environment) and Model Driven Design (providing for automated transformations from models into multiple target platforms) to create a framework for software and services engineering for distributed systems.	WP4, WP6	Crisis Management	They use a workflow enactment module that receives an abstract workflow with some tasks pointing to abstract services rather than to a concrete ones	Possible problems in detecting real- time context changes that may reveal situations that need adaptation, as they don't consider complex events techniques.	Web-page (Finished)	high	ICCS
PADRES	http://msrg.org/projects/padres/	PADRES (Publish/Subscribe Applied to Distributed Resource Scheduling) is an enterprise-grade event management infrastructure that is designed for large-scale event management applications. Ongoing research seeks to add and improve enterprise-grade qualities of the middleware.	Publish/Subscribe based architecture, WP2	none	Full EDA architecture	-	Web page	medium	EBM/FZI/INRIA
Solace Systems	http://www.solacesystems.com/solutions/messaging-	Content-based and semantic routing, Publish/Subscribe service using XML routing	WP2	Telco Use Case (processing many	Performance and intelligent routing	Expensive solution	Web page and contacts with solace systems	medium	FT

	middleware			events)					
DITSEF	http://www.ditsef.eu/	(ICT-SEC-2007-1.0-04 Security systems integration, interconnectivity and interoperability: ICT support for first responders in crisis occurring in critical infrastructures) Digital and innovative technologies for security and efficiency of first responders operation	Crisis management in critical infrastructures, WP6	Low : it focuses on the sensors and the HMI to enhanced the vision of the first responders on the field	-	Concerns only the first responders ?	web-page	low	ARMINES
CRISIS	http://idc.mdx.ac.uk/projects/crisis/	(SEC-2009-4.3-03 Simulation, planning and training tools and methods for management of crisis and complex emergencies) Critical incident management training system using an interactive simulation environment	Simulation of crisis situations for crisis management training, WP6	Very low (no technological solutions)	-	-	web-page	low	ARMINES
EMILI	http://www.emili-project.eu/	(SEC-2009-4.3-03 Simulation, planning and training tools and methods for management of crisis and complex emergencies) Emergency management in large infrastructures	Crisis management in large infrastructures, WP6, Complex Event Processing, WP3	Crisis management	Event Condition Action rules They are writing their own CEP engine	-	web-page	high	ARMINES
EURACOM	http://www.eos-eu.com/EURACOM/tabid/216/Default.aspx	(ICT-SEC-2007-1.0-01 Security systems integration, interconnectivity and interoperability: Risk assessment and contingency planning for interconnected transport or energy network) European risk assessment and contingency planning for	Crisis in energy network, WP6	This is more about risk management approach	-	-	web-page	low	ARMINES

		interconnected transport or energy networks		than crisis management => not relevant for PLAY					
DECOTESSC 1	http://www.decotessc1.eu/	(SEC-2009-1.1-02 CBRNE (Chemical, Biological, Radiological, Nuclear agents and Explosives – Phase 1) Demonstration of counter-terrorism systems-of-systems against CBRNE phase 1	CBRNE crisis, WP6	Low : Focuses on the enhancement of the integrated operational competences rather than on research and development into detailed technological solutions	Analysis and ranking of the gaps between the current situation and the ideal situation of CBRNE system-of-systems counterterrorism	-	web-page	low	ARMINES
WorkPad	http://www.workpad-project.eu/home.jsp	The WORKPAD project (finished in 2009) designed and developed an innovative software infrastructure for supporting collaborative work of human operators in emergency/disaster scenarios.	WP4, WP6	Crisis management	They developed an adaptive process-management system (APMS) as the core element of the front-end middleware for controlling emergency management processes based on contextual information retrieved by a context	They mainly focus on recovering disconnecting nodes (user's PDA) through specific tasks (e.g. assign the "follow disconnected node" task to another node to guarantee the connection)	Web-page (Finished)	high	ICCS

Title	Link	Short Description	Topics/Relevant WPs*	Relation to Use Cases	Strengths	Weaknesses	Tracking Method**	Relevance Degree***	Responsible Partner
					monitor module				
S-Cube	http://www.s-cube-network.eu/	S-Cube is a European Network of Excellence in Software Services and Systems (ends in 2012) and aims to establish an integrated and multidisciplinary research community, in order to help and shape the software-service based Internet of the future.	WP4	Medium (Traffic Case Study, Telecommunication Case Study)	They present adaptation and monitoring techniques that aim to support the prediction and management of the activities of distributed systems that may lead to adaptations, will be the starting point of our PLAY research work in terms of service adaptation.	-	Web-page	high	ICCS
Digital Graffiti	http://dg.jku.at/about.php	Allows users to place smart messages in public and private places using mobile devices (cell phones, PDAs, notebooks)	Complex Event Processing, WP3	None yet.	Interesting use of social media, events and historic events.	unknown	Web-page	high	FZI
Pachube	http://www.pachube.com/	Commercial but free to use online service allowing developers to connect sensor data to the Web. Event sources can be searched and used by others	Complex Event Processing, WP3	none	Platform to subscribe to and publish events, thus creating a marketplace for real-time data	Does not offer any computation (i.e. CEP) on events	Web-page	high	FZI

***** Topics according to the domains involved in PLAY (e.g Complex Event Processing)
** Nature of the tracking/communication/collaboration of the related project, e.g. visit web-page every month, common meetings
*** Relevance to PLAY (i.e. low/medium/high)