

# Towards Automated Composition of Convergent Services: a Survey

Armando Ordóñez<sup>a,\*</sup>, Vidal Alcazar<sup>b</sup>, Oscar Mauricio Caicedo Rendon<sup>c,e,\*</sup>,  
Paolo Falcarin<sup>d</sup>, Juan C. Corrales<sup>e</sup>, Lisandro Zambenedetti Granville<sup>c</sup>

<sup>a</sup>*Intelligent Management Systems Group - University Foundation of Popayán  
Calle 5 No. 8-58 - Popayán, CA Colombia*

<sup>b</sup>*Planning and Learning Group - Universidad Carlos III  
Av. Universidad, 30, 28911 - Leganes, MA Spain*

<sup>c</sup>*Institute of Informatics - Federal University of Rio Grande do Sul  
Av. Bento Gonçalves, 9500 - Porto Alegre, RS - Brazil*

<sup>d</sup>*School of Architecture, Computing and Engineering - University of East London  
4-6 University Way, E16 2RD, London UK*

<sup>e</sup>*Grupo de Ingenieria Telematica- Universidad del Cauca  
Calle 5 No. 4-70 - Popayán, CA - Colombia*

---

## Abstract

A convergent service is defined as a service that exploits the convergence of communication networks and at the same time takes advantage of features of the Web. Nowadays, building up a convergent service is not trivial, because although there are significant approaches that aim to automate the service composition at different levels in the Web and Telecom domains, selecting the most appropriate approach for specific case studies is complex due to the big amount of involved information and the lack of technical considerations. Thus, in this paper, we identify the relevant phases for convergent service composition and explore the existing approaches and their associated technologies for automating each phase. For each technology, the maturity and results are analysed, as well as the elements that must be considered prior to their application in real scenarios. Furthermore, we provide research directions related to the convergent service composition phases.

---

\*Corresponding author.

*Email addresses:* [armando.ordonez@docente.fup.edu.co](mailto:armando.ordonez@docente.fup.edu.co) (Armando Ordóñez), [valcazar@inf.uc3m.es](mailto:valcazar@inf.uc3m.es) (Vidal Alcazar), [omcrendon@inf.ufrgs.br](mailto:omcrendon@inf.ufrgs.br) (Oscar Mauricio Caicedo Rendon), [falcarin@uel.ac.uk](mailto:falcarin@uel.ac.uk) (Paolo Falcarin), [jcorral@unicauca.edu.co](mailto:jcorral@unicauca.edu.co) (Juan C. Corrales), [granville@inf.ufrgs.br](mailto:granville@inf.ufrgs.br) (Lisandro Zambenedetti Granville)

*Keywords:* Automated composition, Computer Networks, Mashups, Telecom2.0, Web Services

---

## 1. Introduction

The telecommunication and computer network industries are facing a great shift from atomic services (*e.g.*, voice and email) towards much more dynamic, convergent services, such as unified social networking and integrated video conferencing. Under this premise, telco and computer network operators need to transform themselves from simple operators into services aggregators carrying the duty of providing services that bring together numerous applications from a variety of sources [1].

The term “convergent” has been used widely in telecommunication and computer networks with two meanings: to describe the fusion of fixed and mobile services [2], or as a synonym of the Next Generation Network (NGN) that is a packet-based network able to transport all type of information and services, like voice, data, and video, integrating traditional telephony networks and Internet [3]. In this paper, we use the term “convergent” as efficient coexistence of services available on the Telecom (*e.g.*, voice and video calls with reverse charging offered by telco operators like Orange and Movistar) and Web (*e.g.*, services of maps and RSS feeds offered by big Internet players like Google and Yahoo) domains to provide more services to the user.

Telecom services can be created by using Services Creation Environments (SCE) [4] and Mashup Development Environments (MDE) [5] [6] that allow the users (*e.g.*, programmers and administrators of telco and/or computer networks) to combine the functionalities of heterogeneous services using drag-and-drop and wire tools for composing new services. Such a way of carrying out service composition is valid as long as the number of traditional communication services is low and the reliability of services is high enough ( $> 99\%$ ) so they do not change continually [7].

In the case of the convergent service composition, that involve atomic services from the Web and Telecom domains, such services may change their interfaces, become available or unavailable, and their number may also grow to unmanageable sizes. Both the dynamic nature of services offered by the Web and the sheer amount of relevant services that may be available on it for a given task hinder greatly the use of approaches offering semi-automated

service composition (*i.e.*, composition based on MDEs and SCEs). If time is a crucial factor, semi-automated service composition may become inefficient, which raises the need of approaches able to perform fully-automated service composition.

Automation is defined as the delegation of tasks to machine or computer systems, thus reducing procedural load and freeing operators from vigilance over routine and tedious tasks [8]. Several approaches have appeared to fully automate the composition of services available in the Web from both academia [9] [10] [11] [12] [13] [14] and industry [15] [16]. Similarly, some research has revolved around fully automation of communication service composition, coming mainly from European Projects [17] [18]. Considering the evolution of service composition in the Telecom and Web domains, we raise the first research question addressed in this paper: *Which is the level of automation that has been achieved to date from previous service composition approaches?*

In addition to the aforementioned automation, it is important to consider that the service composition carried out in the Telecom fundamentally different of the conducted in the Web, which means that some technical issues must be contemplated when combining services from both domains [19] [20]. Moreover, several other characteristics pertaining to specific application domains, like user preferences, response time, adaptability, and scalability must be taken into account. This is because the fact that convergent service composition often involves the challenge of merging two areas of knowledge with their respective concepts, technologies, and architectures. For this reason, a systematic way of analysing the strengths and weaknesses of each existing solution for a particular application domain is needed. In this way, we raise the second research question addressed in this paper: *Which elements must be considered for applying automated services composition in different specific domains?*

Convergent service composition is also a complex process that often involves several elaborated steps or phases, from processing the input to selection of services to its execution. As these phases often focus on one or more aspects of the service composition process, it is needed to classify them. In this way, we raise the third research question addressed in this paper: *Which are the phases for convergent service composition and the approaches useful in each one of them?*

The above raised research questions indicate that building up a convergent service is not trivial, because although there are significant approaches

that aim to automate the service composition at different levels in the Web, telco networks, and computer networks, selecting the most appropriate approach(es) for specific case studies is complex due to the big amount of involved information and the lack of technical considerations. Thus, in this paper, we identify the relevant phases for convergent service composition and explore the existing approaches and their associated technologies for automating each phase. For each technology, the maturity and results are analysed, as well as the elements that must be considered prior to their application in real scenarios. Furthermore, we provide research directions in the convergent service composition phases.

The research conducted to answer the raised questions led to the following contributions:

- Identify the relevant phases for convergent service composition.
- Explore and analyze the existing approaches and their associated technologies for automating each phase.
- Provide research directions in each one of convergent service composition phases.

The remainder of this paper is organized as follows. In Section 2, we present the methodology followed to build up this survey. In Section 3, we review foundations about service composition. In Section 4, we introduce the phases for convergent service composition. In Section 5, 6, and 7, we describe and discuss each phase and its associated approaches. In Section 8, we present research directions. Finally, in Section 9, we provide some conclusions that help to select approaches for convergent service composition.

## 2. Methodology

To solve the research questions, we follow the next steps [21]: *(i)* formalize the service composition process by defining different phases, *(ii)* classify the existing approaches under the umbrella of requirements and criteria for their application in convergent composition; and *(iii)* define the guidelines for employing diverse approaches and their technologies in convergent service composition.

In this paper, we carry out a comparative literature review [22], aiming to identify the existing approaches in service composition in recent years

through the lens of its applicability in convergent composition of services. In order to select the works presenting the approaches useful for our study, we introduced search keywords on three electronic databases: ACM, IEEE, and Science Direct. As convergent composition is a relatively new field, we do not expect to find many publications using the exact keyword. Therefore, we used the following keyword phrases: “convergent service composition”, “automatic service composition”, “automated service composition”, “Telecom 2.0 service composition”, and “mashup composition”. We found 250 works. Afterwards, we looked through the titles of such works to eliminate those not related to the field; for the rest, we browsed the abstracts to estimate their relevance and whether they have experimental results. We selected both recent works from recognized conferences and relevant ones from high impact journals. As a result, 65 works were selected for a full-text review.

Table 1 and Table 2 present, in no order of relevance, the works selected for reviewing. These works are analysed according to: the phase (*i.e.*, adaptation, creation, execution, and generation), the origin of the proposed (*i.e.*, academy, industry, and from both in alliance), the domain (*i.e.*, Web, Telecom, and convergent - deals with methods for convergent composition), the level of automation provided to users (*i.e.*, high, medium, low; when users participate little in the process or needs to know little about technical issues, higher is the automation), the use of semantic in any of the phases above mentioned, and the result of the work (*i.e.*, prototype, model, product, and standard). The detail of the literature review as well as the rest of the analysis carried out in this paper are presented in the following chapters.

Table 1: Works selected for reviewing

Work	Phase	Origin	Domain	Automation	Semantic	Result
[23]	Creation	Both	Convergent	Medium		Prototype
[24]	Creation, execution	Both	Convergent	Medium	✓	Prototype
[25]	Creation	Academic	Convergent	Medium		Prototype
[26]	Creation, execution	Both	Convergent	Medium		Prototype
[27]	Creation, execution	Both	Convergent	Medium	✓	Prototype
[28]	Adaptation, execution	Academic	Convergent	High	✓	Prototype
[29]	Adaptation, execution	Academic	Convergent	High	✓	Prototype
[30]	Creation, execution	Academic	Convergent	Medium	✓	Prototype
[31]	Creation, execution	Academic	Convergent	Medium	✓	Prototype
[32]	Creation, execution	Both	Convergent	Medium	✓	Model
[33]	Creation, execution	Industry	Convergent	Medium	✓	Prototype
[34]	Creation, execution	Academic	Convergent	Medium		Model
[35]	Creation, execution	Academic	n/a	High	✓	Prototype
[36]	Creation, execution	Academic	n/a	High	✓	Prototype

Table 2: Works selected for reviewing

Work	Phase	Origin	Domain	Automation	Semantic	Result
[37]	Creation	Both	Convergent	Medium	✓	Prototype
[38]	All except adaptation	Academic	Convergent	High		Prototype
[39]	All except adaptation	Academic	Convergent	High	✓	Prototype
[11]	All except adaptation	Academic	Web	High	✓	Prototype
[15]	Creation	Industry	Convergent	High		Product
[16]	Creation	Industry	Convergent	High		Product
[40]	Creation	Industry	Convergent	Low	✓	Standard
[41]	Creation	Industry	Web	Low	✓	Standard
[42]	Creation	Industry	Web	Low	✓	Standard
[43]	Creation	Industry	Telecom	Medium		Product
[44]	Creation	Both	Convergent	Medium	✓	Prototype
[45]	Execution	Academic	Convergent	n/a		Prototype
[46]	Execution	Academic	Convergent	n/a	✓	Prototype
[47]	Creation	Industry	Web	n/a	✓	Standard
[48]	Execution	Academic	Convergent	n/a		Prototype
[49]	Adaptation	Academic	Convergent	High		Prototype
[50]	Adaptation	Academic	Convergent	High		Prototype
[51]	Creation	Industry	Convergent	Medium	✓	Prototype
[52]	Adaptation	Academic	Convergent	High		Prototype
[53]	Adaptation	Academic	Convergent	High		Prototype
[17]	Creation	Academic	Convergent	Medium	✓	Prototype
[54]	Execution	Industry	Convergent	n/a		Product
[55]	Execution	Industry	Convergent	n/a		Product
[56]	Execution	Industry	Convergent	n/a		Product
[57]	Execution	Industry	Convergent	n/a		Product
[58]	Execution	Industry	Telecom	n/a		Product
[19]	Execution	Industry	Convergent	n/a		Prototype
[59]	Execution	Industry	Convergent	n/a		Prototype
[60]	Execution	Academic	Convergent	n/a		Standard
[61]	Execution	Industry	Communication	n/a		Standard
[62]	Execution	Industry	Convergent	n/a		Standard
[63]	Execution	Industry	Convergent	n/a		Standard
[10]	Generation, execution	Academic	Web	High		Prototype
[12]	Creation, execution	Academic	Convergent	Medium		Prototype
[13]	Creation, execution	Academic	Convergent	Medium		Prototype
[14]	Creation, execution	Academic	Convergent	Medium		Prototype
[6]	Creation, execution	Academic	Convergent	Medium		Prototype
[64]	Generation	Academic	Web	High		Prototype
[65]	Generation	Academic	Web	High	✓	Prototype
[66]	Generation	Academic	Web	High	✓	Prototype
[67]	Generation	Academic	Web	High	✓	Prototype
[68]	Generation	Academic	Telecom	High	✓	Prototype
[69]	Generation	Academic	Web	High	✓	Prototype
[70]	Generation	Academic	Web	High		Prototype
[9]	Generation	Academic	Web	High	✓	Prototype
[71]	Generation	Academic	Web	High	✓	Prototype
[72]	Creation, generation	Academic	Web	High	✓	Prototype
[73]	Generation	Academic	Web	High	✓	Prototype
[5]	Creation, execution	Academic	Convergent	Medium		Prototype
[74]	Creation, execution	Academic	Convergent	Medium		Prototype
[75]	Creation, execution	Academic	Convergent	Medium		Prototype
[76]	Creation, execution	Academic	Web	Medium	✓	Prototype
[77]	Creation, execution	Academic	Web	Medium	✓	Prototype
[78]	Creation, execution	Academic	Web	Medium	✓	Prototype

### 3. Foundations on Service Composition

In this section, we introduce the concepts of service composition and convergent service composition. Furthermore, we present the differences among the services offered in the domains of Telecom and Web as well as current

approaches for describing such services.

### 3.1. What is a service?

The term “service” is a commonplace in many fields and therefore different definitions can be found in the literature: in the workflow area, the concept of e-service is commonly associated with the completion of a task [79]; meanwhile, in the Web, a Web service can be seen as a way to expose the functionality of an information system and make it available through standard Web technologies [80]. More precisely, the World Wide Web Consortium (W3C) - an organization for developing Web standards - defines a Web service as a software system identified by a URI, whose public interfaces and bindings are defined and described using the eXtensible Markup Language (XML) or the JavaScript Object Notation (JSON). Such a service can be discovered by other software systems which, in turn, may then interact with the Web service in the manner prescribed by its definition, using XML or JSON based messages conveyed by Internet protocols [81]. Usually, in computer science, most authors refer to both “e-services” and “Web services” by simply “services” [82].

On the other hand, the term “service” in the Telecom domain can be understood as the capability to exchange information through a telecommunication medium, provided to a user by a service provider [83]. Different standardization associations in this domain have created similar definitions. One of the most important associations for standardization in Telecom is the Tele-Management Forum (TMF) that describes a service as a set of independent functions being an integral part of one or more business processes. This functional set consists of the hardware and software components as well as the underlying telecommunication medium. The user sees all these components as a fusion; a service can be a service component of another service [84]. Other association of standards for mobile technologies is the 3rd Generation Partnership Project (3GPP). The 3GPP defines a service a component of the portfolio of choices offered by service providers to a user, in the sense of functionalities provided to a user [15].

Basic Telecom services (*i.e.*, networking services) provide connectivity to a network, like voice over the Internet Protocol (IP) and basic signalling, for example. Added-value services, a broader concept of service, provide other functionalities such as: voice/video connectivity, community tools, presence, conferencing, gaming, and TV broadcasting. The added-value services are based on NGN [82], whose purpose is to deploy next generation architectures

that allow the access to independent services over converged fixed and mobile networks.

To sum up, the fundamental objective of services is to have a collection of network-resident software services accessible via standardized protocols, whose functionality can be discovered, used individually or composed to form more complex services [79].

### 3.2. How can a service be described?

As stated before, an atomic service consists of a set of network-accessible functionalities described using a machine understandable description. The set of methods that allow accessing these functionalities is known as the interface of the service. The description of service interfaces will be used afterwards for service discovery, compatibility, verification, and composition. The first step to take when describing a service is to analyse its inputs and outputs; Figure 1 shows diagram of the input and output of a service that performs the sum of two integers. Currently, such description depends on the architectural style (*i.e.*, the Service Oriented Architecture - SOA - and the the REpresentational State Transfer- REST) and the technologies used for implementing the service.

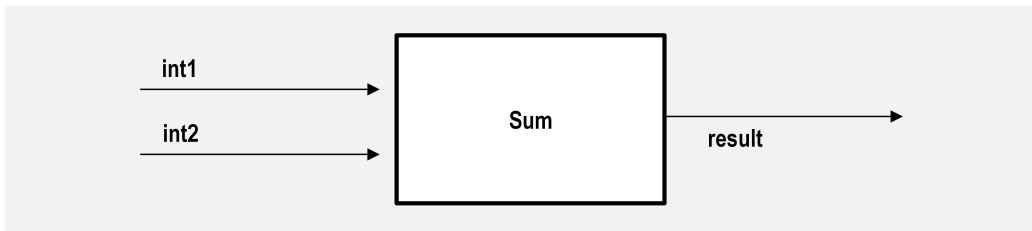


Figure 1: Description of services using inputs and outputs

The Web Services Description Language (WSDL) [85] is the facto standard for describing services based on SOA that are usually implemented using the Simple Object Access Protocol (SOAP) and XML. WSDL [86] provides a computer-understandable description of Web services and describes Web services as collections of communication end points that can exchange certain messages. This description language includes atomic method calls (also called operations) in terms of input and output messages where each operation can be a one-way operation, request-response, solicit-response or notification.



### Listing 1: WSDL Example

```

<definitions name="greeting_service"
  targetNamespace="http://www.unicauca.edu.co/wsd/greeting_service.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsd/"
  xmlns:soap="http://schemas.xmlsoap.org/wsd/soap/"
  xmlns:tns="http://www.unicauca.edu.co/wsd/greeting_service.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <message name="SayHelloRequest">
    <part name="name" type="xsd:string" />
  </message>
  <message name="SayHelloResponse">
    <part name="greeting" type="xsd:string" />
  </message>
  <portType name="greeting_portType">
    <operation name="sayHello">
      <input message="tns:SayHelloRequest" />
      <output message="tns:SayHelloResponse" />
    </operation>
  </portType>
  <binding name="greeting_binding" type="tns:greeting_portType">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="sayHello">
      <soap:operation soapAction="sayHello" />
      <input>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:examples:helloservice"
          use="encoded" />
      </input>
      <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:examples:helloservice"
          use="encoded" />
      </output>
    </operation>
  </binding>
  <service name="Greeting_Service">
    <documentation>WSDL Description for Greeting Service</documentation>
    <port binding="tns:Hello_Binding" name="Hello_Port">
      <soap:address location="http://www.unicauca.edu.co/Greeting/">
    </port>
  </service>
</definitions>

```

Listing 1 depicts an example of WSDL. The code starts with the definition of the namespaces and the definition of messages to exchange (*i.e.*, *SayHelloRequest* and *SayHelloResponse*). The portType includes the allowed operations (*i.e.*, *SayHello*). The Binding (*i.e.*, *greeting\_binding*) specifies a particular protocol and the encoding style for provision of the service. Finally the service element (*i.e.*, *greeting\_service*) includes information about the concrete address of the service.

Unlike the services based on SOAP and XML, there is still not a widely accepted standard for describing services based on the architectural style REST that are usually implemented using the HyperText Transfer Protocol (HTTP) and JSON or XML. Relevant efforts for describing REST-based services are: (i) the Web Applications Description Language (WADL) [87] that offers a machine process-able protocol description format for services using HTTP and XML, (ii) the WSDL 2.0 [88] that provides a machine-readable description for SOAP and HTTP based services by accepting the binding to all the HTTP-request methods (not only GET and POST); and (iii) the HTML for REST (hREST) [89] that offers a XHTML-based microformat for representing and describing Application Program Interfaces (APIs) and services based on REST.

The second level of service description is concerned with pre- and post-conditions. As aforementioned, description solutions, such as WSDL (1.1 or 2.0), WADL, and hREST, describe a single interaction with a service in terms of request and response messages. However, to assess whether a service can be used by other services, it does not suffice to know the type of data generated or received by a service, but also the change in the world that it produces. Pre-conditions must be true in the input in order to the service to be executed; at the same time, post-conditions are conditions that become true after the service is successfully executed. For example, a service called *send\_message* may require as input a well-formed email address and may have the effect (post-condition) of informing someone of something, which represents the change in the world.

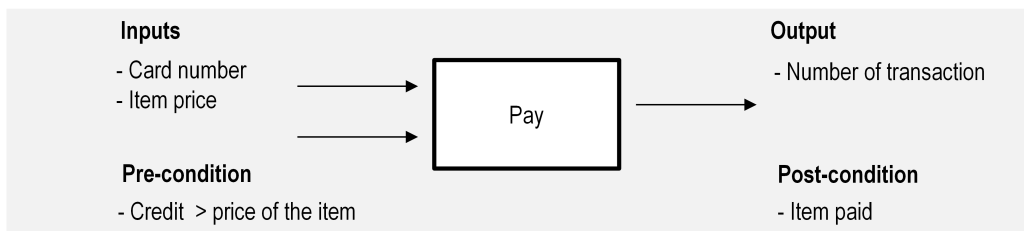


Figure 2: Description of services using pre and post-conditions

Figure 2 depicts the service *pay* that has as input the payment value and the payment data (*e.g.*, card number and customer) as well as an additional pre-condition. Such precondition ensures that the available credit is higher

than the amount to pay. A post-condition for the service *pay* may be to change the state of the transaction to *payed*, for instance. Some languages, such as the Web Ontology Language for Services (OWL-S), the Semantic Annotations for REST (SA-REST), and the micro Web Service Modeling Ontology (microWSMO) [90] [89], allow the annotation of services with pre- and post- conditions apart from the inputs and outputs. It is important to highlight that currently any of these languages is dominant in the second level of service description.

### *3.3. What are the differences between services available in the Web and Telecom domains?*

Telecom services are fundamentally different from Web services based on SOA or REST [17]. Next the most remarkable differences are analysed:

***Standard language for service description.*** In the Web domain, for services based on SOAP, there is a large consensus on the main standard for service description: WSDL. Unlike, for services based on REST, there is not a widely approach/standard accepted for describing them. In the same direction of REST-based services and contrary to SOAP-based services, in spite of some initiatives such as ONEAPI [61] and Twilio [58] for standardizing service interfaces, there is no standard language for service description in the Telecom domain. A review of proposals shows that service description languages in Telecom are often protocol specific, not extensible nor reusable [53].

***Level of detail of services.*** In the Web, the details of atomic services are most of the time open to the world by means of REST and SOAP APIs, widgets, and Web front-ends. That is to say that the users are often aware of the parameters of the demanded services. Two prominent examples of highly parameterized Web services whose APIs are public are for example Google Translator and Google Maps. On the other hand, Telecom operators offer services without authorizing the user for configuring them in deep or even note that he/she is working with them. For example, services such as call forwarding or billing processes are often completely opaque to users.

***Availability of services.*** Availability refers to the ability to access a system or service. In the case of Web services provided through the Internet, the availability depends highly from the underlying networks. Conversely, the percentage of availability of Telecom services is very high, and terms such as 5 x 9 (*i.e.*, 99.999% availability) and fault tolerance are commonly used [91].

**Time constraints.** In the Telecom domain, there are tight real-time requirements in protocols and user requirements (*e.g.*, post-dial delay is typically bounded [19]). In contrast, a best-effort response time is typically offered by services available along the Web.

**Number of protocols.** In the Web most of the communications are based in synchronous protocols, mostly in HTTP. Conversely, in Telecom, there is a broad range of protocols, each one specific for some services (*e.g.*, voice transfer and video) or networks (*e.g.*, Wi-Fi, mobile, and fixed). In spite of the efforts for integrate all the protocols over the Session Initiation Protocol (SIP), that allows phone calls to be made over the Internet [92], current technologies require the implementation of diverse protocols in order to be able to be integrated with legacy hardware and software.

#### 3.4. What is service composition?

In a broad sense, the process of creating a composite service by combining atomic services is known as “service composition”. This composition is required when the request of a client cannot be satisfied by any single service. Due to the differences in standards, service descriptions and formalization of the involved processes, service composition is subsequently described separately for the Web and Telecom domains.

**Service composition in the Web domain.** According to the previously given description, service composition in the Web consists of joining the individual services and defining the control flow and the data flow, similar to workflow approaches. The resulting composition can be viewed from two perspectives: orchestration and choreography. Orchestration describes the executable process that can interact through messages with both internal and external Web services. Orchestration includes business logic and task-execution orderings from one partner perspective.

Choreography tracks the sequences of messages among multiple partners rather than in a specific process executed by a single partner. Choreography is more collaborative than orchestration and allows to describe the public message exchange between services. Also, choreography is useful for establishing the rules for the exchange of messages in order to guarantee interoperability among different agents. In summary, orchestration describes an executable process while choreography specifies the protocol of peer-to-peer interactions. Figure 3 illustrates the relationship among both concepts.

The main language for describing the composition of SOAP-based services is the Web Services - Business Process Execution Language (WS-BPEL)

[47], often shortened to BPEL in many works. Business process is a term used to describe the interaction between services in terms of work and data flow. BPEL is an orchestration language that models business processes that interact with external parties. In BPEL each part is a service described using WSDL. BPEL includes representation in XML of concepts such as data, control flow, data flow, error handling and partners among others.

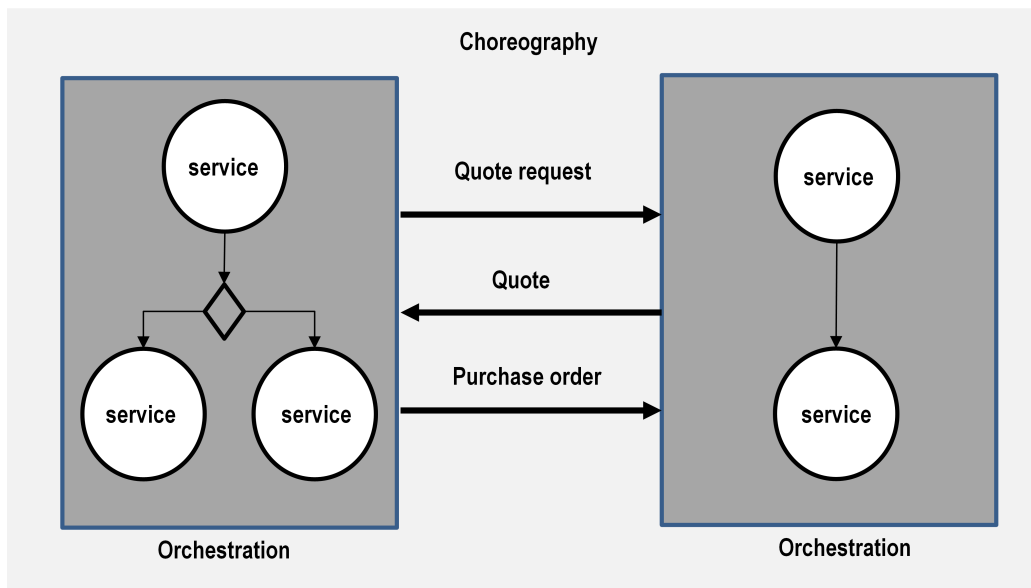


Figure 3: Orchestration vs Choreography

Listing 2 presents an example of a whole process in BPEL. In this example, a process called `AmericanAirline` defines the `partnerLinks` (services involved in the process) as well as the variables of the process. The BPEL-based workflow of process is a sequence of task that includes: receive, assign, and invoke (make a call-back). It is important to note that the element `invoke` includes parameters related to the service description defined in the WSDL such as `operation` and `portType`.

BPEL does not include any graphical representation for business processes, so other standards have appeared to provide such functionality. The most used notation for graphical representation of business processes in service composition is the Business Process Model and Notation (BPMN) [93]. BPMN is a standard that provides a graphical notation for specifying busi-

ness processes and is very similar to the activity diagrams of the Unified Modelling Language (UML) [94]. The goal of BPMN is offering a representation tool for both technical and business users. To do so, BPMN provides a notation that is intuitive to business users, yet able to represent complex process semantics. The BPMN specification provides also a mapping to particular languages such as BPEL. Figure 4 depicts an example of modelling with BPMN.

Listing 2: Example of a BPEL process

```

<process name="AmericanAirline" targetNamespace="http://packtpub.com/
  service/airline/"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:aln="http://packtpub.com/service/airline/" >
  <partnerLinks>
    <partnerLink name="AmericanAirlines" partnerLinkType="aln:flightLT"
      partnerRole="airlineCustomer" myRole="airlineService" />
  </partnerLinks>
  <variables>
    <!-- input -->
    <variable name="FlightDetails" messageType="
      aln:FlightTicketRequestMessage" />
    <!-- output from American Airlines -->
    <variable name="FlightResponseAA" messageType="
      aln:TravelResponseMessage" />
  </variables>
  <sequence>
    <!-- Receive the initial request -->
    <receive partnerLink="AmericanAirlines" portType="
      aln:FlightAvailabilityPT"
      operation="FlightAvailability" variable="FlightDetails"
      createInstance="yes" />
    <!-- Prepare the output -->
    <assign>
      <copy>
        <from>
          <FlightNo>123</FlightNo>
          <ReturnDateTime>2004-01-05</ReturnDateTime>
          <Approved>true</Approved>
          </confirmationData>
        </from>
        <to variable="FlightResponseAA" part="confirmationData" />
      </copy>
    </assign>
    <!-- Make a callback -->
    <invoke partnerLink="AmericanAirlines" portType="aln:FlightCallbackPT"
      operation="FlightTicketCallback" inputVariable="
      FlightResponseAA" />
  </sequence>
</process>

```

Unlike the SOAP-based services, for composing REST-based services there is still no a language standardized or broadly accepted. Significant researches are: (i) BPEL for REST [95] proposes an extension of BPEL

that covers the composition of Web Services based on SOAP and REST by adding specialized activities as  $\langle get \rangle$ ,  $\langle post \rangle$ , and  $\langle resource \rangle$ , (ii) Bite [96] presents a lightweight and extensible composition language to create workflows where the main entities of compositing are REST-based services; and (iii) the Lightweight Process Modeling Language (LPML) [97] allows to model a business process, that uses Web 2.0 solutions, by means a subset of BPMN.

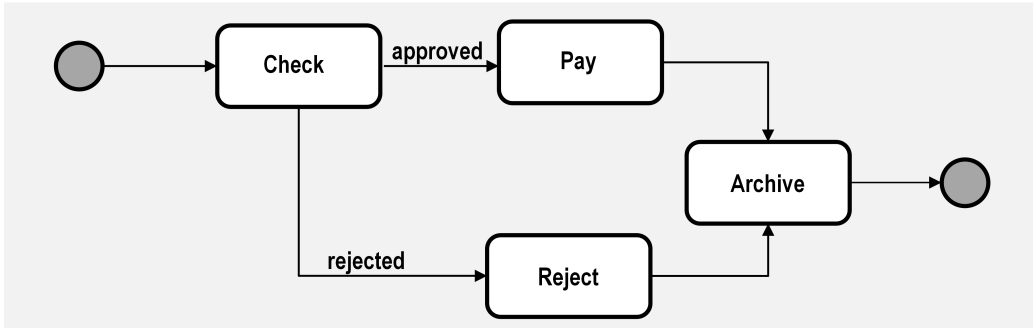


Figure 4: Simplified example of a process modelled using BPMN

On the other hand, regardless of the method used to represent the composite service, service composition in the Web is commonly performed at design time. In this sense, recently, non-technical users have begun to play an active role in service composition, especially by creating their own services, called mashups, in the Web 2.0. Mashups are composite Web applications focused on end-users and created by combining different resources available along the Web [98] [99]. In this sense, MDEs like Yahoo Pipes (see Figure 5) allow users to create their own services by assembling service building blocks.

MDEs work with a rather limited set of services in comparison with all ones available on the Web, but at least offer some degree of personalization/-customization to users. Still, service composition made by users has some problematic issues such as the heterogeneous, dynamic, and spontaneous nature of their needs [37], which current development environments directed towards them cannot address yet.

***Service composition in the Telecom domain.*** In Telecom, service composition is known as feature application composition [19] that is as a process inherent to the establishment of any communication. For instance, a basic call service between two subscribers involves multiple features, such as

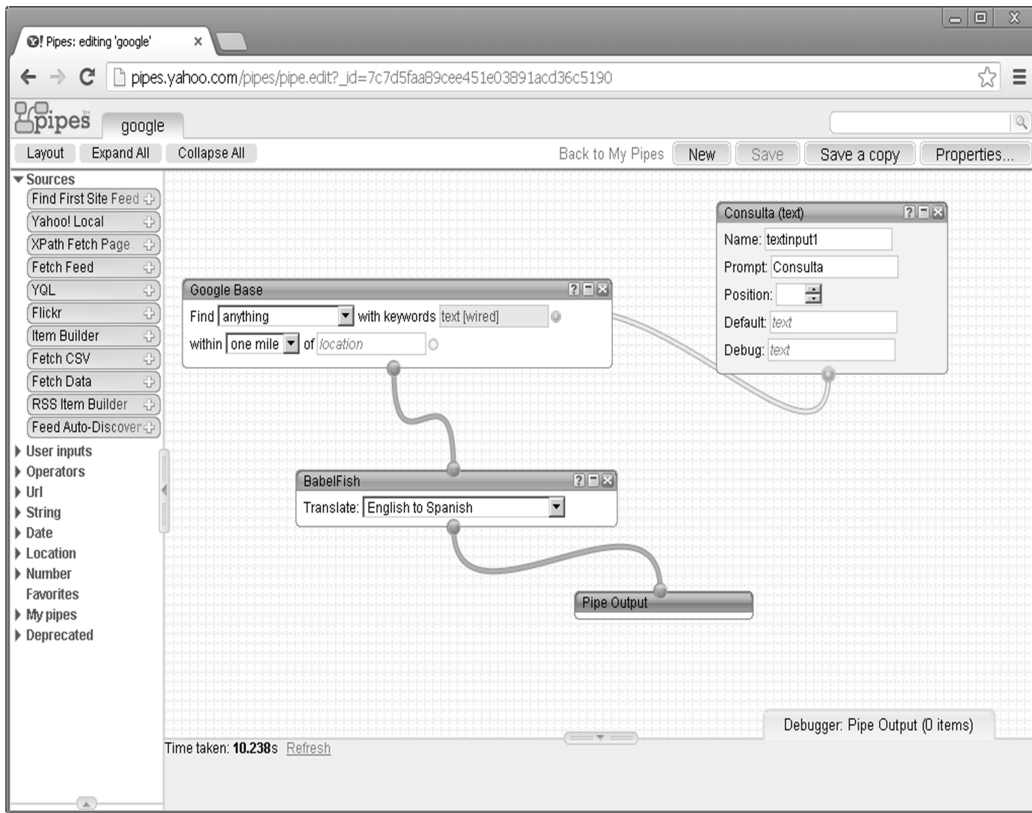


Figure 5: Service composition using yahoo pipes

voice mail, call forwarding, and call waiting. All these features are invoked in a sequence even if the user is never aware of the feature composition.

Similarly to the individual service description, in the Telecom domain, there is a shortcoming related to the little consensus about standard descriptions for service composition. In recent years, the TelcoML [40] and Diagen [100] were proposed for coping with such shortcoming. TelcoML has been proposed by the Object Management Group (OMG) [40], offering a UML Profile for designing advanced and integrated telecom services. In turn, DiaGen is a framework based on the implementation of a declarative language over Java that provides service discovery and high-level communication mechanisms. Such a framework also generates the class skeletons of composite services.



On the other hand, some standards, such as the IP Multimedia Subsystem (IMS) [62], Parlay X [101], and OneAPI [61], aim to carry out composition of individual Telecom services into new more complex services, leveraging the reuse and standardization of subjacent technologies [19]. In general terms: (i) IMS can be seen as an architectural framework for delivering IP multimedia services by permitting the integration of services based among others on SOAP, REST, and SIP, (ii) Parlay X introduces the concept of Telco Web services to allow the creation of composite services; and (iii) ONEAPI exposes network facilities as services, aiming to facilitate the combination of these facilities with services from the Web domain.

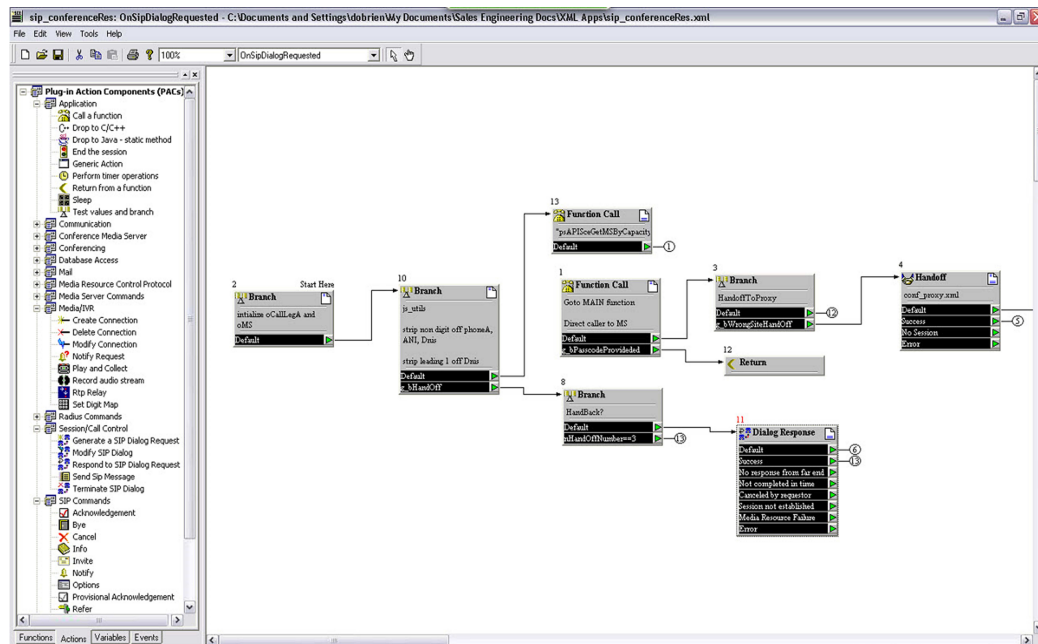


Figure 6: The Radisys SCE [43]

Like service composition in the Web, the service composition in the Telecom domain may also be carried out by using SCEs [4] and MDEs [5] [6]. These environments allow the user to combine services functionalities using drag-and-drop and wire tools that hide the technical details of technologies like IMS, Parlay X, and ONEAPI. Figure 6 depicts a screenshot of Radisys [43] that is an example of a typical SCE. In the user interface of Radisys, the functionalities to be composed graphically in the central canvas are located

at the left pane.

### 3.5. *What is convergent service composition?*

The term “convergent” has been used widely in telecom and computer networks with different meanings: to describe the fusion of fixed and mobile services [2], or as a synonym of the NGN that represents the integration of the Internet and the Public Switched Telephone Network (PSTN) [3]. As previously stated, we use the term “convergent” as efficient coexistence of services available on the Telecom and Web domains to provide innovative and more complex services to the user.

The rise of composition of services appears in the midst of a growing expectation of traffic volume by telecom operators. This is due to the increasing offer of added-value services to customers in addition to traditional voice and data communication [102]. Novel added-value services combine features from the Telecom with technologies of the Web; this combination is known as “convergent composition” [23] or “unified composition” [19].

To illustrate what the process of convergent composition is, we present an example in the Early Warning Management domain in which an environmental manager must make decisions regarding the environmental alarms of a region. He/she can request information from sensor networks deployed at several spots and can also use services available on a telecommunication network and the Web to process basic data and send information to both farmers and actuators. Figure 7 presents a sketch of an environmental management system.

As the environmental manager may come from fields like biology or agriculture, his/her technological background may be low. Also, electronic devices in the area may be scarce and not reliable or obsolete. Furthermore, the data network coverage may be deficient too. Thus, the preferred way to enter information to the system is by voice and in natural language. This way, users do not have to know how the system internally works and can make requests from regular mobile devices or landline phones. Commonly, the user expresses his/her request informally. Here are two examples of requests in natural language: (i) “I need to compute the hydrological balance of zone 1”; (ii) “If the river flow of zone 2 is greater than 15% of the safety limit on average, send a warning to every farmer within a radius of 2 miles from the river and create an action map including emergency and rescue groups”.

In the first request, a service composition could be: gathering information from the sensors in zone 1, using hydrological services from the Internet

to process the sensed data, obtaining the hydrological balance map from Google maps, composing the final image, and sending it to the user by the Multimedia Message Service (MMS).

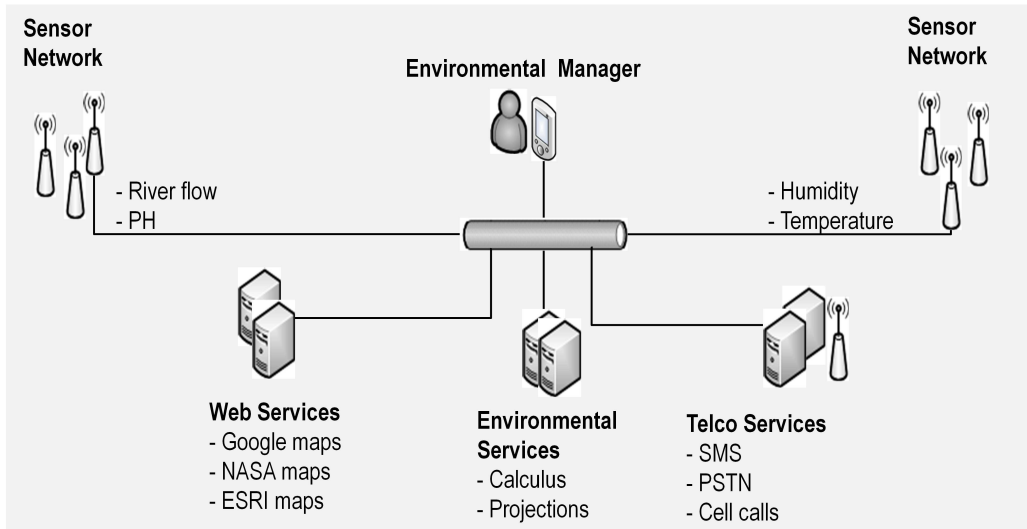


Figure 7: Architecture for management of environmental early warnings

In the second request, a service composition could be: sensor data is evaluated and, if necessary, an emergency map is created. The map is created drawing a radius of 2 miles around each sensor reporting danger. To do so the system may search for geographical services available in the Web. Finally, the system informs about the alarm to the farmers inside the emergency area. The most convenient way to send the information is selected: SMS, Cell Phone call, fixed telephone call, or voice message.

In both request examples, services from the Web and Telecom are used for attending user request. Additionally, the context of the user for both farmers and managers must be considered in order to get a service composition that fulfills the needs of the user adequately. For instance, if the user is currently driving then a service could automatically provide information through voice, such as SIRI [16] and Vlingo [103], instead of text. Another way of taking into account the context of the user is allowing services that can be triggered under specific conditions; for example, it might be possible to determine whether a rescue vehicle is close to an affected area, in which case an emergency notification would be triggered [102].

## 4. Convergent Service Composition

In a general sense, the process for carrying out convergent service composition can be decomposed in several relatively independent steps, beginning with the user request and ending with the execution of the composite service. The whole process (see Figure 8) can thus be divided in four phases: creation, generation, adaptation and execution.

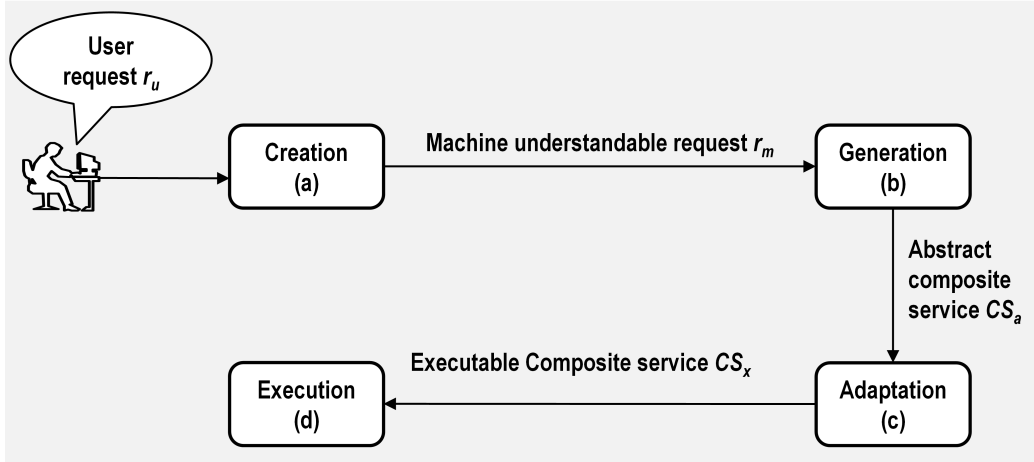


Figure 8: Phases for automated convergent composition

An important consideration to be noted is that the automation of each phase is independent of the automation of the rest of the other phases. This means that the automation of the whole service composition can be described as the sum of the automation of its parts. Next, each phase is described as well as the automation level expected.

### 4.1. Creation phase

In this phase, the user (advanced or novice) specifies the user request (*i.e.*,  $r_u$ ) in a natural way in order to be solved by the system (see Figure 8, phase a). The user specifies  $r_u$  through a device or computer that hides any technical (platform or language) detail and allows him to clearly state his necessities with his own words. Such  $r_u$  needs to be translated to a machine understandable representation (*i.e.*,  $r_m$ ). An example of the input and output of the creation phase may be:  $r_u$  : “I want to receive the route to the closest ATM in my cell phone” and  $r_m$  :  $get(calMap(myPosition, ATM))$ .

In the creation phase, the automation is associated with the capability of the system to extract  $r_m$  from  $r_u$  without the intervention of the user. This type of automation is mainly required when the expertise of the user is not advanced, so he/she cannot specify the request directly in the machine understandable format.

#### 4.2. Generation phase

The next step in the process for convergent service composition consists on translating the machine understandable request  $r_m$  into an abstract composite service (*i.e.*,  $CS_a$ ).  $CS_a$  represents a set of atomic activities that need to be performed in order to satisfy  $r_u$  (see Figure 8, phase b). This set of atomic services can be described as a business process or an algorithm.

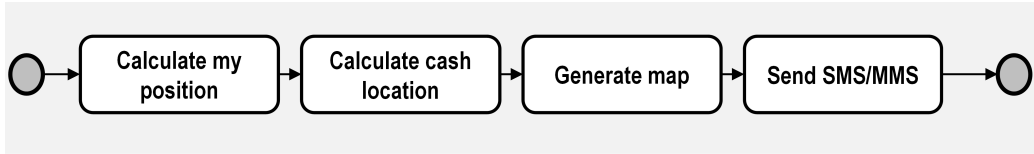


Figure 9: Graphical representation of the composite service

Figure 9 depicts a possible graphical representation for a  $CS_a$  that follows the following actions sequence: (i)  $myPos = calLocation(here)$  - the location of the user is obtained and stored in a variable, (ii)  $dest = calLocation(ATM)$  - the destination location is found, (iii)  $map = calMap(myPos, dest)$  - the drawing of the map from the location of the user to the closest ATM is achieved; and (iv) finally,  $sendMMS(map, phoneNumber)$  - the drawn map is sent to the mobile phone by MMS. This is also an example of service composition in which Telecom services (*e.g.*,  $calLocation$ ,  $sendMMS$ ) and Web functionalities (*e.g.*,  $calMap$ ) are needed.

The process of automation of the generation phase may be carried out in two ways: from scratch or using predefined templates. The former implies that a basic composite service does not exist at all and therefore must be created. The second type of generation is based on templates, which means that a basic template of the composite service describing the flow of activities is already defined and the user request triggers an adaptation of this flow, for example by selecting the implementation services in each occasion.

In both ways of automation, the generation of convergent service composition requires the selection of discovery of services for each task of the

composite service. This process consists on the identification of a service that can potentially be used in the context of a new convergent service. This includes the comparison of expected services with the existing ones, which is called matching. Since individual services are commonly isolated, their format and semantics are also heterogeneous even if their purpose may be similar, so it is necessary to employ matching algorithms.

Researchers have developed diverse matching approaches [104]: (i) exact syntactic equivalence in which a word comparison of names of services and attributes is performed, looking for an exact matching, (ii) approximate match that offers functions to consider acceptable differences between words; and (iii) semantic match that uses semantic distance, which is associated with the conceptual similarity using concepts networks named ontologies.

#### 4.3. Adaptation phase

In this phase, the abstract service composition ( $CS_a$ ) is translated to an executable plan, called  $CS_x$  (see Figure 8, phase c). This translation is conducted to enable the execution of convergent services in technological platforms. Listing 3 depicts, using the Java Language, the  $CS_x$  of the above exemplified  $CS_a$ .

Listing 3: Example of Java executable composite service

```
public class CompositeService{
    public void Request() {
        //calculate coordinates of cash machine and my position
        here = Locator.getActualPosition();
        cash_machine = Locator.getPosition(Gzzter.get(here ,
            cash_machine));
        Location my_position = Locator.getLocation(here);
        Location destination = Locator.getLocation(cash_machine);
        //Generate the map in an image
        MapImage map= MapService.render(my_position , destination);
        //Send the message to the user
        TelcoApi.sendSMS(user.getPhoneNumber , map, "result");
    }
}
```

The automation of the adaptation phase consists on creating the executable or interpretable code in a programming language like Java/Python or in an declarative language like BPEL/Bite without requiring the intervention of a developer.

#### 4.4. Execution and reconfiguration phases

The execution phase (see Figure 8, phase d) is responsible for managing the lifecycle of  $CS_x$ . Therefore, in this phase, the convergent service is

deployed into an execution environment (*e.g.*, a server) to be subsequently executed. The execution phase is inherently automatic: once the user deploys the service, it runs in the execution environment without human intervention.

The reconfiguration phase is required, for instance, when a malfunction (*e.g.*, server failure) or an error (*e.g.*, a Web service becomes offline) happens during the execution of  $CS_x$ . The usual procedure for reconfiguring a convergent service consists in notifying the developer/user, who must go to the corresponding SCE/MDE for re-examining the whole composition process. Complete automation of this phase requires the composite service to be monitored and reconfigured automatically in case of failure [105].

## 5. Service Creation

As afore defined, Service creation is the process of translating the user request to a language or descriptor understandable by the computer. For instance, the translation may be done from the request expressed in natural language to BPEL/Bite language or to executable code. This phase depends largely on the level of expertise and the device used by the user. If the user is an expert and uses a desktop personal computer, a graphical interface for creating workflows is very suitable. On the contrary, the same interface is not suitable for a non-expert user, or for another user who may be driving when needs to compose a service for satisfying his/her needs.

In terms of academic research, a clear separation between programmers/developers (*a.k.a.*, professional users) and end-user needs to be done according to the conclusion of European SPICE [31] and OPUCE [27] projects. In this way, in the following subsections, the main approaches for service creation will be exposed considering focus on professional users (*i.e.*, SCE-based approaches) and end-users (*i.e.*, approaches using natural language processing - NLP - and mashup-based approaches). Most of them comes from academic initiatives.

### 5.1. SCE-based approaches

SCEs ease the service creation by offering mechanisms for integrating services and functionalities graphically. This idea was initially suggested by researchers that proposed the use of Integrated Development Environments (IDE) able to visually integrate atomic components representing Telecom and Web 2.0 services [30].

The SCE proposed by the European OPUCE project offers intuitive tools for graphical creation of composite services [30] [26] [27]. This SCE allows the integration of Web and Telecom services taking into account some user profile criteria. Furthermore, the SCE of OPUCE includes a semantic browser and a translator to BPEL.

HSCEE is other research work that carries out SCE [28]. This work apart of providing a low latency execution environment for convergent services, offers also a template-based service creation platform. An important shortcoming of HSCEE is that it is based on BPEL templates where most of the design tasks must be done manually, which makes it ill-suited for automatic service composition.

Another SCE-based approach was proposed in the field of the Artificial Intelligence (AI) [11]. Such approach aims at automating the Web Services composition under semantic awareness by AI planning techniques and a graphical environment in which even non-expert users (*i.e.*, end-users) can create composite services. One of the most striking characteristics of this approach is that the expertise required by non-experts for carrying out composition activities is essentially similar to discover and invoke an atomic Web Service.

In the TeamCom project arose other SCE-based approach that describes the development of added-value services through a tool based on BPEL [53]. This tool additionally allows to translate to Java and deploy on a Telecom execution server the developed added-value services.

Finally, regarding SCE-based approaches, it is important to highlight that in recent years, the concept of Service Delivery Frameworks has also been proposed to provide service creation in convergent scenarios [25]. We consider these frameworks just like other form to refer a solutions using SCEs.

## 5.2. Mashup-based approaches

As an alternative to SCEs, an important trend of research has emerged around the mashup technology that allows creating solutions, named mashups. These mashups are composite Web applications centered in end-users and created by combining different resources available along the Web [98] [99]. End-user centric means that mashups may be developed by users who usually do not have advanced programming skills [37], leveraging MDEs that tend to be user friendly and hide technical details [34]. Furthermore, mashups encourage the cooperation among end-users and the reuse of existing Web applications [106] [107].



The proponents of the use of mashups for creation of convergent services promote the generation of mechanisms for service composition from both Web and Telecom using friendly interfaces. This is done following the Web mashups philosophy based on the fact that a large number of services are accessible over the Web through open interfaces.

An example of mashup-based approach for service composition is a framework that introduces the concept of mobile mashups [33]. This framework proposes a device-independent architecture applicable to both the server and the device side that allows integrating device capabilities (*e.g.*, camera and Bluetooth) with Telecom features and Web accessible services.

MARIO [51] is a tool that implements a “pursuit of desires” pattern, in which users set their preferences by selecting tags describing services to which they want to subscribe. The internal composer of MARIO makes the correspondence between the tags and the existing services. Such correspondence is in turn refined by the user until he/she gets the final composition.

In the domain of added-value convergent services, a reference architecture was proposed to facilitate the provisioning of telco mashups for end-users [20]. A telco mashup is a composite service that combines functionalities from telecommunication networks (*e.g.*, streaming, quality of service, and billing) and device capabilities (*e.g.*, geographic location and health information) with services (*e.g.*, instant messaging and IP telephony) accessible via Web-based protocols. In the literature, we did not find a prototype of a mashup system or a telco mashup that implements this reference architecture.

In the network management domain, a mashup-based approach was proposed to deal with the security problem of botnets, in an more flexible, extensible, and usable way [74]. This approach was carried out in a mashup application. The mashup built (called botnet mashup) integrates dynamically botnet information collected from existing mitigation tools, such as sandboxes and antiviruses, with geographic location retrieved from online mapping and geolocation APIs.

A mashup system was also introduced to evaluate qualitatively the feasibility of using Web 2.0 technologies on network management [108]. Using the proposed system prototype, a mashup (called BGP mashup) was built in order to monitor the traffic of the Border Gateway Protocol (BGP) among two autonomous systems by integrating traffic router information. This integrated information was collected by using the Simple Network Management Protocol (SNMP) and presented by combining images retrieved from a generator of traffic graphs and maps generated from a mapping service.

As a continuation of the two works above cited, a generic architecture was proposed to support the composing of network management applications [5]. This architecture was carried out in a MDE that allows network administrators to design their own management applications through the composition of network management and services available in the Web. The qualitative evaluation of the proposed architecture was conducted by analyzing the development of the BGP mashup and the botnet mashup previously referred.

As an evolution on using mashups for network management arose the concept of mashments [13] [14]. Mashments are solutions based on the Situation Management (SM) discipline and the mashup technology that aim to facilitate the work of network administrators when facing unexpected, heterogeneous, and dynamic situations. To achieve such aim, mashments combine diverse types of resources (data, application logic, and user interfaces) from multiple providers (from the Web and the network management) and automate the investigative and control aspects of SM.

An important shortcoming of the mashup-based approaches above cited is that the selection of atomic services used for creating mashups is performed manually by using mainly searches based on keywords. In this sense and with the goal of overcoming such shortcoming, in the literature of service composition arose diverse and relevant semantic mashup approaches. For instance, SensorMashup is a semantic-based platform aimed to facilitate the creation of mashups by following the linked open data principle [76] that enables the dynamic discovery, integration, and querying of heterogeneous data sources. This platform provides a visual composer that allows end-users to easily publish sensors as data sources identified by URIs and to rapidly combine/link such sensors for building up mashups.

In service composition, other groundbreaking approach was proposed for improving the process of developing mashups by adding semantic service discovery capabilities [77]. In a general way, these capabilities were added by combining semantic Web Services discovery frameworks, Universal Description Discovery and Integration (UDDI) repositories, and existing mashup tools.

In the Cloud Computing domain, a semantic-based approach was proposed to make the engineering process of creating mashups more agile and flexible [78]. In particular, this approach uses the Cloud Computing technology to increase the number of public REST APIs and further introduces the Semantic Annotation for Web Application Description Language (SAWADL) to carry out the semantization of such APIs. SAWADL is a WADL extension

that enables the annotation, combination, and deployment of REST APIs, aiming to facilitate the development of mashups in a cloud environment.

Instead of relying on the expertise of the user as the above mentioned mashup-based approaches, other ones focus on capturing the user request, defined in a specific/proprietary description language of composition, and translating it into composite services. In the approaches using these specific or proprietary languages, the composite service description made by the user is translated into a process definition language (*e.g.*, BPEL and/or BPEL for REST) before being executed. For instance, in the SOA4All project [97], non-expert users interact with a graphical process editor called SOA4All Studio to create new composite services. This editor directly interacts with a reasoning engine that creates a composed service by inferring in a semantic way the most suitable connection among parameters of the services exposed on it.

### 5.3. NLP-based approaches

Recent approaches have also explored the application of NLP techniques to automate the composition of services. These approaches have been used mostly in intelligent home environments [35] [36] and offer mechanisms to map words to basic functionalities of the system. The goal of a NLP-based system is to extract the workflow of the composite service from the user request expressed in natural language.

In the domain of environmental early warnings, a NLP-based system named AUTO was proposed for composite service creation [38] [39]. In AUTO, the user defines his/her demand and context information using natural language. This demand is handled by a request processing module that is responsible for transforming natural language and context information into a plan of composite service. This plan combines services from the Web and Telecom domains.

In the SPICE project, a specialized language called SPATEL [31] was proposed to describe service compositions formed by RESTful services (*i.e.*, RESTful is a Java implementation of the REST architectural style) and Telecom features. These descriptions are interpreted by a translator module [32] that is responsible for generating code for specific execution technologies. SPATEL and the translator module are the core of the generation and execution of composite services offered by the SPICE platform.

#### 5.4. Comparison

After surveying the previous approaches, it can be stated that the works in the service creation phase can be classified in three groups: SCE-based, mashup-based, and NLP-based. Each one of them has different particular elements that make them suitable for distinct environments. Table 3 summarizes the differences between the three approaches.

Table 3: Comparison of approaches for service creation

Approach	Access	Automation	Expertise required
Mashup-based	Any Web-enabled device	Low	Medium
SCE-based	A personal computer with the SCE installed	None	High
NLP-based	Any personal computer or phone	High	Low

SCE-based approaches offer an incomparable technical detail for convergent service composition. When employing SCEs, the limit on the intricacy of the service composition depends only on the number of available services and the knowledge of the user. However, this obviously requires the user to be an expert in the specific application domain. Furthermore, because the complexity and nature of SCEs, usually they must be installed in a desktop personal computer, which greatly restricts the mobility of the user.

SCE allows to create convergent services with a remarkable level of detail, these interfaces allow to customize different aspects of the services and the subjacent protocols. Unlike Web services, the protocol in telecom environments is very important for the provision of the service and may directly benefit the quality of services. Therefore SCE might be a very good alternative for technical and experts when defining the template of convergent services that could be used later in further phases of automated composition.

Mashup-based approaches, in turn, offer more alternatives by enabling service creation from Web graphical interfaces that do not need to be installed in any desktop personal computer. Additionally, the usability as well as the range of audience for these interfaces is bigger than for SCEs. The main

disadvantage over SCEs is the loss of expressive power when creating complex compositions, which may be too simple for the needs of some users.

Some approaches create Mashups automatically from natural language request and allow to refine some details at a later time. On the other hand, Mashups are generally open and may be shared and reused which may lead to a collaborative creation of convergent services.

The approaches based on SCE and mashups allow creating composite services that may be defined using different languages, such as BPEL, Bite, LPML, and BPEL for REST. In these approaches, the core of the service composition task is completely carried out in a manual way. In this way, the novelty of the application of NLP-based approaches is that, in some scenarios, the user just requests his/her desire or goal in natural language and does not define the workflow of the composite service. This workflow is obtained automatically from the user request by the NLP system. Furthermore, by using voice as the primary input medium, a NLP system may be used with any kind of calling device.

Existing research results have shown that creation phase have been automated significantly when natural language is used. However, several important issues are open:

i) To map user request and context to a service is a big challenge, especially when the service does not have a standard interface description like in telecommunications domain, many semantic approaches translate requests to formal representations using natural language processing techniques and ontologies. These approaches have good performance in specific domains [109]. The main idea is extract concepts from the user request and match them with service concepts [110].

ii) There exist some differences between finding services on Telecom and on the Web. The main challenge of finding a service in telecom environment is the lack of standard descriptions for the functional properties of services. Some approaches for automated creation use traditional Web representations for Telecom environments such as WSDL, however the trend of moving telecom facilities to the Web such as Skype or Twilio allow to glimpse the use of semantic representations of Services.

## 6. Service Generation

As above described, the service generation is related to the creation/composition of a chain or a linked set of services that can be represented as a busi-

ness process encoded, for instance, in BPEL or BPEL for REST. There exists a broad range of research approaches around the topic of automated/semi-automated service composition in the Web domain, most of which is applicable to convergent composition. Such approaches can be divided into two categories: workflow-based approaches and approaches that employ AI techniques.

### 6.1. Workflow-based generation

The approaches on service generation using workflow-based techniques assume that a composite process can be perceived as a workflow. The workflow of a composite service includes a set of atomic services along with a data and control flow [104]. Workflow-based approaches can be subdivided into two groups: static, template, and dynamic composition.

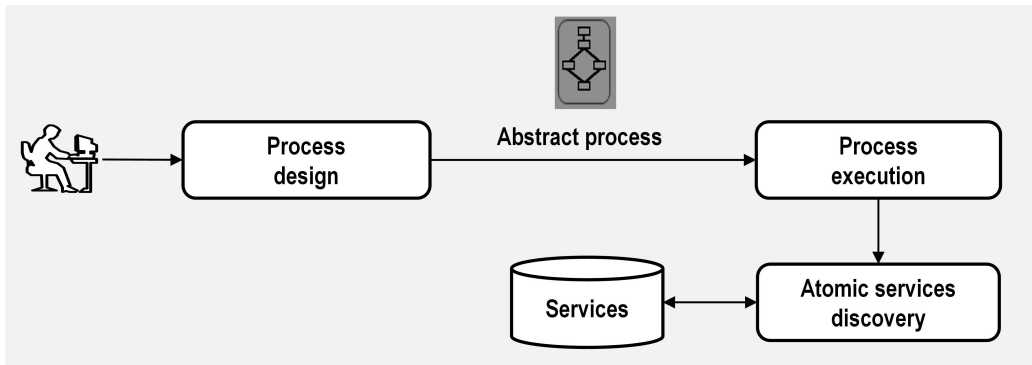


Figure 10: Static workflow-based composition

The static composition (see Figure 10) means that user builds an abstract process model or template at design time, and then at execution time atomic services are selected automatically from a set of candidate services. Graphically the composite service is depicted as a directed acyclic graph, with control and dataflow [111]. This graph defines the order of execution of the nodes of the process, in which the nodes are automatically bound with particular services. This automation focuses mainly on the automation of the instantiation process [73].

The use of template-based composition is located at the middle of static and dynamic composition. Using templates, the user retains control over the

process and automatic techniques can be employed to customize and implement the process, achieving very good performance and quality [112] [113]. The template-based composition depends on whether standard procedures are available for the particular domain of application, such as in medical protocols [111]. The templates can be parameterized with respect to some variables to allow customization based on user needs and preferences [92].

A concept that uses templates is the Hierarchical Task Networks (HTN) [64]. HTN planning is especially promising for service composition because its concept of task decomposition is very similar to the concept of composite process decomposition. In HTN planning, a task represents the activities to perform. A task may be either primitive or compound (which in our case represents an individual and composite service respectively). A compound task is one that needs to be decomposed into smaller tasks using a method. Methods indicate how to decompose a compound task into a partially ordered set of subtasks, each of which can be again compound or primitive. The lowest level task that cannot be decomposed further is called a planning operator. In service composition the highest level task or goal is extracted from the user request and the lowest level planning operator represents a primitive service directly associated with implementation services.

The dynamic composition proposes the generation of the service abstract model automatically. In this way, individual services are selected in an automatic way (see Figure 11). Some relevant approaches on dynamic composition are, first, a rule-based solution for semi-automated composition that offers the user the ability to guide the composition process [71]. In this solution, the user defines the flow and the rules that allow automatically building the composition. Second, a mechanism that works by synthesizing its workflow based on the semantics of the components and the context of users [72]. This mechanism composes applications requested in natural language considering in addition to the context the preferences of the user. These preferences are extracted from user-specified rules and also via learning.

Important shortcomings of the workflow-based composition is that it usually requires extensive domain knowledge and at least some degree of manual implementation by developers, making it difficult, time consuming, and error-prone [111]. In order to overcome such shortcomings arose the service composition based on AI.

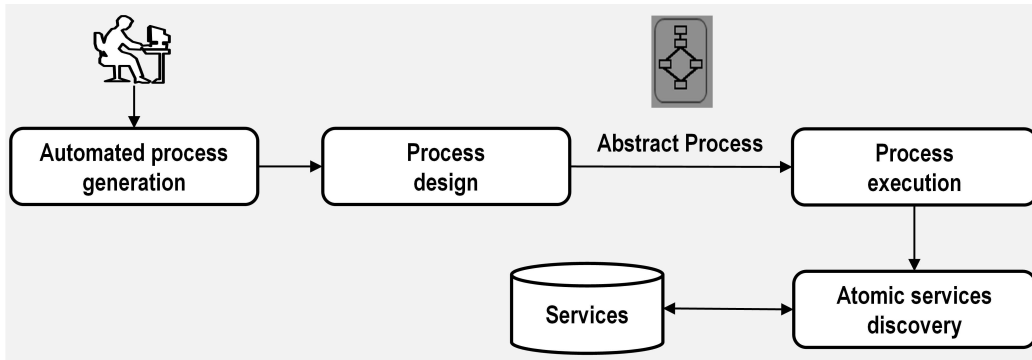


Figure 11: Dynamic workflow-based composition

### 6.2. AI-based generation

One important group of approaches is focused on semantics. Semantic composition has been used widely for services composition. The main idea is to add semantic information to methods and I/O parameters in order to judge the matching degree between services by calculating semantic similarity of concepts in ontology tree [114]. Furthermore, some authors extract concepts from user goal description and matched them with ontology concepts so as to guide the services composition [115]. For example, Slaimi et al. [116] present an approach called OVWSC (Ontology based Web Service Composition) that includes two phases: first the domain service ontology is filtered for reducing the services research's space. Secondly, an abstract workflow representing the composite service is generated.

In spite of the high precision that might be achieved with semantic techniques for services composition, some limitations exist: i) this process is time-consuming due to the inference required during search. ii) The lack of unified ontologies in different domains, prevents its spread. ii) It requires users to have enough domain knowledge to perform semantic annotation and searching, which is not feasible in real application [117]. For improving performances of semantic search, some approaches present structures such as quick Extended Quick Service Query List (EQSQL)[117]; these structures associate service and request concepts at design time in order to decrease the search time. In [118] the author present a platform for context-aware service composition based on user-defined rules. The platform interprets ontology descriptions of services and creates user-defined rules based on the ontology



description of services. Zhang et al. [119] present a review of the recent research methods on Semantic Web service composition.

Nowadays, many approaches for service composition integrate diverse AI techniques: such as rule-engines and logic reasoning, for example. Tang et al. [120] present a method for automatic Web service composition based on logical inference of Horn clauses and Petri nets. The problem of service composition is transformed into the logical inference problem of Horn clauses by exploring the dependency relations among services. Besides, the authors, present an algorithm for determining if the composite service exists in linear time. Li et al. [121] present a novel composition mechanism by extending OWL-S model. This mechanism consider user preferences and combines the strong knowledge representation and reasoning capabilities of Description logic with the modeling capability of action state transformations in AI planning, and finally testifies its availability in DL. Finally, Priya et al. [122] present an approach for QoS based web service selection. The authors address the composition of Web services using a Fuzzy rule based system. This composition allows imprecise information to be considered and help to model highly complex problems that have multi dimensional data.

Different techniques from AI have been applied in automated composition, the most successful up to date being automated planning [121] [10]. Figure 12 depicts the general automated service composition in which the process model (*i.e.*, service model) is created automatically.

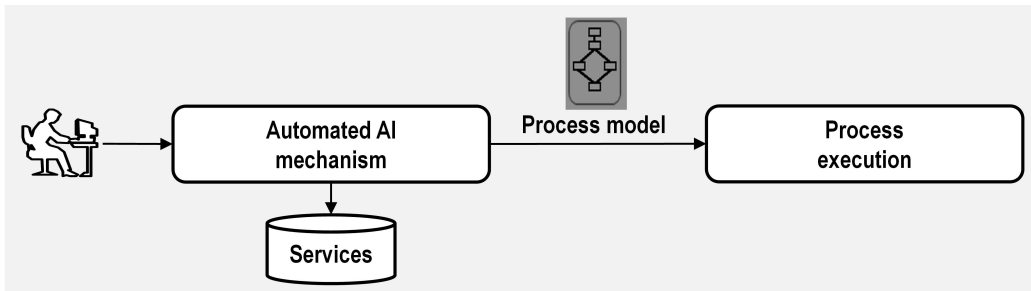


Figure 12: Automated service composition

Automated planning can be defined as the task of coming up automatically with a sequence of actions that achieve a goal [123]. In planning there are three types of elements: states, goals, and actions. Given an initial state, the goals and the set of possible actions, a planning system must find a se-

quence of actions (called plan) that leads to some state in which the goals are true.

An action in automated planning is specified in terms of preconditions and effects. The preconditions are the facts that must hold true before the action can be executed, while the effects depict how the state changes when the action is executed. In this way, a service can be seen as an action and the composite service as a plan [73]. This kind of composition is automatic and does not require knowledge about some predefined workflow.

Several approaches have emerged from AI strengthened by the fact that OWL-S [90], the language of choice for the semantic description of Web services based on SOAP, maintains a similar structure to the description languages used in Automated Planning [10]. In this sense, for instance, an adaptation and extension of Golog was proposed to allow automated construction of composite services [66]. Similarly, an approach that combine HTN and automated planning was introduced for generating service composition [69]. This approach allows the implementation of preferences during service composition.

Other relevant AI-based approaches are, first, a proposal that includes a semantic description of the composition process and of the discovery of the atomic services [11]. Second, a framework for composition based on automated planning that uses an algorithm for service discovery for consider QoS requirements as well as transactional constraints during the service generation process [65]. Third, a solution that propose to employ planning in NGN by modelling Telecom services using situation calculus and reducing the composition problem to a planning task [68].

### 6.3. Comparison

For service generation, there are a lot of manual (*i.e.*, workflow-based) and automatic (*i.e.*, AI-based) approaches. The service generation based on workflow requires that the users to know the domain in order to define the workflow. Thus, in a broad sense, a workflow-based approach may become cumbersome, error-prone, and infeasible for large or even medium scale service composition.

Contrary to manual composition, AI-based automatic generation tends to scale better as the size of the composition increases. Most of the work so far has been done on automated planning; a small description of the basic planning algorithms and their application to the service composition

problems can be found in the survey “Current State and Challenges of Automatic Planning in Web Service Composition” [124]. In general terms, the automatic service composition problem can be reduced in some cases to the boolean satisfiability problem [125], a well-known NP-complete problem. In this sense, algorithms that carry out exhaustive search may work for small scale compositions, but for larger problems approximate algorithms that find sub-optimal solutions are more desirable [104].

To finish this comparison, it is also important to highlight that for service generation, approaches based on AI and workflow can be combined (*i.e.*, these approaches can be complementary). For instance, a framework was introduced for intelligently navigating service composition [70]. This framework consists of two primary processing modules: the Planning Module and the Constraint Satisfaction Problems (CSP) Solving Module. Taking advantage of the functional attributes of the services, the Planning Module aims to produce referred composite plans that function as general patterns. The CSP Solving Module focuses on selecting qualified services to implement the composite plans, based on their non-functional attributes evaluation.

## 7. Service Adaptation and Execution

In this section, at starting, we present approaches about the service adaptation phase. Later, we present research addressing the service execution and reconfiguration phases.

### 7.1. Approaches for service adaptation

The adaptation phase translates the composite service into an executable component runnable on a Telecom application. In this sense, the automation of the adaptation phase consists of translating the composite service into an executable component without human intervention.

The most remarkable approach for automated service adaptation is presented within the scope of the TeamCom project [50] [49] [52] (see Figure 13), which deals with cases in which the developer describes the business process using BPEL but the resulting BPEL process is not necessarily executable on a BPEL-engine. The TeamCom platform translates the resulting BPEL process in XML; next, these files are in turn transformed into Java code that is deployable on a JAIN Service Logic Execution Environment (JSLEE) server [53].

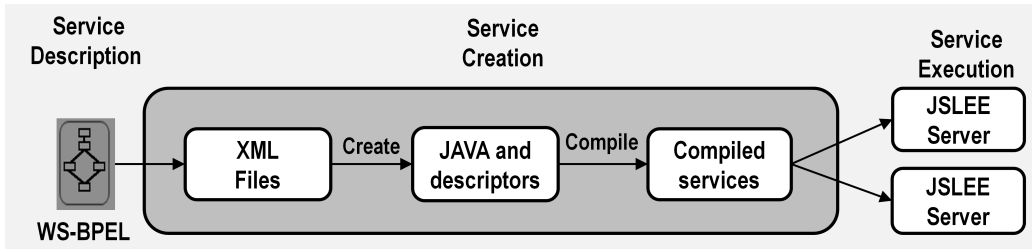


Figure 13: Graphical representation of the composite service [50]

### 7.2. Industrial approaches for service execution

Regarding composition of services in the Web, the predominant vision for representing composite services is BPEL and its extensions like BPEL for REST (see Figure 14). Notwithstanding such predominance for representing business process, BPEL/BPEL4REST have not been developed for control of specific communication services [50], particularly in real time like usually Telecom services require. Consequently, BPEL and its associated execution servers are not appropriate in the sense that they cannot provide the availability requirements necessary for convergent service execution [48]. In this way, we consider that the best option for executing convergent services must be found in the range of approaches coming from the Telecom domain. The following paragraphs present a summary of the Telecom technologies that provide alternatives for service composition execution.

From the Telecom industrial perspective, IMS [62] is the most important global architecture for service execution. IMS is an NGN architectural framework for delivering multimedia services based on IP and SIP. In its origins IMS represented an approach to delivering “Internet services” over data services for 2G and 3G mobile networks. The purpose of IMS was later updated for supporting also Wireless Local Area Networks (WLAN), mobile protocols, and fixed lines. An important characteristic of IMS is that it uses standard technologies for its communications in order to provide convergence of services. Despite the potential that IMS has, the migration to IMS-based platforms has been overall slow due to the high cost of updating the equipment and software for Telecom management necessary to adopt IMS.

Figure 15 presents the layers of IMS. The Access layer manages the connection of users with IMS terminals, such as phones, personal digital assistants (PDAs), and computers, through different protocols. The only require-

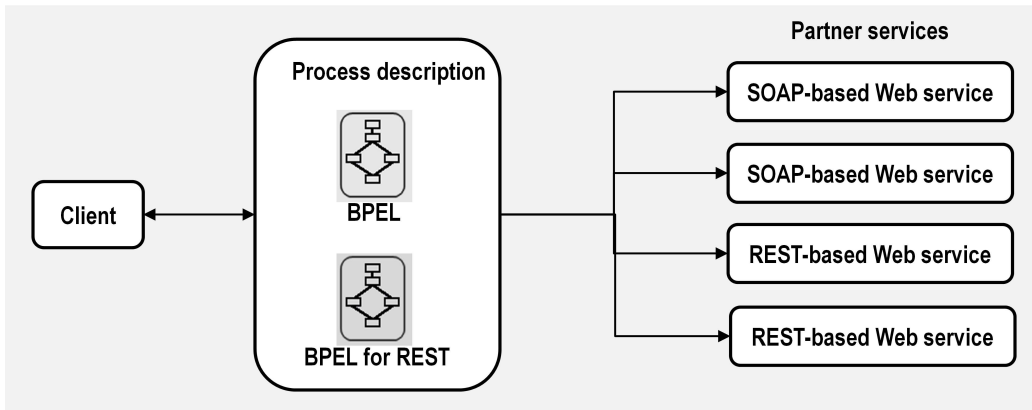


Figure 14: Graphical representation of a composite service

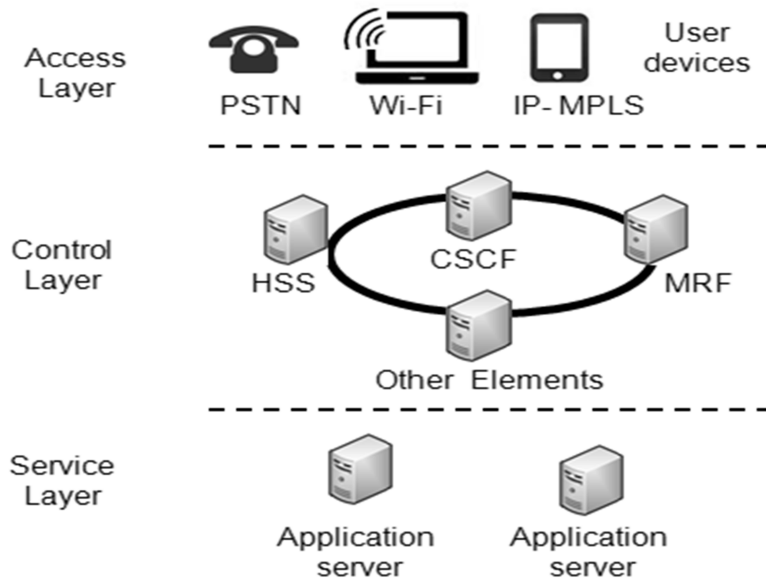


Figure 15: Layers of IMS

ment is that these devices must support IP and SIP. Most of the protocols of mobile Telecom networks (*e.g.*, the General packet radio service - GPRS) and wireless networks (*e.g.*, WLAN) are supported. Other non IMS-compatible phone systems like traditional telephone devices (*i.e.*, PSTN terminals) are supported through gateways.

The Control layer manages the authentication and routing; this layer also provides the capability of associating or composing services from the Services layer. The main element in this layer is called the Call Session Control Function which orchestrates the interaction between the services in the service layer. IMS uses a simple orchestration mechanism to select the services that are needed within a Telecom session. However, this method is not dynamic, so any service that a user might need is placed in the chain of services, regardless of their actual usefulness for a given session.

The Services layer holds all of the actual set of services. This set includes traditional Telecom services, such as call forwarding and voicemail, as well as services accessible via Web-based protocols. The IMS services may be installed in application servers or in media servers. In the Service layer, most of the researchers [50] [49] [52] [126] [127] [128] have focused on the use of JSLEE as the subjacent application server for executing convergent services.

JSLEE (see Figure 16) is a high throughput and low latency event processing application environment. It provides a robust platform for rapid development and deployment of convergent services. The access to network resources and services available in the Web is offered by the Resource Adapters (RAs). Furthermore, the use of RAs allows equipping the JSLEE platform with capabilities such as Authentication, Authorization, and Accounting (AAA) among many others [129].

JSLEE includes a component model to structure the logic of convergent applications where the basic services (from the Telecom and/or the Web domain) are represented as Service Building Blocks (SBBs). JSLEE also enables the composition of SBBs by combining two or more basic services. Recent studies have demonstrated that JAIN SLEE is suitable for provisioning of convergent services in production environments by using commercial solutions, such as the Mobicents JAIN SLEE [131] and the Rhino Application Server [132]. In this sense, for instance, some works [130] [128] has demonstrated that even for complex services, the throughput obtained is comparable with the traffic handled by a large gateway node of an VoIP operator.

The other type of Java application servers that can be used in the Services layer of IMS is based on the SIPServlet specification [63]. SIPServlet contains and manages SIP applications and provides access to SIP mechanisms via a Java API. The SIPServlet API defines a component called Application Router (AR) which is responsible for the selection and composition of ser-

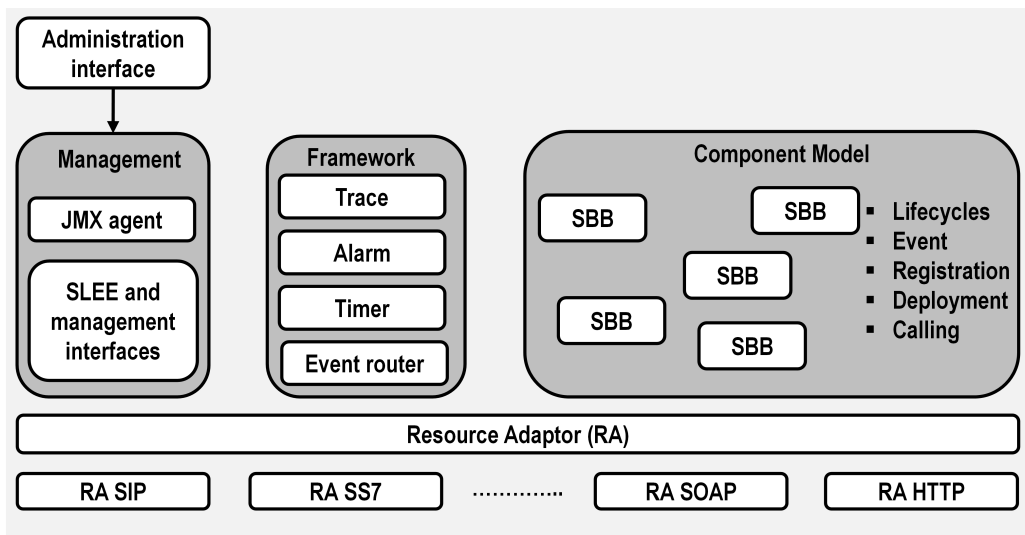


Figure 16: Overview of JSLEE architecture [130]

vices. In this sense, the Ericsson composition platform [59] is a SOA-based solution that allows the composition of convergent applications using services from Web, enterprise and Telecom domains, for both circuit-switched and IP-based services. This solution selects the services to be used and informs IMS about the SIP service to be executed. Other approaches [133] [134] have explored the integration of SIP-based servers (*e.g.*, Sailfin) with Java Enterprise Edition Servers (*e.g.*, GlassFish) to conduct IMS service composition.

Parlay X [101] is an important alternative to provide convergent services in NGN-based networks, permitting the use of Web services in the Telecom domain. Parlay X [60] [135] is a standard that comprises a set of services (from the Web and Telecom domains) offered like APIs that facilitate the composition of more complex functionalities to be provided in Telecom networks. An example of a commercial server supporting Parlay X is the Unified Service Platform ZXUP10 of ZTE Inc [136].

OneAPI [61] appears as an evolution of Parlay X. OneAPI is also a set of supported APIs that exposes network capabilities, such as messaging, payments, and location, on the Internet. Using such guidelines any mobile operator or service provider may implement or use OneAPI. As an example, any messaging service could be used by any other application for sending SMS messages. Regarding the ONEAPI is also important to note that it

already provides simple interfaces for different Telecom and Web protocols, HTTP among them. Supporting HTTP, ONEAPI can execute convergent applications formed by SOAP-based services, Telecom features, and RESTful services. In this vein, some companies provide Telecom services under subscription over the internet using SIP protocol, among them: Hoiio [54], Tropo [55], Twilio [58], CallFire [56], and Skype [57]. Each one of these companies provides their own languages to describe the offered functionalities.

### 7.3. Academic approaches for service execution

Recently, some academic alternatives have reaffirmed the preponderance of JSLEE [137]. In this sense, a service platform was proposed in order to fulfilling the Telecom requirements (*i.e.*, low latency, real time, and fine-grained events) when convergent services are requested. This platform is event-based architecture, provides asynchronous iteration, and supports JSLEE [45].

Other existing approaches [48] [135] [29] [128] mix JSLEE with Enterprise Service Bus (ESB) to support convergence service composition over heterogeneous networks. ESB (see Figure 17) is an architectural model that provides a common connectivity layer between services of the Web. Services supported by ESBs are not restricted by a protocol like HTTP nor have to be described by a specific descriptor such as WSDL. In the ESB world, a service is a functional component that is used as a meta-data descriptor understandable by computers. Such descriptor is published so it can be used without needing the implementation of the service. All clients communicate with the ESB in the same way; in turn the ESB is on charge of the translation of messages (format and structure) between client and provider services [138].

On the other hand, some approaches [139] [140] propose the integration of SIP and BPEL for the execution of convergent services. These approaches present, for instance, the abstract design of a hybrid multimedia conference framework with low latency services, which allows the execution of both primitive and composite conference services.

Finally, HSCEE [28] is an approach that employs ESB, JSLEE, and BPEL to carry out template-based service composition. HSCEE was employed to build a prototype of a typical communication multimedia conferencing system based on Web services. This system can concurrently run multiple instances of conferences in common public networks, being able to support multi-user conferences employing various network terminals from anywhere. Authors also present promising performance results in the presented prototype.



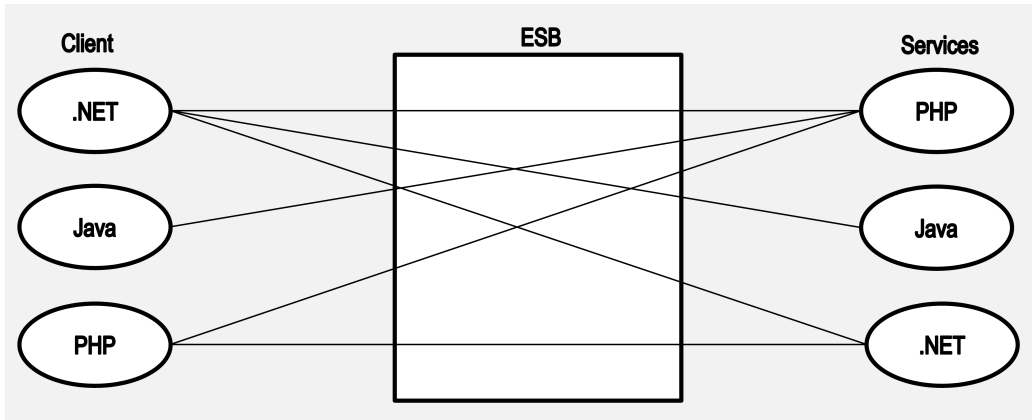


Figure 17: General diagram of ESB functioning

#### 7.4. Comparison

To select a technology for Execution of convergent services, different criteria must be considered. Previous works have focused on the comparison of these technologies [91] and most of them agree that the best selection depends exclusively on of the needs of the user. As aforementioned, IMS is an architectural framework that defines a series of guidelines for future implementation of convergent scenarios. In the Service layer of IMS both SIPServlets and JSLEE are applicable. Thus, our analysis will focus on these two technologies.

JSLEE and SIPServlets are both standards regarding the execution of convergent applications. Both standards define a container-based model for applications, which describes how applications may be built and deployed in their respective containers and how services and features can be supplied by the containers. JSLEE is a more complex specification than SIPServlets. JSLEE includes mechanisms for fault tolerance and high availability, whereas SIPServlets does not. Essentially SIPServlets remain only as a programming model whereas JSLEE aims to be a robust application environment.

In this paper, we are interested in providing guidelines about the convergent service composition. Such labour establishes a set of basic requirements that are the basis of our comparison (see Table 4) in the execution phase: Support for different protocols, Support for reconfiguration, and Model for representing the composition.

Table 4: Comparison of approaches for convergent service execution

Standard	Protocols	Reconfiguration	Model
SipServlet	HTTP and SIP	Timer	Servlet-based
JSLEE	Agnostic	Timer, transactions, failure management, and JMX	Object-based

**Support for different protocols.** Whereas SIPServlet provides support for SIP and HTTP, JSLEE is protocol agnostic; that is to say that JSLEE can be extended to support additional protocols (adding RAs) and external resources with a consistent event model, regardless of protocol/resource. As convergent services are formed by atomic services (using a lot of different protocols) available in the Telecom and Web domains, IMS application servers must support many protocols and APIs (not just SIP).

**Support for reconfiguration.** Consumers of convergent services will expect high quality of service, so an execution environment must be a carrier-grade platform that addresses corresponding requirements of continuous availability and performance. To provide an acceptable degree of availability a mechanism for monitoring and reconfiguration is mandatory due to the inherent changing nature of services available in the Web domain.

About reconfiguration is important to highlight: (i) SIPServlet has no standard management mechanisms defined until now, JSLEE counts with standard management interfaces based on the Java for Management Extension (JMX) independent of the subjacent management protocol. The main supported functions are life cycles, upgrades, profiles, and tracing, (ii) regarding the utilities for applications that can be useful for monitoring and reconfiguration. SIPServlet has the timer function, while JSLEE has: Timer, Trace, Alarm, and Statistics among others, (iii) JSLEE has a container-managed state and wide support for transactions, while SIPServlet has no such functionality, (iv) SIPServlet cannot define a failure model while JSLEE provides a robust failure model via transactions; and (v) while SIPServlets are stateless and the shared state may be stored in a separate session object as a name-value (String, Object) visible within the session, the SBBs may be stateful or stateless. SBB state is private, type safe, transacted and a

property of the SBB itself (i.e. an SBB is an object); the shared state may be stored in a separate context via a type safe interface and the access to the shared state may be specified at deploy time.

***Model for representing the composition.*** As the convergent service composition must be the most automated possible, the model of the subjacent technology must easily support the automation of the composition. In this sense, it is important to noteworthy: (i) Services in SIPServlet are represented as entities based in HTTP Java Servlets without a standard model for composition; and (ii) JSLEE represents services using an Object Oriented model that see the services as SBB, providing support for composition and reuse.

From the previous discussion it is clear that SIPServlet provides support for the adoption of the SIP protocol and the development of simple SIP services. However, convergent services have requirements that exceed the capabilities of SIPServlet. In contrast, JSLEE has been developed to satisfy the requirements of Telecom services in tier-one service providers [91]. This leads us to declare JSLEE as the better option for executing convergent services.

Regarding the automation of the execution phase, JSLEE has still relevant shortcomings. As mentioned previously, the execution phase is inherently automatic, as once the user deploys the service it runs in the execution environment without human intervention. The problem arises when a malfunction or an error appears. In this vein, the role of automated reconfiguration in the convergent service composition is very important. However little has been said about this topic in spite of the existing tools provided by the JSLEE server. Some approaches [141] [142] [105] offer alternatives for JSLEE service reconfiguration based on advanced selection algorithms of services in case of failure. Nonetheless, we think is important to analyze the feasibility on using current approaches of Web service recovery [143] for reconfiguring convergent compositions.

## 8. Research directions

From the above review, we can see that many advances have been made in various research aspects, including automated generation, graphical creation, and execution of convergent services in telecom environments. However, there are still many open research issues that need to be solved before the automated composition can be of practical use. This section presents the

research directions and how existing approaches can be used or contribute to address the challenges of convergent service composition.

**Availability of services.** whereas in the Telecom domain the availability of services is close to 100%, in the Web domain, services can raise, change, or disappear continually. Recent approaches are focusing its efforts on defining approaches for reconfiguring compositions that involve services from both domains. These approaches tackle the ongoing changing in the service libraries and, consider failover mechanisms (e.g., to replace services in failure) that allow the reconfiguration of composite services. These approaches can be divided in two groups:

One group of approaches for reconfiguration aims for solving convergent reconfiguration adapting mechanisms from Web services reconfiguration in convergent domains. These approaches mostly dealt with replacement of failing services, and consequently, most of the existing related approaches are focused on service selection of potential replacement services. In [144] the authors present architectures for automated reconfiguration of convergent services in Telecommunication environments. In order to reduce reconfiguration times while holding the initial constraints, the authors present an iterative algorithm which does not replace individual services but whole regions of services, specified with Hierarchical Tasks Networks (HTNs).

Other group of approaches for convergent reconfiguration adapt services composition according to the changing situation and environment of the user, most of the proposed techniques may be used in the convergent composition: Chen et al. [145] present a model for service composition in high mobility environments. The model for decentralized service composition adapts itself by composing fragments on-demand, as appropriate to the constraints of services costumers and providers. [145]. Neiat et al. [146] present a failure-proof composition model for Sensor-Cloud services based on dynamic features such as spatio-temporal aspects. The authors present a failure-proof composition algorithm based on D\* Lite to handle QoS changes of Sensor-Cloud services at run-time.

**Number of services.** In the Telecom domain, the number of services is relatively low (e.g., messaging, phone calls, location, and video call). In contrast, the number and nature of services in the Web domain is essentially high and changing (e.g., shopping, translator, and social networks). This high number of services turns complex the use of traditional methods, such as SCEs and MDEs, for creating convergent services. Therefore, some of the research approaches analyzed here specially the group of approaches based

on automated planning may be used for discovering convergent services in Telecom infrastructure; for example, Ordonez et al. [109] present a framework for including automated composition approaches based on automated planning in Telecommunications scenarios. On the other hand, recent algorithms such as [147] [148], a uses mechanisms for reducing the services search space by pruning candidates by dominance relationships and constraint validations at candidate level. Finally, some approaches are focused on the use of data mining in services composition. Bayati et al. [149][150] use data mining techniques on Web services in order to find the best composition in a set of possible compositions. These works use techniques such as association, Classification and clustering are for improving performance of the Web Service Composition.

With the increasing presence of web services on the Internet, Quality of Service (QoS) is becoming important for describing non-functional characteristics of services, and is often employed in web service composition [147][151]. In [147] a service composition algorithm using partial selection technique for QoS-Aware Web Service Composition, this algorithm reduces the search space due to the fact that it prunes candidates by dominance relationships and constraint validations at candidate level [147]. Equally Moustafa et al. [148] propose two algorithms based on reinforcement learning for composing and adapting Web services in open and dynamic environments: the first algorithm introduces an offline learning scheme for service composition where one online learning task is transformed into a series of supervised learning steps. The second algorithm presents a coordination mechanism in order to enable multiple agents to learn the service composition task cooperatively. In [152] the authors present a novel multi-objective that addresses QoS-aware Web service composition with conflicting objectives and various restrictions on quality matrices. The approach uses reinforcement learning to deal with the uncertainty inherent in open and decentralized environments. Experimental results reveal the ability of the proposed approach to find a set of Pareto optimal solutions, which have the equivalent quality to satisfy multiple QoS-objectives with different user preferences

The aspect of services discovery is related to the description of services that will be addressed below.

***Services description.*** Though standardized or broadly accepted language do not yet exist for describing convergent services, the trend indicates that, as in Social Web and Telecom network management, REST-based services have turned a de facto standard to publish functionalities, we consider

that existing languages (e.g., WSDL, WADL, and hREST) may be adapted to specific domains. Existing languages such as: BPEL, BPEL for REST, Bite, and LPML offers the base for unifying description of service compositions, specially when these compositions include services based on protocols such as SOAP-XML/HTTP, JSON-XML/HTTP, RSS, Atom, and SIP. Regarding Telecommunication providers, different service providers based on cloud computing, such as Skype or Twilio, may offer its own standard or language. In this scenario the option of Enterprise Service Buses (ESB) acquires high relevance due to its potential for translating different standards. Despite improvements, research for describing Telecom services using well established techniques from the Web services must be expanded. These extensions and improvements will be fundamental to facilitate the evolution of convergent service composition

***Interfaces of user.*** In the Web domain, the user interacts with the system using mostly interfaces controlled with a keyboard and a pointing device or a tactile screen. In contrast, in the Telecom domain, the voice is the primary element to interact with the phone, especially when the user is moving or driving. Therefore, existing academic and industrial approaches based on natural language (both commercial and open source) can be used for providing and consuming convergent services with richer user interfaces. Here, rich user interfaces must be understood as these in which the user can access the convergent applications using keyboards, tactile screens, and voice depending of the user context. For example [153] Natural language processing is used to identify services required to accomplish the user request. This approach represent the identified tasks in model and from this model a user interface (UI) is generated. The authors claim that the system can extract the tasks from how-to instructions Web pages with high precision. The work of Ordoñez et al. [110] present the use of natural language processing in convergent domains by translating user request and context into automated planning representations.

***Context of user.*** In the Web domain, the context and preferences of the user considered during service composition rarely change in real time. Conversely, in the Telecom domain, the user is often in constant movement and the type and capabilities of the device cannot be known beforehand in many cases, which means that the user context must be considered, modeled, and included in the composition process. In this regard, many research works can be applied for particular solutions of convergent composition. For example [154] presents an automated composition systems that incorporates SOA

paradigm and devices implemented by device profile for web service (DPWS). This approach presents a composition engine that considers context information, composition templates, and predefined policies. Equally NGSON [155] is an IEEE sponsored effort to standardize an open service ecosystem that allows composing and delivering services dynamically across different service providers and network operators domains. NGSON uses context information to drive the interactions of the NGSON overlay nodes with service and network operator infrastructures.

***Time constraints and number of protocols.*** During development of convergent composition solutions, the heterogeneity of protocols, networks and time restrictions may be tackled by selecting robust environments for service execution. JSLEE offers a high reliable environment for supporting Telecommunications constraints. JSLEE provides independence of protocol, and interfaces for diverse networks; equally JSLEE considers time constrains proper of telecom domains

## 9. Conclusions

Selection of technologies for specific convergent composition solutions is complex due to the big amount of considerations. In this paper, we have identified the relevant phases for convergent service composition and explored the existing approaches for automating each one of these phases. It has been analyzed that all phases of convergent composition have reached high level of automation using different approaches. Following, as conclusion, we provide general guidelines that help to select the most suitable approach for each domain or problem

***Domain knowledge.*** If the user possesses domain knowledge, he will be able to create a full specification that details the specific tasks and the data/-control flow among them, as well as other non-functional requirements. Conversely, it cannot be expected of a non-expert user to create a full workflow. Thus, for non-expert users, a specification stated in terms of preconditions and effects should be created and used to perform the composition of services. Such specification should be done with domain-independent planning techniques that do not require knowledge of the process model.

***Level of automation required.*** When the number of services and the complexity of services composition increase, automatic composition becomes more desirable. There are different variations; for example, a semi-automated

approach can generate basic templates automatically that are later presented to the user so he/she chooses one of them and refines it manually.

***Context of device and user.*** If the user requires mobility and does not have advanced knowledge on programming, he/she may use Web-enabled graphical interfaces provided by mashup-based approaches or alternatively even consider voice-based interfaces. Conversely, if the domain is oriented to expert users that do not have such constraints, SCE-based approaches are the best choice.

***Dynamic domains.*** In convergent composition, services may become unavailable or be redefined by their provider. This consideration justifies the need for continuous monitoring and reconfiguration during the execution. In workflow-based approaches, this means that the user must supervise the execution process and perform the necessary changes; in automatic approaches the system must react and adapt itself to the circumstances with as little human intervention as possible. For convergent domains, JSLEE offers a set of features that could ease reconfiguration implementation: standard management interfaces based on JMX, independence of management protocol, and interfaces for managing applications. Also, it is important to highlight that some AI-based planning platforms include execution and replanning, which may alleviate the reconfiguration problem to some extent. These AI-based approaches are an interesting alternative, but they need to consider the complexity and performance requirements of the Telecom domain.

## Acknowledgements

The research of PhD(c) Caicedo is supported by the PECPG (Agreement Program Students Graduate) of the CAPES (Brazil) and the University of Cauca (Colombia).

## References

- [1] Z. Luo, J. Wang, Q. Zhu, Self-aware Services of NGSDP: Using Bayesian Networks for Measuring Quality of Convergent Services, *Journal of Internet Services and Information Security (JISIS)* 1 (1) (2011) 46–58.
- [2] S. Moyer, A. Umar, The impact of Network Convergence on Telecommunications Software, *Communications Magazine, IEEE* 39 (1) (2001) 78–84.



- [3] M. Vrdoljak, S. I. Vrdoljak, G. Skugor, Fixed-mobile Convergence Strategy: technologies and market opportunities, *Communications Magazine*, IEEE 38 (2) (2000) 116–121.
- [4] J. Simoes, S. Wahle, The Future of Services in Next Generation Networks, *IEEE Potentials* 30 (1) (2011) 24–29.
- [5] C. dos Santos, R. Bezerra, J. Ceron, L. Granville, L. Rockenbach Tarouco, On Using Mashups for Composing Network Management Applications, *Communications Magazine*, IEEE 48 (12) (2010) 112–122.
- [6] O. M. C. Rendon, C. R. P. dos Santos, A. S. Jacobs, L. Z. Granville, Monitoring Virtual Nodes using mashups, *Computer Networks* 64 (0) (2014) 55 – 70.
- [7] M. Saadatmand, A. Cicchetti, D. Corcoran, M. Sjodin, Toward a Tailored modeling of Non-functional Requirements for Telecommunication Systems, in: *Information Technology: New Generations (ITNG)*, 2011 Eighth International Conference on, IEEE, 2011, pp. 1044–1045.
- [8] R. Berkan, B. Upadhyaya, L. Tsoukalas, R. Kisner, R. Bywater, Advanced Automation Concepts for Large-scale Systems, *Control Systems*, IEEE 11 (6) (1991) 4–12.
- [9] K. Sycara, M. Paolucci, A. Ankolekar, N. Srinivasan, Automated Discovery, Interaction and Composition of Semantic Web Services, *Web Semantics: Science, Services and Agents on the World Wide Web* 1 (1) (2003) 27–46.
- [10] E. Kaldeli, A. Lazovik, M. Aiello, Continual Planning with Sensing for Web Service Composition, in: *AAAI*, 2011.
- [11] O. Hatzi, D. Vrakas, M. Nikolaidou, N. Bassiliades, D. Anagnostopoulos, L. Vlahavas, An Integrated Approach to Automated Semantic Web Service Composition through Planning, *Services Computing*, IEEE Transactions on 5 (3) (2012) 319–332.
- [12] O. M. C. Rendon, F. Estrada-Solano, L. Z. Granville, A Mashup-Based Approach for Virtual SDN Management, in: *Computer Software and*

Applications Conference (COMPSAC), 2013 IEEE 37th Annual, 2013, pp. 143–152.

- [13] O. Caicedo Rendon, F. Estrada-Solano, L. Zambenedetti Granville, A Mashup Ecosystem for Network Management Situations, in: Global Communications Conference (GLOBECOM), 2013 IEEE, 2013, pp. 2249–2255.
- [14] O. Caicedo Rendon, F. Estrada Solano, L. Zambenedetti Granville, An Approach to Overcome the Complexity of Network Management Situations by Mashments, in: Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on, 2014, pp. 875–883.
- [15] P. Corporation, Voice Controlled Internet Mashups (2012).  
URL <http://www.zypr.net/>
- [16] Apple, Use Your Voice to Do Even More with Siri (2013).  
URL <http://www.apple.com/ios/siri/>
- [17] M. Belaunde, P. Falcarin, Realizing an MDA and SOA Marriage for the Development of Mobile Services, in: Model Driven Architecture–Foundations and Applications, Springer, 2008, pp. 393–405.
- [18] P. Falcarin, Service Composition Quality Evaluation in SPICE Platform, in: High assurance services computing, Springer, 2009, pp. 89–102.
- [19] G. Bond, E. Cheung, I. Fikouras, R. Levenshteyn, Unified Telecom and Web Services Composition: problem definition and future directions, in: Proceedings of the 3rd International Conference on Principles, Systems and Applications of IP Telecommunications, ACM, 2009, p. 13.
- [20] H. Gebhardt, M. Gaedke, F. Daniel, S. Soi, F. Casati, C. A. Iglesias, S. Wilson, From Mashups to Telco Mashups: a survey, IEEE internet computing 16 (3) (2012) 70–76.
- [21] V. Agarwal, G. Chaffe, S. Mittal, B. Srivastava, Understanding Approaches for Web Service Composition and Execution, in: Proceedings of the 1st Bangalore Annual Compute Conference, COMPUTE '08, ACM, New York, NY, USA, 2008, pp. 1:1–1:8.

- [22] J. Ma, J. V. Nickerson, Hands-on, Simulated, and Remote Laboratories: A Comparative Literature Review, *ACM Comput. Surv.* 38 (3).
- [23] R. Trapero Burgos, D. Fuentes, J. del Alamo Ramiro, A. Martin, Y. Garcia, I. Arnal, A. Reol, J. Garcia, TelecomI+D02: Next Generation Mashups: How to Create my Own Services in a Convergent World, *Latin America Transactions, IEEE (Revista IEEE America Latina)* 7 (3) (2009) 390–394.
- [24] J. C. Yelmo, J. M. del Álamo, R. Trapero, P. Falcarin, J. Yi, B. Cairo, C. Baladrón, A User-centric Service Creation Approach for Next Generation Networks, in: *Innovations in NGN: Future Network and Services, 2008. K-INGN 2008. First ITU-T Kaleidoscope Academic Conference, IEEE, 2008*, pp. 211–218.
- [25] S. Hall, Evolving the Service Creation Environment, in: *Intelligence in Next Generation Networks (ICIN), 2010 14th International Conference on*, 2010, pp. 1–6.
- [26] C. Baladrón, J. Aguiar, B. Carro, L.-W. Goix, A. León, P. F. Martín, J. Sienel, User-centric Future Internet and Telecommunication Services, *Towards the Future Internet* (2009) 217–226.
- [27] J. C. Yelmo, J. M. del Álamo, R. Trapero, Y.-S. Martín, A User-centric Approach to Service Creation and Delivery over Next Generation Networks, *Computer Communications* 34 (2) (2011) 209–222.
- [28] D. Zhu, Y. Zhang, B. Cheng, B. Wu, J. Chen, HSCEE: A Highly Flexible Environment for Hybrid Service Creation and Execution in Converged Networks, *Journal of Convergence Information Technology* 6 (3).
- [29] Y. Zhang, Dependable ESB Routing in Hybrid Service Execution Environment, *AISS* 2 (1) (2010) 83–93.
- [30] J. Yu, Q. Z. Sheng, P. Falcarin, A Visual Semantic Service Browser Supporting User-Centric Service Composition, in: *AINA, IEEE Computer Society, 2010*, pp. 244–251.
- [31] O. Droegehorn, I. Konig, G. Le-Jeune, J. Cupillard, M. Belaunde, E. Kovacs, Professional and End-user-driven Service Creation in the

- SPICE platform, in: World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a, IEEE, 2008, pp. 1–8.
- [32] M. Shiaa, P. Falcarin, A. Pastor, F. Lécué, E. Goncalves da Silva, L. Ferreira Pires, Towards the Automation of the Service Composition Process, in: P. Cunningham, M. Cunningham (Eds.), ICT-MobileSummit 2008 Conference Proceedings, IIMC International Information Management Corporation, 2008, pp. 1–8.
- [33] V. Agarwal, S. Goyal, S. Mittal, S. Mukherjea, J. Ponzio, F. Shah, A Middleware Framework for Mashing Device and Telecom Features with the Web, IBM Research, RI 10009.
- [34] P. de Vrieze, L. Xu, A. Bouguettaya, J. Yang, J. Chen, Building Enterprise Mashups, *Future Generation Computer Systems* 27 (5) (2011) 637–642.
- [35] A. Yates, O. Etzioni, D. Weld, A Reliable Natural Language Interface to Household Appliances, in: Proceedings of the 8th international conference on Intelligent user interfaces, ACM, 2003, pp. 189–196.
- [36] M. Cremene, J.-Y. Tigli, S. Laviotte, F.-C. Pop, M. Riveill, G. Rey, Service Composition Based on Natural Language Requests, in: Services Computing, 2009. SCC'09. IEEE International Conference on, IEEE, 2009, pp. 486–489.
- [37] N. Laga, E. Bertin, R. Glitho, N. Crespi, Widgets and Composition Mechanism for Service Creation by Ordinary Users, *Communications Magazine, IEEE* 50 (3) (2012) 52–60.
- [38] J. C. Corrales, P. Falcarin, User Centred Automated Composition in Telco 2.0, in: SERVICE COMPUTATION 2012, The Fourth International Conferences on Advanced Service Computing, 2012, pp. 104–110.
- [39] A. Ordonez, V. Alcázar, D. Borrajo, P. Falcarin, J. C. Corrales, An Automated User-Centered Planning Framework for Decision Support in Environmental Early Warnings, in: Advances in Artificial Intelligence—IBERAMIA 2012, Springer, 2012, pp. 591–600.

- [40] O. M. Group, IP Multimedia Subsystem (2013).  
URL <http://www.omg.org/spec/TelcoML/>
- [41] O. M. Group, UML Infrastructure (2011).  
URL <http://www.omg.org/spec/UML/>
- [42] B. . OMG, Business Process Modeling Notation Specification (2011).  
URL <http://www.omg.org/spec/BPMN/2.0/>
- [43] Radysis, RadySis Service Creation Environment (2012).  
URL <http://www.radisys.com/services>
- [44] A. Duke, S. Stincic, J. Davies, G. Á. Rey, C. Pedrinaci, M. Maleshkova, J. Domingue, D. Liu, F. Lecue, N. Mehandjiev, Telecommunication Mashups using RESTful Services, in: Towards a Service-Based Internet, Springer, 2010, pp. 124–135.
- [45] P. Falcarin, C. Venezia, Communication Web Services and JAIN-SLEE Integration Challenges, International Journal of Web Services Research (IJWSR) 5 (4) (2008) 59–78.
- [46] K. Christos, C. Vassilakis, E. Rouvas, P. Georgiadis, Qos-driven Adaptation of BPEL Scenario Execution, in: Web Services, 2009. ICWS 2009. IEEE International Conference on, IEEE, 2009, pp. 271–278.
- [47] Web Services Business Process Execution Language Version 2.0, Tech. rep., OASIS Web Services Business Process Execution Language (april 2007).  
URL <http://docs.oasis-open.org/wsbpel/2.0/OS>
- [48] Y. Bo, Z. Da, Z. Yang, C. Junliang, The Design of an Orchestrated Execution Environment Based on JBI, in: Computer Science-Technology and Applications, 2009. IFCSTA'09. International Forum on, Vol. 3, IEEE, 2009, pp. 367–371.
- [49] T. Eichelmann, W. Fuhrmann, U. Trick, B. Ghita, Creation of Value Added Services in NGN with BPEL, in: Proceedings of the Fourth Collaborative Research Symposium on Security, E-learning, Internet and Networking (SEIN 2008), Wrexham, United Kingdom, 2008, pp. 5–9.

- [50] T. Eichelmann, W. Fuhrmann, U. Trick, B. Ghita, Support of Parallel BPEL Activities for the TeamCom Service Creation Platform for Next Generation Networks, Proc. of SEIN (2009) 69–80.
- [51] A. V. Riabov, E. Boillet, M. D. Feblowitz, Z. Liu, A. Ranganathan, Wishful Search: Interactive Composition of Data Mashups, in: Proceedings of the 17th international conference on World Wide Web, ACM, 2008, pp. 775–784.
- [52] A. Lehmann, T. Eichelmann, U. Trick, R. Lasch, B. Ricks, R. Toenjes, TeamCom: a Service Creation Platform for Next Generation Networks, in: Internet and Web Applications and Services, 2009. ICIW'09. Fourth International Conference on, IEEE, 2009, pp. 12–17.
- [53] T. Eichelmann, W. Fuhrmann, U. Trick, B. Ghita, Enhanced Concept of the TeamCom SCE for Automated Generated Services based on JSLEE, in: Proceedings of the Eighth International Network Conference (INC 2010), U. Bleimann, PS Dowland, S. Furnell, and O. Schneider, Eds., ISBN, 2010, pp. 978–1.
- [54] Hoiio, Hoiio Framework (2014).  
URL <http://content.hoiio.com/>
- [55] Tropo, Tropo Framework (2014).  
URL <https://www.tropo.com/>
- [56] Callfire, Callfire framework (2014).  
URL <https://www.callfire.com/>
- [57] Skype, Skype API (2014).  
URL [https://support.skype.com/en/category/SKYPE\\_API/](https://support.skype.com/en/category/SKYPE_API/)
- [58] Twilio, Twilio framework (2014).  
URL <https://www.twilio.com/>
- [59] J. Niemöller, E. Freiter, K. Vandikas, R. Quinet, R. Levenshteyn, I. Fikouras, Composition in Converged Service Networks: Requirements and Solutions, in: International Workshop on Business Systems Management and Engineering (BSME 2010), TOOLS Federated Conferences, 2010.

- [60] E. T. S. Institute, The ETSI OSA Parlay-X 3.0 Specifications (2008).  
URL [http://etsi.org/deliver/etsi\\_es/202500\\_202599/20250406](http://etsi.org/deliver/etsi_es/202500_202599/20250406)
- [61] GSMA, Gsma-oneapi for developers (2013).  
URL <http://www.gsma.com/oneapi/developers>
- [62] T. G. P. P. G. T. S. Group, IP Multimedia Subsystem (2011).  
URL <http://www.3gpp.org/DynaReport/23228.htm>
- [63] Oracle, JSR 289: Java Specification Requests SIP Servlet v1.1 (2008).  
URL <http://jcp.org/aboutJava/communityprocess/final/jsr289>
- [64] C. Kun, J. Xu, S. Reiff-Marganiec, Markov-htn Planning Approach to Enhance Flexibility of Automatic Web Service Composition, in: Web Services, 2009. ICWS 2009. IEEE International Conference on, IEEE, 2009, pp. 9–16.
- [65] J. El Hadad, M. Manouvrier, M. Rukoz, TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition, Services Computing, IEEE Transactions on 3 (1) (2010) 73–85.
- [66] S. Sohrabi, N. Prokoshyna, S. A. McIlraith, Web Service Composition via the Customization of Golog Programs with User Preferences, in: Conceptual Modeling: Foundations and Applications, Springer, 2009, pp. 319–334.
- [67] S. Sohrabi, J. A. Baier, S. A. McIlraith, HTN Planning with Preferences, in: Twenty-First International Joint Conference on Artificial Intelligence, 2009.
- [68] S. Shanbhag, X. Huang, S. Proddatoori, T. Wolf, Automated Service Composition in Next-generation Networks, in: Distributed Computing Systems Workshops, 2009. ICDCS Workshops' 09. 29th IEEE International Conference on, IEEE, 2009, pp. 245–250.
- [69] E. Sirin, B. Parsia, D. Wu, J. Hendler, D. Nau, HTN Planning for Web Service Composition using SHOP2, Web Semantics: Science, Services and Agents on the World Wide Web 1 (4) (2004) 377–396.

- [70] X. Song, W. Dou, J. Chen, A Workflow Framework for Intelligent Service Composition, *Future Generation Computer Systems* 27 (5) (2011) 627–636.
- [71] E. Zahoor, O. Perrin, C. Godart, Rule-based Semi Automatic Web Services Composition, in: *Services-I, 2009 World Conference on*, IEEE, 2009, pp. 805–812.
- [72] K. Fujii, T. Suda, Semantics-based Context-aware Dynamic Service Composition, *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 4 (2) (2009) 12.
- [73] P. Yue, Automatic Service Composition, in: *Semantic Web-based Intelligent Geospatial Web Services*, Springer, 2013, pp. 21–25.
- [74] C. dos Santos, R. Bezerra, J. Ceron, L. Granville, L. Tarouco, Botnet Master Detection using a Mashup-based Approach, in: *Network and Service Management (CNSM), 2010 International Conference on*, 2010, pp. 390–393.
- [75] C. dos Santos, R. Bezerra, L. Granville, L. Bertholdo, W. Cheng, N. Anerousis, A Data Confidentiality Architecture for Developing Management Mashups, in: *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, 2011, pp. 49–56.
- [76] D. L. Phuoc, M. Hauswirth, Linked Open Data in Sensor Data Mashups, in: *Workshop on semantic sensor networks, in conjunction with ISWC, CEUR*, 2009, pp. 1–16.
- [77] G. Meditskos, N. Bassiliades, A Combinatory Framework of Web 2.0 Mashup Tools, OWL-S and UDDI, *Expert Syst. Appl.* 38 (6) (2011) 6657–6668.
- [78] A. Malki, S. M. Benslimane, Semantic Cloud: Building Dynamic Mashup in Cloud Environment, *International Journal of Information Technology and Web Engineering (IJITWE)* 8 (4) (2013) 20–35.
- [79] R. Hull, M. Benedikt, V. Christophides, J. Su, E-services: A Look Behind the Curtain, in: *Proceedings of the Twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS '03*, ACM, New York, NY, USA, 2003, pp. 1–14.



- [80] G. Alonso, F. Casati, H. Kuno, V. Machiraju, Web Services. Concepts, Architectures and Applications, Springer-Verlag, 2004.
- [81] W3C Working Group, Web Services Architecture Requirements (2004). URL <http://www.w3.org/TR/wsa-reqs/>
- [82] I. Grida Ben Yahia, E. Bertin, J.-P. Deschrevel, N. Crespi, Service Definition for Next Generation Networks, in: Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on, 2006, pp. 22–22.
- [83] J. Gozdecki, A. Jajszczyk, R. Stankiewicz, Quality of Service Terminology in IP Networks, Communications Magazine, IEEE 41 (3) (2003) 153–159.
- [84] T. Forum, TMF Service Framework GB924 V1.9 (2004). URL <http://www.zypr.net/>
- [85] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, Web Service Definition Language (WSDL), Tech. rep. (Mar. 2001). URL <http://www.w3.org/TR/wsdl>
- [86] C. Granell, R. Lemmens, M. Gould, A. Wytzisk, R. de By, P. van Oosterom, Integrating Semantic and Syntactic Descriptions to Chain Geographic Services, Internet Computing, IEEE 10 (5) (2006) 42–52.
- [87] M. J. Hadley, Web Application Description Language (WADL), Tech. rep., Mountain View, CA, USA (2006).
- [88] R. Chinnici, J.-J. Moreau, A. Ryman, S. Weerawarana, Web Services Description Language version 2.0 part 1: Core Language, W3C Recommendation 26 (2007) 19.
- [89] J. Kopecký, K. Gomadam, T. Vitvar, hRESTS: An HTML Microformat for Describing RESTful Web Services, in: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT '08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 619–625.

- [90] D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, et al., Bringing Semantics to Web Services: The OWL-S Approach, in: *Semantic Web Services and Web Process Composition*, Springer, 2005, pp. 26–42.
- [91] C. Chrighton, D. Long, D. Page, JAIN SLEE vs SIP Servlet Which is the best choice for an IMS Application Server?, in: *Telecommunication Networks and Applications Conference, 2007. ATNAC 2007. Australasian*, IEEE, 2007, pp. 448–453.
- [92] K. Trivedi, D. Wang, D. J. Hunt, A. Rindos, W. E. Smith, B. Vashaw, Availability modeling of SIP protocol on IBM® Websphere®, in: *Dependable Computing, 2008. PRDC'08. 14th IEEE Pacific Rim International Symposium on*, IEEE, 2008, pp. 323–330.
- [93] Notation, Business Process Modeling, Specification, Final Adopted Specification, Tech. rep., Technical report, Object Management Group (OMG) (2006).
- [94] C. Atkinson, T. Kühne, Rearchitecting the UML infrastructure, *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 12 (4) (2002) 290–321.
- [95] C. Pautasso, RESTful Web Service Composition with BPEL for REST, *Data Knowl. Eng.* 68 (9) (2009) 851–866.
- [96] F. Rosenberg, F. Curbera, M. Duftler, R. Khalaf, Composing RESTful Services and Collaborative Workflows: A Lightweight Approach, *Internet Computing, IEEE* 12 (5) (2008) 24 –31.
- [97] R. Krummenacher, B. Norton, E. Simperl, C. Pedrinaci, SOA4All: Enabling Web-scale Service Economies, in: *Semantic Computing, 2009. ICSC '09. IEEE International Conference on*, 2009, pp. 535 –542.
- [98] E. M. Maximilien, A. Ranabahu, S. Tai, Swashup: Situational Web Applications Mashups, in: *OOPSLA*, ACM, Montreal, Canada, 2007, pp. 797–798.
- [99] C. Cappiello, F. Daniel, M. Matera, C. Pautasso, Information Quality in Mashups, *IEEE Internet Computing Magazine* 14 (4) (2010) 14–22.

- [100] W. Jouve, N. Palix, C. Consel, P. Kadionik, A SIP-based Programming Framework for Advanced Telephony Applications, in: Principles, Systems and Applications of IP Telecommunications. Services and Security for Next Generation Networks, Springer, 2008, pp. 1–20.
- [101] G. Branca, L. Atzori, A survey of SOA technologies in NGN network architectures, Communications Surveys & Tutorials, IEEE 14 (3) (2012) 644–661.
- [102] A. C. Santos, L. Tarrataca, J. M. Cardoso, D. R. Ferreira, P. C. Diniz, P. Chainho, Context Inference for Mobile Applications in the UPCASE Project, in: MobileWireless Middleware, Operating Systems, and Applications, Springer, 2009, pp. 352–365.
- [103] Vlingo, Vlingo System (2014).  
URL <http://www.vlingo.com/>
- [104] S.-C. Oh, D. Lee, S. R. Kumara, A Comparative Illustration of AI Planning-based Web Services Composition, ACM SIGecom Exchanges 5 (5) (2006) 1–10.
- [105] O. Caicedo, F. Hoyos, A. Munoz, J. Corrales, A Telecommunication Service Reconfiguration Architecture Based on JSLEE, in: Communications Conference (COLCOM), 2012 IEEE Colombian, 2012, pp. 1–6.
- [106] D. E. Simmen, M. Altinel, V. Markl, S. Padmanabhan, A. Singh, Damia: Data Mashups for Intranet Applications, in: ACM SIGMOD, ACM, Vancouver, Canada, 2008, pp. 1171–1182.
- [107] K. Huang, Y. Fan, W. Tan, An Empirical Study of Programmable Web: A Network Analysis on a Service-Mashup System, in: ICWS, IEEE, Honolulu, HI, 2012, pp. 552–559.
- [108] R. Bezerra, C. dos Santos, L. Bertholdo, L. Granville, L. Tarouco, On the Feasibility of Web 2.0 Technologies for Network Management: A Mashup-based Approach, in: NOMS, IEEE, Osaka, Japan, 2010, pp. 487–494.
- [109] A. Ordóñez, V. Alcázar, J. C. Corrales, P. Falcarin, Automated context aware composition of advanced telecom services for environmental early

warnings, *Expert Syst. Appl* 41 (13) (2014) 5907–5916.  
URL <http://dx.doi.org/10.1016/j.eswa.2014.03.045>

- [110] A. Ordóñez, J. C. Corrales, P. Falcarin, Natural language processing based services composition for environmental management, in: *SoSE*, IEEE, 2012, pp. 497–502.  
URL <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6324004>
- [111] A. Kim, M. Kang, C. Meadows, E. Ioup, J. Sample, A Framework for Automatic Web Service Composition, Tech. rep., DTIC Document (2009).
- [112] R. J. A. Ali, O. F. Rana, D. W. Walker, S. Jha, S. Sohail, G-QoS: Grid Service Discovery using QoS Properties, *Computing and Informatics* 21 (4) (2012) 363–382.
- [113] M. Junghans, S. Agarwal, R. Studer, Towards Practical Semantic Web Service Discovery, in: *The Semantic Web: Research and Applications*, Springer, 2010, pp. 15–29.
- [114] X. Wu, C. Chen, H. Huang, A survey on web service composition: from service description, automatic process generation to process evaluation., *International Journal of Digital Content Technology & its Applications* 6 (17).
- [115] H. Xiao, Y. Zou, R. Tang, J. Ng, L. Nigul, Ontology-driven service composition for end-users, *Service Oriented Computing and Applications* 5 (3) (2011) 159–181.
- [116] F. Slaimi, A. B. Hassine, M. Tagina, Ontology based vertical web service composition, *International Journal of Knowledge-based and Intelligent Engineering Systems* 18 (1) (2014) 1–12.
- [117] K. Ren, N. Xiao, J. Chen, Building quick service query list using wordnet and multiple heterogeneous ontologies toward more realistic service composition, *Services Computing, IEEE Transactions on* 4 (3) (2011) 216–229.

- [118] V. Beltran, K. Arabshian, H. Schulzrinne, Ontology-based user-defined rules and context-aware service composition system, in: *The Semantic Web: ESWC 2011 Workshops*, Springer, 2012, pp. 139–155.
- [119] J. Zhang, P. Lin, P. Huang, B. Kuang, H. Huang, Review on semantic web service composition., *Advances in Information Sciences & Service Sciences* 5 (8).
- [120] X. Tang, C. Jiang, M. Zhou, Automatic web service composition based on horn clauses and petri nets, *Expert Systems with Applications* 38 (10) (2011) 13024–13031.
- [121] Y. Li, X. Yu, R. Wang, L. Geng, L. Wang, A study on dynamic semantic web service composition, *Web Intelligence and Agent Systems* 11 (2) (2013) 125–135.
- [122] N. H. Priya, S. Chandramathi, Qos based optimal selection of web services using fuzzy logic, *Journal of Emerging Technologies in Web Intelligence* 6 (3) (2014) 331–339.
- [123] S. J. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd Edition, Pearson Education, 2003.
- [124] S. Rabah, D. Ni, P. Jahanshahi, L. F. Guzman, Current State and Challenges of Automatic Planning in Web Service Composition, arXiv preprint arXiv:1107.1932.
- [125] T. Vossen, M. Ball, R. H. Smith, On the Use of Integer Programming Models in AI Planning, in: *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, 1999, pp. 304–309.
- [126] C. Bo, G. Jie, C. Junliang, L. Xiangtao, A Preliminary Practice for BPEL based Multimedia Conference Web Services Orchestration, in: *Telecommunications, 2008. AICT'08. Fourth Advanced International Conference on*, IEEE, 2008, pp. 321–326.
- [127] Y. Chen, G.-K. Ni, C.-H. Kuo, C.-Y. Lin, A BPEL-Based Fault-handling Architecture for Telecom Operation Support Systems, *Journal of Advanced Computational Intelligence* Vol 14 (5).

- [128] G. R. Sierra, F. Martínez, O. M. Caicedo, L. González, Architecture for provisioning Added Value Services supported on ESB and SLEE, *Revista Educación en Ingeniería* 7 (13) (2012) 1–12.
- [129] H. Khlifi, J.-C. Gregoire, IMS Application Servers: Roles, Requirements, and Implementation Technologies, *Internet Computing, IEEE* 12 (3) (2008) 40–51.
- [130] M. Femminella, R. Francescangeli, F. Giacinti, E. Maccherani, A. Parisi, G. Reali, Scalability and Performance Evaluation of a JAIN SLEE-based Platform for VoIP services, in: *Teletraffic Congress, 2009. ITC 21 2009. 21st International, 2009*, pp. 1–8.
- [131] J. Deruelle, JSLEE and SIP-servlets interoperability with Mobicents communication platform, in: *Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST'08. The Second International Conference on, IEEE, 2008*, pp. 634–639.
- [132] OpenCloud, The Rhino Platform (2014).  
URL [www.opencloud.com/products/rhino-application-server/](http://www.opencloud.com/products/rhino-application-server/)
- [133] T. Dinsing, G. A. Eriksson, I. Fikouras, K. Gronowski, R. Levenshteyn, P. Pettersson, P. Wiss, Service composition in IMS using Java EE SIP servlet containers, *Ericsson Review* 3 (2007) 92–96.
- [134] J. M. E. Carlin, D. Thissen, A SIP Application Router for Presence-Driven Composition of IMS Services, in: *Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST'08. The Second International Conference on, IEEE, 2008*, pp. 207–212.
- [135] D. Zhu, Y. Zhang, J. Chen, B. Cheng, Enhancing ESB based Execution Platform to Support Flexible Communication Web Services over Heterogeneous Networks, in: *Communications (ICC), 2010 IEEE International Conference on, IEEE, 2010*, pp. 1–6.
- [136] OpenCloud, A Unified Service Platform: ZXUP10 (2014).  
URL [www.zte.com.cn/](http://www.zte.com.cn/)
- [137] B. Cheng, J. Guo, X. Meng, J. Chen, Real-Time Hybrid Services Orchestration Solutions, in: *Computer and Information Technology*

- Workshops, 2008. CIT Workshops 2008. IEEE 8th International Conference on, IEEE, 2008, pp. 678–682.
- [138] M.-T. Schmidt, B. Hutchison, P. Lambros, R. Phippen, The Enterprise Service Bus: making service-oriented architecture real, *IBM Systems Journal* 44 (4) (2005) 781–797.
  - [139] S. Bessler, J. Zeiss, R. Gabner, J. Gross, An Orchestrated Execution Environment for Hybrid Services, in: *Kommunikation in Verteilten Systemen (KiVS)*, Springer, 2007, pp. 77–88.
  - [140] B. Cheng, X. Lin, X. Hu, J. Chen, Formal Analysis for Multimedia Conferencing Communication Services Orchestration, in: *Web Services, 2009. ICWS 2009. IEEE International Conference on*, IEEE, 2009, pp. 1010–1011.
  - [141] T. Yu, Y. Zhang, K.-J. Lin, Efficient Algorithms for Web Services Selection with end-to-end QoS Constraints, *ACM Transactions on the Web (TWEB)* 1 (1) (2007) 6.
  - [142] S.-Y. Hwang, E.-P. Lim, C.-H. Lee, C.-H. Chen, Dynamic Web Service Selection for Reliable Web Service Composition, *Services Computing, IEEE Transactions on* 1 (2) (2008) 104–116.
  - [143] L. Gao, S. D. Urban, J. Ramachandran, A Survey of Transactional Issues for Web Service Composition and Recovery, *International Journal of Web and Grid Services* 7 (4) (2011) 331–356.
  - [144] A. Ordóñez, H. Ordóñez, C. Figueroa, C. Cobos, J. Corrales, Dynamic reconfiguration of composite convergent services supported by multimodal search, in: W. Abramowicz (Ed.), *Business Information Systems*, Vol. 208 of *Lecture Notes in Business Information Processing*, Springer International Publishing, 2015, pp. 127–139. doi: 10.1007/978-3-319-19027-3\_11.  
URL [http://dx.doi.org/10.1007/978-3-319-19027-3\\_11](http://dx.doi.org/10.1007/978-3-319-19027-3_11)
  - [145] N. Chen, S. Clarke, A dynamic service composition model for adaptive systems in mobile computing environments, in: *Service-Oriented Computing*, Springer, 2014, pp. 93–107.

- [146] A. G. Neiat, A. Bouguettaya, T. Sellis, H. Dong, Failure-proof spatio-temporal composition of sensor cloud services, in: *Service-Oriented Computing*, Springer, 2014, pp. 368–377.
- [147] Y. Chen, J. Huang, C. Lin, Partial selection: An efficient approach for qoS-aware web service composition, in: *ICWS*, IEEE, 2014, pp. 1–8.  
URL <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6923037>
- [148] A. Moustafa, M. Zhang, Learning efficient compositions for qos-aware service provisioning, in: *Web Services (ICWS), 2014 IEEE International Conference on*, IEEE, 2014, pp. 185–192.
- [149] S. Bayati, A. Bahreininejad, A. F. Nejad, S. Kharazmi, Improving semantic web services composition performance, using data mining techniques, *Journal of Algorithms & Computational Technology* 4 (4) (2010) 409–423.
- [150] Z. D. Zhao, Y. H. Cui, J. J. Li, Composition oriented semantic relationships mining framework research, *Applied Mechanics and Materials* 513 (2014) 470–473.
- [151] Z. Ye, A. Bouguettaya, X. Zhou, Qos-aware cloud service composition using time series, in: *Service-Oriented Computing*, Springer, 2013, pp. 9–22.
- [152] A. Moustafa, M. Zhang, Multi-objective service composition using reinforcement learning, in: *Service-Oriented Computing*, Springer, 2013, pp. 298–312.
- [153] B. Upadhyaya, Y. Zou, S. Wang, J. Ng, Automatically composing services by mining process knowledge from the web, in: *Service-Oriented Computing*, Springer, 2013, pp. 267–282.
- [154] S. N. Han, G. M. Lee, N. Crespi, Semantic context-aware service composition for building automation system, *Industrial Informatics*, *IEEE Transactions on* 10 (1) (2014) 752–761.
- [155] F. Paganelli, M. Ulema, B. Martini, Context-aware service composition and delivery in ngsons over sdn, *Communications Magazine*, *IEEE* 52 (8) (2014) 97–105.