Sheffield Hallam University

Application of a new service-oriented architecture (SOA) paradigm on the design of a crisis management distributed system

DOMDOUZIS, Konstantinos <http://orcid.org/0000-0003-3679-3527>, ANDREWS, Simon <http://orcid.org/0000-0003-2094-7456> and AKHGAR, Babak <http://orcid.org/0000-0003-3684-6481>

Published version

DOMDOUZIS, Konstantinos, ANDREWS, Simon and AKHGAR, Babak (2016). Application of a new service-oriented architecture (SOA) paradigm on the design of a crisis management distributed system. International Journal of Distributed Systems and Technologies, 7 (2), 1-17.

Application of a New Service-Oriented Architecture (SOA) Paradigm on the Design of a Crisis Management Distributed System

Konstantinos Domdouzis, Centre of Excellence in Terrorism, Resilience, Intelligence & Organised Crime Research (CENTRIC), Sheffield Hallam University, Sheffield, UK

Simon Andrews, Centre of Excellence in Terrorism, Resilience, Intelligence & Organised Crime Research (CENTRIC), Sheffield Hallam University, Sheffield, UK

Babak Akhgar, Centre of Excellence in Terrorism, Resilience, Intelligence & Organised Crime Research (CENTRIC), Sheffield Hallam University, Sheffield, UK

ABSTRACT

The complexity and the intensity of crisis-related situations require the use of advanced distributed systems infrastructures. In order to develop such infrastructures, specific architectures need to be applied such as Component-based Modelling, Object-Oriented, Aspect-Oriented and Service-Oriented Design. This paper focuses on the use of Service-Oriented Design techniques for the development of the ATHENA Crisis Management Distributed System. The function of the ATHENA Crisis Management Distributed System is based on the use of data generated by social media for the evaluation of the severity of the conditions of a crisis and the coordination of the appropriate measures in response to the crisis. The paper presents a new definition for Service-Oriented Architecture (SOA) and specifies the benefits that are generated by the use of this new definition in the development of the ATHENA system. Useful conclusions are also drawn in relation to how the definition considers the different technical backgrounds of users.

KEYWORDS

Architecture, Aspect, ATHENA, Crisis, Dashboard, Interface, Management, Service

1. INTRODUCTION

A number of natural disasters across the globe (e.g.) has shown the necessity of organized crisis and response systems that will be able to provide automated ways to coordinate the work of search and rescue teams. The complexity of such systems requires the use of an efficient architecture that will allow the adjustment of the crisis and response system to the specific conditions of a crisis. This architecture will allow the flexible modification of the system according to the requirements of the crisis. In this case, there will be efficient use of resources and exact clarification of the roles of the different users of the system, its capabilities and its limitations. The paper explores different system design architecture and focuses how Service-Oriented Architecture can be used but also improved through a new definition in the context of the ATHENA Crisis Response System.

The ATHENA Project is a European Union project that aims to develop a crisis communication and management system that enables the public to communicate in an ethical and lawful way with the emergency search and rescue services during a crisis. The goal of the ATHENA project is the delivery of two major outputs. The first output is the development of a set of guidelines for the police, Law-Enforcement Agencies (LEAs) and the first responders for the use of social media during crisis

situations. The second output is the development of a set of software tools in order to enhance the emergency response options for the search and rescue services using mobile devices in crisis situations. The ATHENA System includes Crisis Command & Control Intelligence Dashboard (CCCID), the Crisis Mobile and the Crisis Information Processing Centre (CIPC). The structure of the ATHENA System is shown in Figure 1.

The Command Control & Intelligence Dashboard (CCCID) has a number of functionalities, such as the real-time crisis monitoring through filtered social media data, the provision of a crisis headlines banner, the provision of crisis statistics and the use of a social content management tool which is used to generate messages through social media. The CCCID is also used for the geo-location of crisis incidents, crowd-sourced reports and key locations such as emergency supplies and medical aid.

The Crisis Mobile is a web service for crisis pre-first responders. It includes the ATHENA Citizen Reporter 'Point & Shoot' system and the Crisis Mobile Receiving Tools. The ATHENA Crisis Information Processing Centre (CIPC) includes the information acquisition and pre-processing tools and the aggregation and analysis tools. The first set of tools includes a social media scanner, a citizen report streaming/recording centre, a speech recognition system, a filter system and a crisis taxonomy system. The second set of tools includes a classification/clearance system, a Formal-Concept Analysis (FCA) system, a data fusion system, a credibility scoring system and a sentiment analysis tool.

The use of the ATHENA CIPC tools allows the real-time acquisition of voice and video messages from crisis-dedicated social media. The filtering of the collected information is based on the use of crisis taxonomies and Twitter hash-tags. The use of text mining techniques (e.g. data fusion, Formal Concept Analysis (FCA) and rule-based inference) is required for data analysis and aggregation. Formal Concept Analysis (FCA) is used for the analysis of data that describe the relationship between a set of objects and a set of attributes. FCA has many applications in biological sciences, linguistics, data mining and semantic searching (Domdouzis et al., 2014).

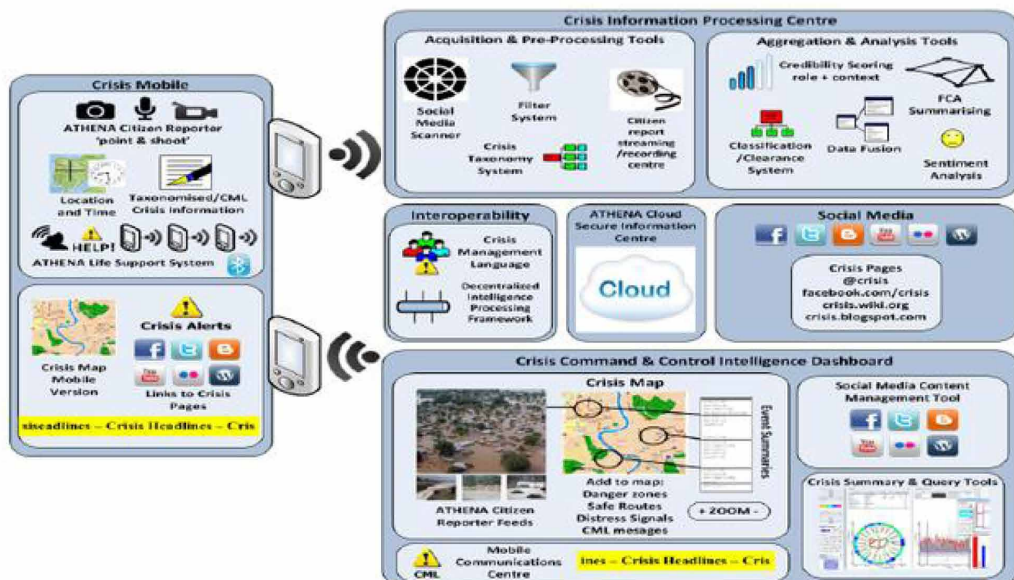Figure 1. The structure of the ATHENA system

Figure 2 depicts how the components of the Crisis, Command, Control and Intelligence Dashboard interact with all the other ATHENA components. The specific figure shows the complexity of the ATHENA system and the different data flows.

According to Figure 2, citizens can send live photos, videos or sound through the use of the Crisis Mobile Citizen Reporter 'Point & Shoot' system. The posted messages are transferred to the ATHENA CIPC Citizen Report Streaming/Recording Centre. Depending on the bandwidth, the messages are transferred to the ATHENA CCCID or the ATHENA Cloud. If geo-located, the ATHENA Cloud data are displayed to the ATHENA CCCID Crisis Map and/or to the ATHENA CCCID Crisis

Figure 2. Interaction of components of crisis, command, control and intelligence dashboard with other ATHENA components
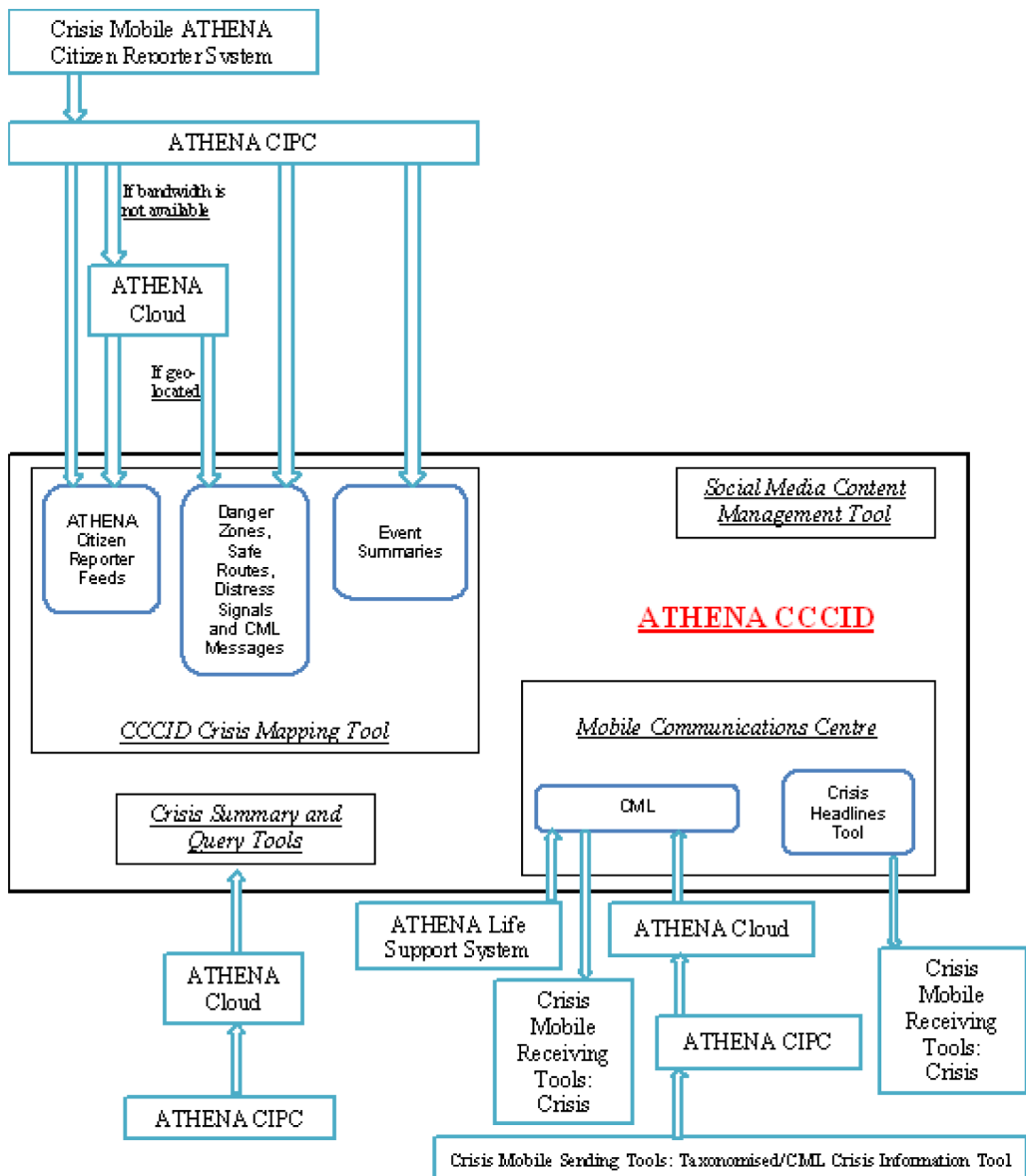
Summary and Query Tools. When the citizens use the Crisis Mobile Sending Tools Taxonomized/ CML Crisis Information Tool, they can send text-based messages using a crisis taxonomy and/or Crisis Management Language (CML) selection/menu system. The messages are transferred to the ATHENA CIPC and since they are text messages, they are filtered using Natural-Language Processing. The filtered messages are then analysed by the ATHENA CIPC Aggregation & Analysis Tools for credibility scoring and sentiment analysis purposes. After the filtering, the messages are stored in the ATHENA Cloud. The messages are then displayed in the ATHENA Cloud Mobile Communications Centre. The police/LEAs/First responders can use the CML button to send messages to the Crisis Mobile Receiving Tools Crisis Alerts. Furthermore, they can use the Crisis Headlines Tool in order to populate and manage the Crisis Headlines Display.

Since the ATHENA system includes a number of interconnected elements, it is necessary that an efficient architectural approach is selected. A number of software engineering approaches have been examined for the design of the ATHENA system. Such approaches are service-oriented design, component-based architecture, object-oriented design and aspect-oriented design. Component-based architecture is a paradigm of overall system design while object-oriented design could be used within an individual component. Furthermore, object-oriented design does not support cross-platform deployments. Component-based architecture allows a component created by one group for one application to be used by another group for a completely different application. Aspect-oriented design is based on the clarification of the various concerns or different areas of interest of a system. Aspect-oriented software development is an effective technique for the modularization of cross-cutting concerns (Kiczales et al., 1997), (Eclipse, 2015).

The architecture of the ATHENA system requires the clarified representation of services among the different elements of the system. The complexity of the system is such that a consistent mechanism that will define the communications among the different elements of the system must be developed. Specifically, it is necessary that this mechanism considers s the different requirements under which communication among different ATHENA system elements is realised.

The selection of SOA as the approach, on which the ATHENA system architecture is based, was made according to a number of criteria. SOA services can be seen as normal software components. The main distinction of these components from common components is that SOA components require business functionalities and this raises the level of abstraction. Furthermore, SOA defines in more detail the software architecture that the services are part of. Specifically, SOA describes that services operate in a distributed environment and the communication among services is document-centred. In contrast, component-based architecture does not consider how components communicate with each other. Aspect-orientation can be seen as a complementary approach. Aspects must be considered when relationships between different components are defined. Therefore, they affect the composition of services and the final structure of SOA (Koskela et al., 2007). The comparison between SOA and object-orientation is not possible. Object-orientation can be part of SOA and SOA can be implemented using object-orientation.

Service-Oriented Architecture offers a number of benefits and especially for the ATHENA project, the SOA benefits are the better representation of the complexity of the system which allows people with non-technical background to easily understand the different functionalities of the system, the re-usability of the different system components and the definition of service contracts that allow the specification of the inputs and outputs to services. Also, the user of a service does not need to know the location or the technology platform of the elements of a service. This allows flexibility in the selection of the technologies that will consist of a service. SOA allows performance optimization by allowing easy introduction of system upgrades in parallel to the operation of existing system components. In this case, the cost is reduced and the overall system performance is not affected (The Open Group, 2014). The use of services allows the handling of multiple requests to multiple service instances (Stevens, 2002). For maintenance purposes, developers can easily locate any defects in the system through the inspection of the system services. The use of SOA allows software developers to have specific roles in the development of the system (Stevens, 2002).

2. SERVICE-ORIENTED ARCHITECTURE (SOA) LITERATURE REVIEW

Service-Oriented Architecture (SOA) is an architectural paradigm for developing distributed applications based on the use of available services over a network (Papazoglou, 2008). It is a paradigm for the organisation and utilisation of distributed services that are owned by different entities. SOA may support the use of different communication protocols but protocols based on open standards are used in current SOA implementations (Ciganek et al., 2009). Zhao et al. (2010) specify that more research needs to be done urgently in the fields of service availability, service re-use and service composition.

Bassil (2012) suggests a three-layer definition of SOA focused on battle command and control systems. The proposed SOA is based on the use of heterogeneous multi-platform components. The first layer represents the client that is represented by the military hardware equipment. The second layer is the server which hosts and runs the different service components that provide the advanced functionalities that are required for the client's hardware equipment. The third layer is the middleware represented by an Enterprise Service Bus which offers a standard interface and a data path for the first and the second layer to interact with each other (Bassil, 2012).

Hoque (2000) defines a service as a software component that contains a collection of related software functionalities that are reusable for different purposes. The service delivers a number of capabilities, such as data storage, scientific data processing, and networking. A model of producer-consumer is the basis for the service. The producer hosts, runs and maintains the service while the consumer connects to the service via a remote method invocation mechanism (Bassil, 2012).

Jones (2005) defines a service as a discreet domain of control that includes a collection of tasks to achieve related goals. There are a number of aspects in which a service could be specified. These aspects are performance, capacity, risks, ownership, security, business impact and service contract (pre-conditions, post-conditions, invariants). SOA uses the consumer-oriented view of a service, in other words a service represents how its consumers would like to use it (Jones, 2005).

Komoda (2006) defines service-oriented architecture as a framework used for the design of information systems. This framework is based on the combination of services. A service is a unit of program that can be called through standardized procedures and which can independently execute assigned functions. Services are managed by a service coordination platform. The flow of execution of services is defined as a business process and is described by the Business Process Modelling Notation (BPMN). Different types of services exist. These are re-usable systems, universal services, compound services (including business processes), services provided by external organizations and newly-developed services.

According to Barry (2003), a service is a function that is well-defined, self-contained and does not depend on the context or the state of other services. The connections that are used among services are based on the use of web services. Web services use XML, EDI standards, CORBA and DCOM (Barry, 2003).

Bih (2006) presents Service-Oriented Architecture as a three-party structure. Specifically, SOA includes the service producers, the service consumers and the broker. The operation of a service requestor is based on two clear business activities: content aggregation and service aggregation. Content aggregation is an activity where a business entity interacts with a variety of content providers in order to process or reproduce such content in the desired presentation format of its customers. Service aggregation is an activity where a business entity interacts with various service providers in order to re-brand, host or offer services to its clients. The operation of a service provider is based on the execution to some degree of an electronic service, such as data processing. The intermediate layer between the service provider and the service consumer is the registry. The registry is a collection of data about other business. The registry comes with a broker. Brokers offer intelligent search capability and business classification or taxonomy data (Bih, 2006).

Bih (2006) and Graham et al. (2005) define the service registry as the linking point between the service consumer and the service provider. The service registry is the directory where the service provider publishes its service description (Bih, 2006), (Graham et al. 2005).

The service provider creates and publishes a service description using a Universal Discovery, Description, and Integration (UDDI) directory so that the published services can be discovered by the service requestors (Monday, 2003), (Pulier and Taylor, 2005).

Table 1 compares the different definitions providing details on their characteristics and their missing elements.

Service-oriented architecture is a way of re-arranging software applications and infrastructure into a set of interacting services. The basic SOA is not an architecture only about services. It is a relationship of three kinds of participants: the service provider, the service discovery agency, and the service client. The interactions among the participants involve the "publish", "find" and "bind" operations. In a typical service-based scenario, a service provider hosts a network accessible software module which is an implementation of a given service. The service provider defines a service description of the service it publishes to a service discovery agency through which a service description is published and is made discoverable. The service requestor uses a 'find' operation to retrieve the service description from the service discovery agency and uses the service description to bind with the service provider and invoke the service or interact with the service implementation (Papazoglou, 2003).

Based on the presented definitions of SOA, it can be concluded that there is no provision of an exact definition of SOA by the literature. All the provided definitions are generic and they do not provide a detailed image of what SOA is. Some definitions present the elements of SOA and the interconnections among them but they do not provide in an exact manner how the communication among these elements is realised. Furthermore, there is not an exact definition of services. The provided definitions do not reveal the exact capabilities of SOA. The way information flows from one service to another is not shown clearly. Although some definitions provide information about the capabilities of services (e.g. data processing), they do not provide information on how these capabilities are realized. For these reasons, a new exact definition of a service and service-oriented architecture (SOA) must be developed. This new definition is presented in Section 3.

3. PRESENTATION OF NEW DEFINITION FOR SERVICES

Services are computational elements that serve as self-contained units of functionality. They support low-cost composition of distributed applications in a rapid manner. Services perform functions that can be either simple requests or complicated business processes. SOA uses services for the development of applications and solutions (Papazoglou and van den Heuvel, 2003). Service-Oriented Architecture (SOA) is a software architecture that starts with the definition of an interface and continues with the building of the entire application topology as a topology of interfaces, interface implementations and interface calls. SOA shows the relationships between services and service consumers. Services are software modules that are accessed by name via an interface in a request-reply mode. Service consumers are the client representations of the interfaces (Natis, 2003).

A service can be represented as a set of interfaces and the mechanisms that enable the communication of these interfaces. Figure 3 shows a new definition of how a service is used in order to enable the communication between two users.

As Figure 3 shows, a service includes SOA interfaces, service contracts and the communication mechanism between the service contracts. A service contract allows the user to initiate a service. It is a mechanism and not an interface. However, in order to access this mechanism, an interface is used which is called SOA interface. A SOA interface allows the users to define the type of data that they want to transmit. A SOA interface is a Graphical-User Interface (GUI) interface that allows the users to specify the syntax and type of data and above all, access the service contract. The service contract allows the connection of the SOA interface to a communication mechanism that elaborates the transmitted data and passes them to other users. It is the connection between a SOA interface and the intermediate communication mechanism. The intermediate communication mechanism differs

Table 1. Comparison of SOA definitions

Definition	Authors	Characteristics	Missing Elements
SOA is an architectural paradigm for developing distributed applications based on available services over a network.	Papazoglou (2008)	Generic definition of SOA	The definition does not provide an exact specification of how SOA-based distributed applications are developed.
Three-Layer Definition of SOA based on battle and control systems.	Bassil (2012)	Definition is characterized by complexity while it is restricted on applications on the battlefield and on the use of specific technologies, such as satellites.	The definition describes a SOA architecture that allows the communication of military equipment with web services that perform algorithmic operations. It does not however describe whether military equipment of a specific type can communicate with military equipment of a different type.
A service is a software element that includes a reusable collection of software capabilities.	Hoque (2000)	Generic definition of SOA	The definition does not provide a detailed description of SOA.
A service is a discreet domain of control that includes a number of tasks to achieve related goals.	Jones (2005)	Although the definition refers to the independence of SOA as a discreet unit, it does not detail how it achieves its goals and which tasks it includes.	Jones (2005) describes areas associated with SOA but he does not provide a detailed definition for SOA.
SOA is a framework that is used for the design of information systems. The framework is based on the use of services which can be called through standardized procedures and that can independently execute assigned functions.	Komoda (2006)	Services can be re-used, added or replaced. Information systems are built based on a combination of services.	The definition does not provide information on how a service is administered by the service execution platform. It describes how services can communicate with each other but not what a service is.
Use of XML-based web services as the connection link between the service provider and the service requestor.	Barry (2003)	The definition focus on the three-layer SOA structure which considers a web service as the intermediate layer between a service requestor and a service provider.	Vague definition of SOA. It does not provide any analysis of how the different elements of a SOA communicate with each other.
SOA is a three-party structure that includes service providers, service consumers and an intermediate layer called registry.	Bih (2006)	Three-layer structure with service providers, requestors and brokers. A registry is a form of broker.	The definition does not specify how the three operations (publish, find, bind) are realized.

Figure 3. New definition of how a service is used to enable communication between two users
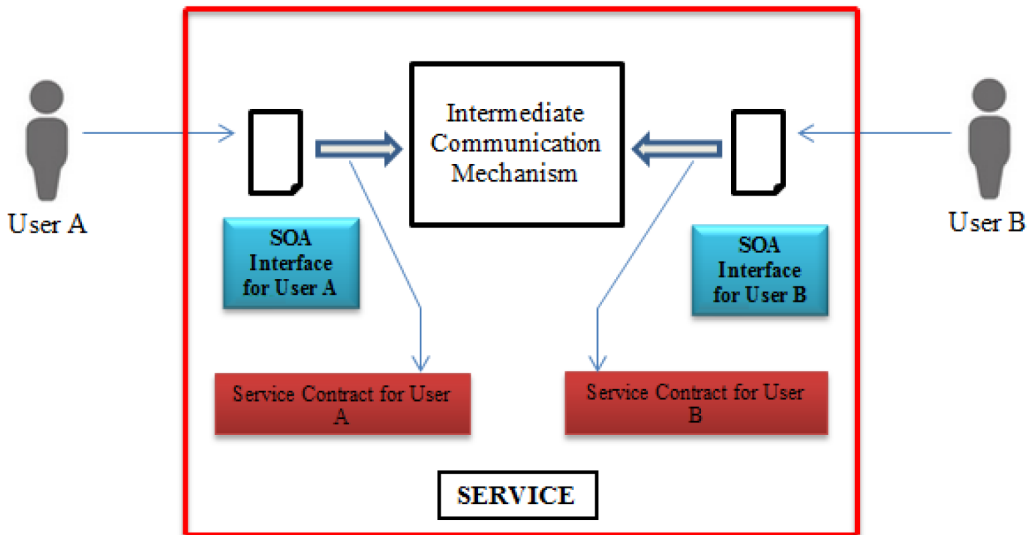
depending on the requirements of the scenarios to which the presented SOA definition is applied. The design of the SOA interfaces, the service contract and the intermediate communication mechanism changes in order to adapt to the requirements of the specific problem to which they are implemented.

The structure of SOA that is shown in Figure 3 considers the dynamic requirements of a crisis. Disaster environments are characterized by increased difficulty in the inter-organizational and inter-jurisdictional coordination. When disaster threatens a community, different responses from different organizations are required in a simultaneous manner (Comfort et al., 2001). The different conditions that characterize a crisis require the use of an adaptive information processing system that will provide synchronization among the agents that face the crisis. The efficient handling of resources and the coordination of the search and rescue teams as well as the as much as possible efficient communication with the citizens are the main characteristics of the steps that need to be taken during a crisis. The new SOA definition that is presented in Figure 3 provides the needed automated adaptation of software systems to the requirements of a crisis. The definition adopts the three tier-model that is presented in most of the SOA definitions found in the literature. In contrast to the existing definitions however, the new SOA definition provides a clearer way of the different elements of SOA and how SOA can be applied to different scenarios. Furthermore, the new SOA definition is not restricted to a specific software or communication technology. The new SOA definition includes the technologies that are presented by the other definitions but it is not restricted by them.

4. ATHENA SYSTEM SERVICE-ORIENTED DESIGN

There are three stages in the development of a service-oriented system. These stages were followed for the development of the Service-Oriented Architecture for the ATHENA System and they are presented below. The new definition presented in Section 2 impacts the three stages of SOA development as it improves the definition of service contracts. Additionally, the notation used that is used to depict the different stages of SOA development and which is based on the proposed SOA definition, is more accurate than the notation used in the literature to describe the SOA development. The new

definition also improves the time required for the realization of the three stages as it provides to the SOA developers a clear image of what to aim for in each stage.

4.1. Business Process Modelling Notation (BPMN)

Business Process Modelling Notation (BPMN) was created by BPMI and has emerged as a standard notation for process modelling, along with other notations, such as UML Activity Diagrams, Icam Definition for Function Modelling (IDEF) and ebXML. BPMN allows the bridging of the gap between business process design and business process implementation (Abramowicz at al., 2007). A business process is the combination of a set of activities within an enterprise with a structure that describes the logical structure of these activities. A business process model can be based on a business map (Nurcan et al., 2005). A business map includes a number of paths from 'start' state to 'end' state, each of them representing a business process model (Edirisuriya, 2009). An enterprise can be analysed and integrated through its business processes. A common understanding of a process can be provided by a process model. Business Process Modelling enables a common understanding and analysis of a business process (Aguilar-Savén, 2004).

4.2. Business Architecture Modelling (BAM)

Business Architecture can be defined as an architecture that takes into consideration the people, the processes, the functions that support the processes, the business information that support the processes, usability that considers the usability aspects of the system and its environment, and performance which considers the performance aspects of the system and its environment (Ulrich, 2015). Business Architecture Modelling (BAM) is used for the identification of the business perspectives of a Service-Oriented Architecture, the acquisition of its business requirements and the specification of the business community and its services. The steps of BAM include the specification of the participants in a specific service, the services architectures and the service contracts which represent the collaborations/transactions between the different participants (Elvesæter, 2011).

4.3. System Architecture Modelling (SAM)

System Architecture Modelling (SAM) represents the Information Technology perspectives of SOAs. It segments the system into different software components and interfaces. SAM defines how these interfaces are used, which components use which interfaces and what interfaces these components provide (Elvesæter et al., 2011). SAM is based on structural and behavioural modelling. Structural modelling specifies system components, their interfaces and their dependencies. Behavioural modelling specifies component interactions and protocols.
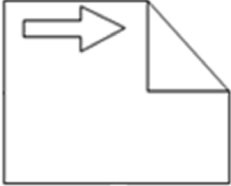
System Architecture Modelling defines the structure of the system in terms of configurations of system services and the interactions among these services in order to provide the necessary business services related to the system's business requirements. This type of modelling includes rules related to information dynamics and processing that should be used by the system (Lloyd and Peckham, 2003).

5. APPLICATION OF NEW SERVICE DEFINITION TO THE DESIGN OF THE ATHENA SYSTEM ARCHITECTURE

The ATHENA Command and Control Intelligence Dashboard interacts with a number of components within the ATHENA system. Furthermore, some of the components of CCCID interact with each other. The ATHENA CCCID Service-Oriented Architecture is based on the use of SOA Mark-Up (SoaML) scripting language and the use of the Modelio Software (Modeliosoft, Paris) and the EDrawMax ver. 7.6 (EdrawSoft, HongKong) software. Based on SoaML, the notation that is shown in Table 2 has been identified.

The paper focuses on the Business Process Modelling Notation (BPMN) and the Business Architecture Modelling (BAM) stages as these are considered necessary for the identification of

Table 2. BPMN notation

Symbol	Role	Symbol	Role
○	Start of a process		Data Warehouse
●	End of a process	·······>	Association
	Data Insertion	⟶	Sequence Flow
	Activity		Service Contract
	Output of an activity		Service

services related to the ATHENA CCCID. The next sections describe an example of using the SOA definition described in Section 3 for the identification of the service-oriented architecture of the ATHENA Citizen Reporter Feeds which is an element of the ATHENA Crisis Command & Control Intelligence Dashboard (CCCID) Mapping Tool as described in Section 4 of the paper.

The ATHENA CCCID Mapping Tool includes the ATHENA Citizen Reporter Feeds and the CrisisMap. The ATHENA Citizen Reporter Feeds element of the CCCID is responsible for the collection of user data through the transfer of images, videos and sound files. The Crisis Map displays information based on crowd-sourced crisis data combined with geo-spatial and temporal information. The map also displays automated summaries, event reports, danger zones, and safe routes. A symbology is used to define the level of data credibility. A zoom facility in the interface will allow the focusing to specific areas.

5.1. Service-Oriented Architecture of ATHENA Citizen Reporter Feeds

Figure 4 shows the Business Process Modelling Notation (BPMN) diagram that describes the sequence of steps that are realized when the user posts a message using the ATHENA Citizen Reporter 'Point & Shoot' button till it reaches the ATHENA CCCID Citizen reporter Feeds. Citizens can acquire live photos, videos or sounds and the acquired files can be transferred to the ATHENA Crisis Information Processing Centre (CIPC). If there is not available bandwidth for streaming, the files are sent for pre-processing and they are then stored to the ATHENA Cloud. If these files are geo-located, then they become available to the ATHENA Citizen Reporter Feeds.

The Business Architecture Modelling (BAM) stage includes the following steps:

Figure 4. BPMN diagram
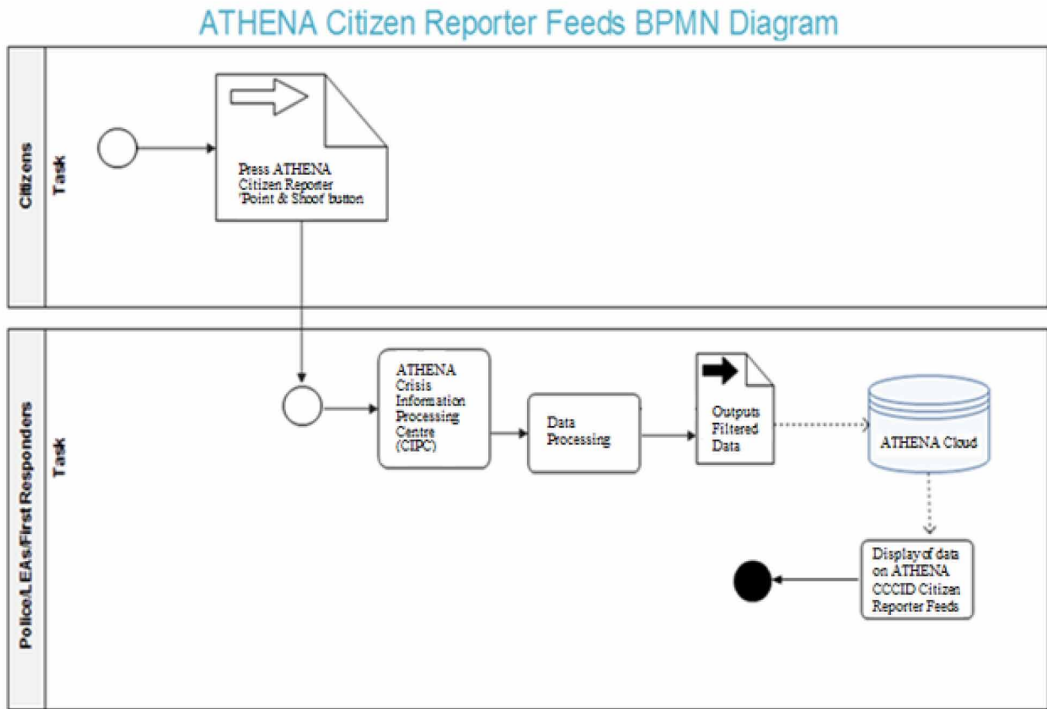
1. **Identification of Service Contracts:** Service contracts are the fundamental elements of a service architecture. Service contracts show the connection between the service consumers (in the case of the ATHENA Citizen Reporter Feeds, the consumers are the police/Law Enforcement Agencies (LEAs)/First Responders) and the service producers (in the case of the ATHENA Citizen Reporter Feeds, these are the citizens). The service contracts specify the bi-directional interaction between the citizens and the police/Law Enforcement Agencies (LEAs)/First responders. The consumers request data while the producers initiate the production of data, thus the beginning of a service. Using SoAML notation, these two service contracts are shown in Figures 5 and 6 respectively.

Figure 5 shows that the citizens use the ATHENA 'Point & Shoot' system which is on the Crisis Mobile. The citizens need to press the ATHENA 'Point & Shoot' system Button in order to send a message to the police/Law Enforcement Agencies (LEAs)/First responders. Thus, the specific service contract which is entitled "ATHENA Citizen Reporter 'Point & Shoot' system Button' service contract shows how the citizens communicate with the police/LEAs/First responders through the 'Point & Shoot' system and this is achieved by pressing the 'Point & Shoot' Button.

The service contract that is used by the police/Law Enforcement Agencies (LEAs)/First responders for the purpose of receiving the citizens' data in the ATHENA CCCID Citizen Reporter Feeds is depicted in Figure 6.

Two interfaces are used that allow the interaction of the consumers and the producers with their respective service contracts and these interfaces are used to bind the citizens (producers) and the police/Law Enforcement Agencies (LEAs)/First responders (consumers) to the service contract. These are the SOA interfaces for the ATHENA CCCID Citizen Reporter Feeds.

The interface between the 'Citizens' and the ATHENA 'Point & Shoot' System is specified in Table 3.

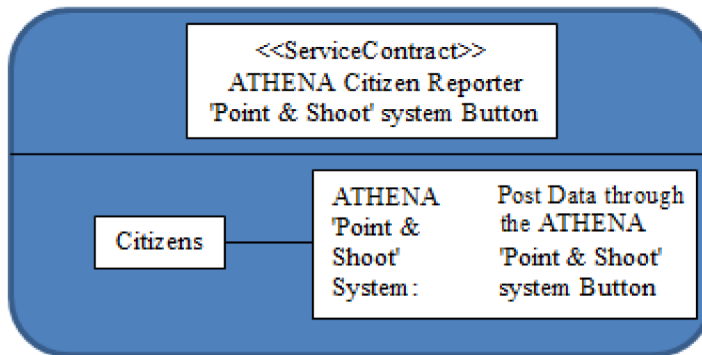**Figure 5. SoaML notation service contract ('Point & Shoot')**

Figure 6. SoaML notation service contract ('Citizens')
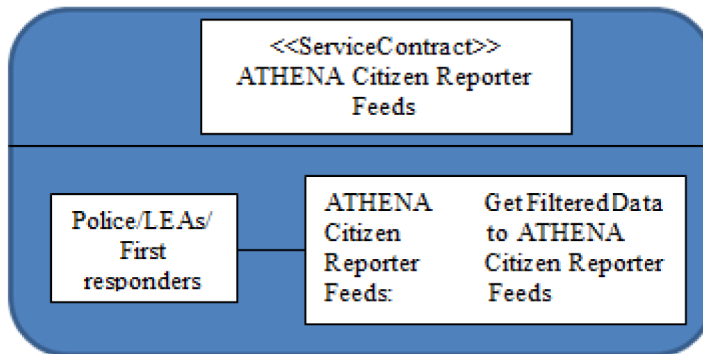

Table 3. SOA interface of the 'ATHENA citizen reporter 'point & shoot' system button' service contract

| <<interface>>
ATHENA Citizen Reporter 'Point & Shoot' system Button |
| --- |
| ButtonToSendLivePhoto
(in ATHENA 'Point & Shoot' System: Photos) |
| ButtonToSendLivePhoto
(in ATHENA 'Point & Shoot' System: Videos) |
| ButtonToSendLivePhoto
(in ATHENA 'Point & Shoot' System: Sounds) |

The interface between the 'Citizens' and the ATHENA 'Point & Shoot' System needs also to be specified as in Table 4.

2. **Identification of Service Architecture:** The service contracts identified in Step 1 for both the citizens and the police/LEAs/First responders are connected together through the Data Processing & Storage in order to create the service for the ATHENA CCCID Citizen Reporter Feeds. The Data Processing & Storage as can be seen from the BPMN Diagram of Figure 4 includes the

Table 4. SOA Interface of the 'ATHENA citizen reporter feeds' service Contract

<<interface>> ATHENA Citizen Reporter Feeds
DisplayOfFilteredLivePhoto (in ATHENA Citizen Reporter Feeds: Photos)
DisplayOfFilteredLiveVideo (in ATHENA Citizen Reporter Feeds: Videos)
DisplayOfFilteredLiveSound (in ATHENA Citizen Reporter Feeds: Sounds)

Figure 7. Interaction between 'Point & Shoot' and 'Citizens' interfaces
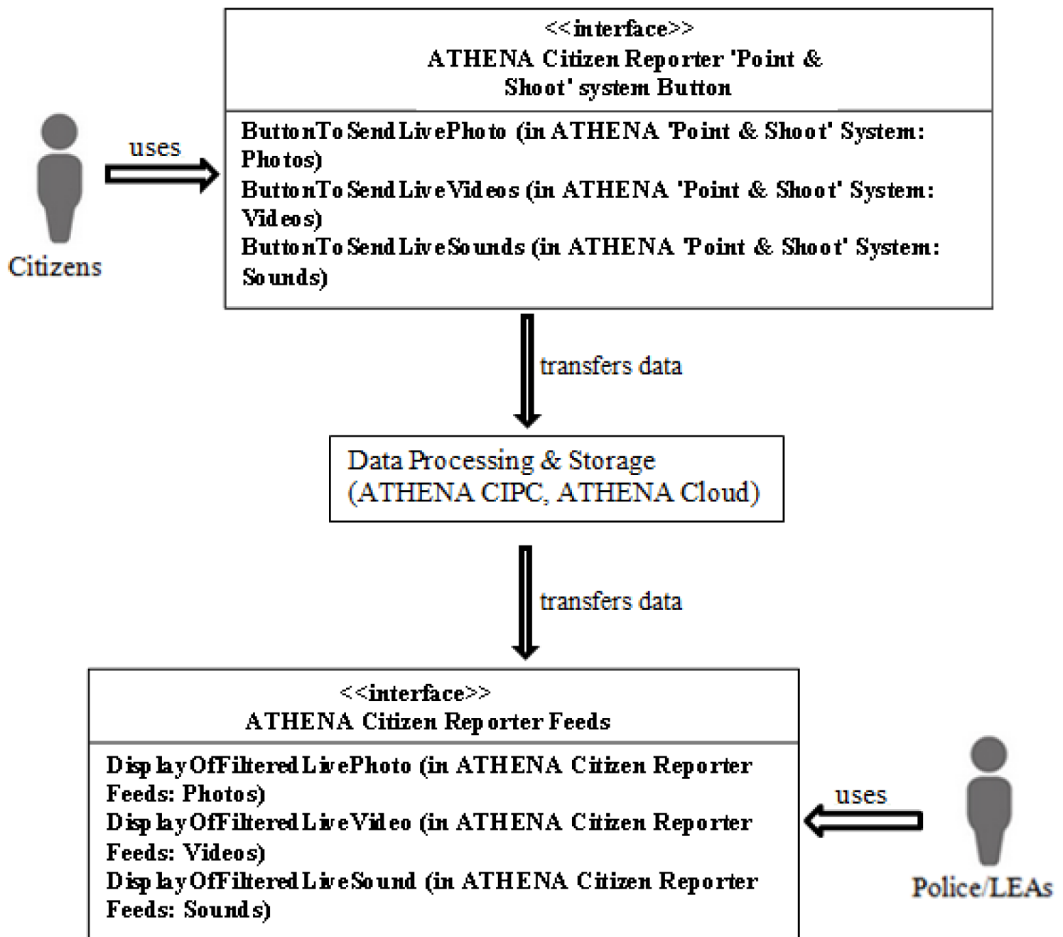
ATHENA Crisis Information Processing Centre (CIPC) and the ATHENA Cloud. Figure 8 shows how citizens and the police/Law Enforcement Agencies (LEAs)/First responders interact through their interfaces. The service contacts identified in Step 1 together with the Data Processing & Storage are the "ATHENA CCCID Citizen Reporter Feeds" Service.

Figure 8 shows the ATHENA CCCID Citizen Reporter Feeds Service. The service is the mechanism that enables the interaction between the citizens and the police/Law Enforcement Agencies (LEAs)/First responders and it includes the ATHENA Citizen Reporter 'Point & Shoot' system Button service contract, the Data Processing & Storage (that includes the ATHENA CIPC and the ATHENA Cloud) and the ATHENA Citizen Reporter Feeds service contract.

The main advantage of using the definition of SOA as it is presented in Section 2 of the paper is that it allows better clarification in the way different components of a SOA service interact with each other. This is important as it allows users, especially with non-technical background, to understand the flow of data among the different components of the system. The new SOA definition provides a better image of the structure of the system as it allows the choice of the most appropriate elements and it optimizes its performance as it allows system administrators to replace components quickly if they do not operate properly and see whether and how possible replacements of existing system components can affect the overall operation of the system. The new definition also allows the design of better data flows that optimize the speed of operation of the system. Furthermore, it allows better re-usability of system components to different applications and it allows flexibility in the selection of the technologies that will consist of a service. In addition, the new SOA definition can be used for the development of more complex distributed applications as it can be used for the definition of the connections among the different services.

The implementation of the presented SOA definition requires updated knowledge from the developer. Specifically, in order for the SOA definition to be adjusted to the requirements of a specific

Figure 8. ATHENA CCCID citizen reporter feeds service
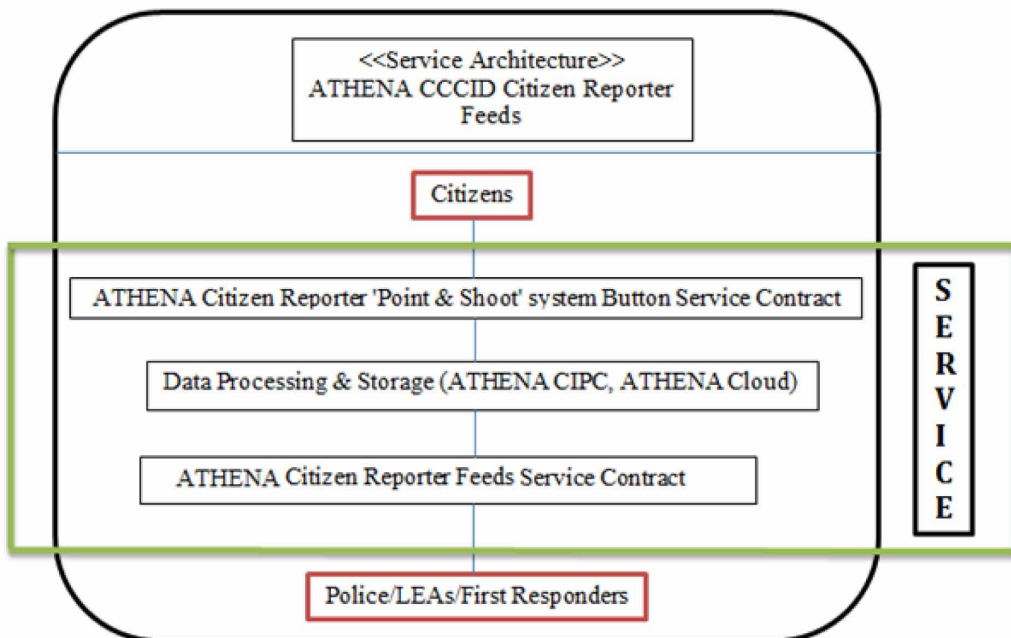
scenario, specific technologies need to be selected and applied. In order for these technologies to be selected, the developer needs to have an updated technical knowledge of the available technologies in the market and how he/she can implement them. Furthermore, it is possible that the proposed SOA definition may not be accepted by the police and LEAs as these people may have learnt to administer crises in a specific way.

6. CONCLUSION

The paper focuses on the development of a new, more detailed definition of a service used in an advanced Service-Oriented Architecture (SOA). The definition aims to cover gaps in the existing knowledge for Service-Oriented Architecture that characterise previous definitions. These gaps have significant impact in the clarification of the structure of distributed systems. The new service definition considers the fact that not all users have the same technical background, therefore it identifies the elements that need to be presented to the different users in order to access and use a service. Users can access a service through a SOA interface and unless the communication mechanism between the elements of the service is clarified, then the interface cannot provide the necessary access mechanism to the users. Especially when it comes to users without sufficient technical expertise, it becomes even more difficult for them to comprehend how to access a service. Thus, in terms of user-friendliness, the new service definition clearly identifies the elements that need to be offered to the different users in order to use a service.

The new definition for a service presented in this paper has been used for the development of a Service-Oriented Architecture for the ATHENA Crisis Management Distributed System. The ATHENA system will be used by members of the public and the police, the Law Enforcement Agencies (LEAs) and first responders during intense crisis situations, such as earthquakes, floodings, etc. The new definition for a service provides for the ATHENA system a flexible services architecture both from the point of view of the ATHENA system administrator(s) and also for the common users. This can be shown by the fact that for the ATHENA system administrator(s), the service-oriented architecture provides capabilities of better re-usability and maintenance of the system, while for the common users, it provides enhanced user-friendliness and more efficient communication with rescue services during crisis situations. The new definition also allows the adaptation of the ATHENA service-oriented architecture to new scenarios as it considers the dynamic features of each crisis situation separately. The service contracts which are fundamental elements of a service are the dynamic links of the service with real-life conditions as they consider the different characteristics of the environment to which the service is aimed for.

REFERENCES

Abramowicz, W., Filipowska, A., Kaczmarek, M., & Kaczmarek, T. (2007). Semantically enhanced Business Process Modelling Notation.*Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management SBPM '07).*

Aguilar-Savén, R. S. (2004). Business process modelling: Review and framework. *International Journal of Production Economics*, *90*(2), 129–149. doi:10.1016/S0925-5273(03)00102-6

Barry, D. K. (2003). *Web Services and Service-Oriented Architecture: The Savvy Manager's Guide*. San Francisco, CA: Morgan Kaufmann; doi:10.1016/B978-155860906-8/50005-6

Bassil, Y. (2012). Service-Oriented Architecture for Weaponry and Battle Command and Control Systems in Warfighting. *International Journal of Information and Communication Technology Research*, *2*(2).

Bih, J. (2006). Service oriented architecture (SOA) a new paradigm to implement dynamic e-business solutions. *Ubiquity*, 2006.

Ciganek, A. P., Haines, M. N., & Haseman, W. D. (2009). Service-oriented architecture adoption: Key factors and approaches. *Journal of Information Technology Management*, *XX*(3), 42–54.

Comfort, L. K., Sungu, Y., Johnson, D., & Dunn, M. (2001). Complex Systems in Crisis: Anticipation and Resilience in Dynamic Environments. *Journal of Contingencies and Crisis Management*, *9*(3), 144–158. doi:10.1111/1468-5973.00164

Domdouzis, K., Andrews, S., & Gibson, H. (2014, December 8-11). Service-Oriented Design of a Command and Control Dashboard for Crisis Management.*Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC)*, London (pp. 202-207). doi:10.1109/UCC.2014.114

Eclipse. (2015). AspectJ. Retrieved from https://eclipse.org/aspectj/

Edirisuriya, A. (2009). *Design Support for e-Commerce Information Systems using Goal, Business and Process Modelling* [Doctoral Thesis]. Stockholm University and Royal Institute of Technology, Stockholm, Sweden.

Elvesæter, B. (2011). Service Modelling Service Modelling with SoaML. Retrieved from http://www.uio.no/studier/emner/matnat/ifi/INF5120/v11/div/SoaML_Tutorial.pdf

Elvesæter, B., Carrez, C., Mohagheghi, P., Berre, A.-J., Johnsen, S. G., & Solberg, A. (2011). Model-driven Service Engineering with SoaML. In *Service Engineering - European Research Results* (pp. 25–54). Wien: Springer.

Graham, S., Davis, D., Simeonov, S., Daniels, G., Brittenham, P., Nakamura, Y., & Zentner, C. (2005). *Building Web Services with Java – Making sense of XML, SOAP, WSDL, and UDDI* (2nd ed.). United States: Sams Publishing.

Hoque, F. (2000). e-Enterprise: Business Models, Architecture, and Components. Cambridge University Press: Cambridge

Jones, S. (2005). Toward an Acceptable Definition of Service. *IEEE Software*, *22*(May/June), 87–93. doi:10.1109/MS.2005.80

Koskela, M., Rahikainen, M., & Wan, T. (2007). Software development methods: SOA vs. CBD, OO and AOP. In *Seminar on Enterprise Information Systems: Service-Oriented Architecture and Software Engineering*. Helsinki University of Technology.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J., & Irwin, J. (1997, June 9-13) Aspect-oriented programming.*Proceedings of the European Conference on Object-Oriented Programming (ECOOP)*, Jyväskylä, Finland (pp. 220-242).

Komoda, N. (2006). Service Oriented Architecture (SOA) in Industrial Systems.*Proceedings of the 2006 IEEE International Conference on Industrial Informatics* (pp. 1-5). doi:10.1109/INDIN.2006.275708

Koskela, M., Rahikainen, M., & Wan, T. (2007). *Software development methods: SOA vs. CBD, OO and AOP. Proceedings of the seminar on Enterprise Information Systems: Service-Oriented Architecture and Software Engineering*, Helsinki University of Technology.

Lloyd, S., & Peckham, J. (2003). *Practicing Software Engineering in the 21st Century*. Hershey, PA, USA: IRM Press; doi:10.4018/978-1-931777-50-6

Monday, P. B. (2003) Web Service Patterns: Java Edition (eBook). Springer. Retrieved from http://www.springer.com/computer/swe/book/978-1-59059-084-3

Natis, Y. V. (2003). Service-Oriented Architecture Scenario (Technical Report, ID Number: AV-19-6751). Retrieved from http://info.lnpu.edu.cn/website/jpkc/rjgc/ywzl/Service-Oriented%20Architecture%20Scenario.pdf

Nurcan, S., Etien, A., Kaabi, R., Zoukar, I., & Colette, R. (2005). A strategy driven business process modelling approach. *Business Process Management Journal*, *6*, 2005.

The Open Group. (2014). Service Oriented Architecture: SOA Features and Benefits. Retrieved from http://www.opengroup.org/soa/source-book/soa/soa_features.htm#soafabmsgsec

Papazoglou, M. P. (2003). Service -Oriented Computing: Concepts, Characteristics and Directions.*Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE'03)*, Washington, United States. doi:10.1109/WISE.2003.1254461

Papazoglou, M.P., & van den Heuvel, W.J. (2003). Service-Oriented Computing: State-of-the-Art and Open Research Issues. *IEEE Computer, 40.*

Ulrich, W. (2015). Business Architecture and SOA. Retrieved from http://www.bainstitute.org/resources/articles/business-architecture-and-soa

Papazoglou, M. P. (2008). *Web Services: Principles and Technology. Harlow*. England: Pearson.

Pulier, E., & Taylor, H. (2005). *Understanding Enterprise SOA*. Manning Publications.

Stevens, M. (2002). The Benefits of a Service-Oriented Architecture. Retrieved from http://www.developer.com/services/article.php/1041191/The-Benefits-of-a-Service-Oriented-Architecture.htm

Zhao, X., Wang, T., Liu, E., & Clapworthy, G. J. (2010, January-March). Web Services in Distributed Information Systems: Availability, Performance and Composition. *International Journal of Distributed Systems and Technologies*, *1*(1), 1–16. doi:10.4018/jdst.2010090801