Evaluating Web Service Composition Methods: the Need for Including Multi-Actor Elements

Ralph W. Feenstra, Marijn Janssen, and René W. Wagenaar
Faculty of Technology, Delft University of Technology, The Netherlands
r.w.feenstra@tudelft.nl
m.f.w.h.a.janssen@tudelft.nl

Abstract:. New systems can be designed by composing them out-of-existing software components which are accessible as web services and provided by the service providers. Governmental organizations can act as service providers by providing information or functionality like authenticating. The basic premise is that by reusing components, development and maintenance costs can be lowered and flexibility is created. As such, public agencies are looking for support to create new compositions. Several composition approaches can be found in the literature, however none of these evaluations take into account the e-government specific requirements originating from the involvement of multiple parties having different interests. In this paper we present a composition evaluation approach which extends the existing evaluation approaches by including the multi-actor dimension. We illustrate this method using an example. Further research is aimed at executing the proposed approach and comparing semantic and multi-actor-based compositions methods.

Keywords: web service, web service composition, evaluation, workshop, multi-actor networks.

1. Introduction

High quality experiences with responsive, integrated private sector information systems are leading citizens to expect the same from public bodies and agencies (Hazlett & Hill, 2003). A great long-term public sector concern is the need for focusing on effective and efficient systems (Beynon-Davies & Williams, 2003). Efficient service provisioning can be accomplished by reusing existing software components to avoid a large duplication of efforts. New systems can be constructed by reusing the already available software components. This duplication of efforts is founded in the highly fragmented and unrelated computerized applications within the same public organization. Many of these applications overlap in functionality and content and are in fact 'isolated islands'. With the advent of web service technology, the functionality and content can be opened and remotely accessed. Yet, the benefits of reusing information and functionality have not been attained as creating service composition is difficult (Feenstra, Janssen, & Wagenaar, 2006).

A modular approach to information systems engineering has become more popular in recent years. The basic idea is to break a system down into parts, design these parts individually and construct a new system by composing the single parts (Fremantle, Weerawarana, & Khalaf, 2002). In e-government public agencies can make their information and systems functionality available using web services, or services for short. As such public agencies act as service providers and citizens and business and other agencies act as service requesters. Often many services provided by multiple service providers need to be combined, e.g. composed, to create a new service.

The service-oriented paradigm enables the creation of a class of modular enterprise services that allows to create services that are modular, accessible, well-described, implementation-independent and interoperable (Fremantle, Weerawarana, & Khalaf, 2002). Developers can then create new systems by combining available web services provided by the service providers, resulting in a composition. The creation of compositions is supported by several compositions methods available in literature (e.g. Beek, Bucchiarone, & Gnesi, 2006; Maximilien & Singh, 2004; e.g. Milanovic & Malek, 2004). Yet, none of them are focused on situations containing multiple actors, e.g. departments and organizations, with is the case in e-government. In e-government, often no single department or agency can provide integrated service delivery, as they lack the resources, do not have all the required information or do not want to have all software in-house. As such, these organizations are looking for a composition method to support the creation of new service compositions. We make a distinction between a composition approach and a composition method. An composition approach is the philosophy and main idea. A composition method is the implementation of an approach and includes the subsequent steps that need to be taken. For example, the concept of a service ontology can be a part of a composition approach, the concrete steps taken to find a candidate service that meets certain requirements can be part of a composition method.

In this paper we propose an approach aimed at evaluating composition methods. This approach extends currently available evaluation methods by adding the multi-actor dimension. This is especially important for

Reference this paper as:
Feenstra, R. W., Janssen, M., and Wagenaar, R. W. (2007) "Evaluating Web Service Composition Methods: the Need for Including Multi-Actor Elements" *The Electronic Journal of e-Government* Volume 5 Issue 2, pp 153 - 164, available online at www.ejeg.com

e-government as organizations cooperate more and more in public service networks (Provan & Milward, 1995). This paper is structured as follows. In the next section we analyze the composition problem using an example of the police force. After that, we present a literature review of existing composition methods. In the following section we present an approach for evaluating composition methods. We illustrate the evaluating approach using a fictive, but realistic example in section four and finally we draw conclusions.

2. Background

The SOA paradigm focuses on building information systems by discovering, matching and integrating pre-developed services (Linthicum, 2004). The basic idea of SOAs is to decompose a system into parts that are made accessible by services, to design these services individually and to construct new systems using these single services. SOA is a way of reorganizing a portfolio of previously siloed software applications and support infrastructure into an interconnected set of services, each accessible through standard interfaces and messaging protocols (Papazoglou & Georgakopoulos 2003). Central to the design of new services is the concept of a service composition.

A service composition combines services following a certain composition pattern to achieve a business goal, solve a scientific problem, or provide new service functions in general (Curbera, Khalaf, Mukhi, Tai, & Weerawarana, 2003). Service compositions may themselves become services, making composition a recursive operation. During an online order process for example, a composition can exist of a service that checks a customer's zip code as well as the delivery time of the goods ordered.

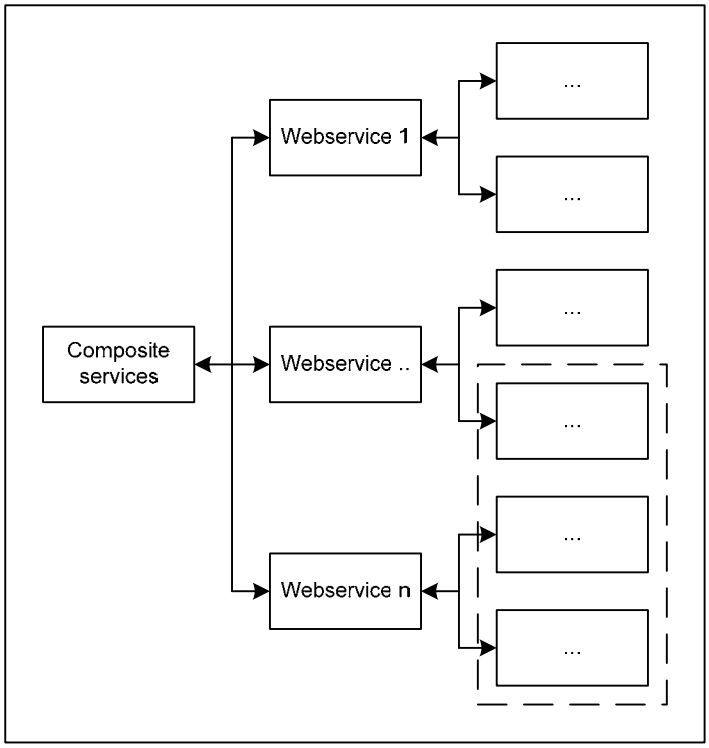

Figure 1: Design of a composite service.

The basic SOA defines an interaction between involved parties, as an exchange of messages between service requesters (clients) and service providers.

Providers are responsible for publishing a description of the service(s) they provide. Clients must able to find the description(s) of the services they require and must be able to bind to them. In general the process consists of five steps:

1. The provider publishes a service in a UDDI
2. The requester searches the UDDI for a specific service
3. Descriptions of candidate services are returned
4. The requester decides which service to use and invokes this service at the provider
5. The service is executed and the result is sent back to the requester.

In reality the UDDI search is not executed on each service invocation. The UDDI is only searched when the composition is designed or updated. The ideal of creating on-the-fly compositions is not used in reality today. The use of each other services is often dependent on long-term agreements among organizations.

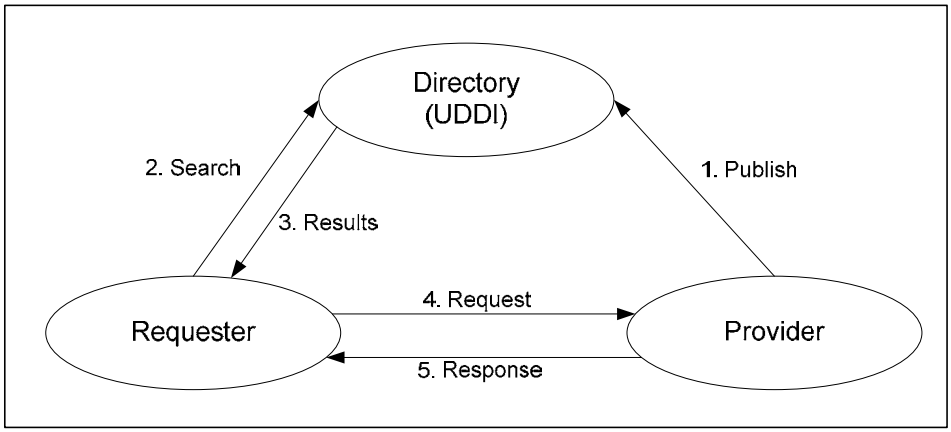

Figure 2: Service discovery, selection and enactment.

In this way fully integrated enterprises are being replaced by business networks in which each participant provides the others with specialized services (Curbera, Khalaf, Mukhi, Tai, & Weerawarana, 2003).

There are different approaches to create a service composition. These approaches enable different organizational units to interconnect their applications and to share data in distributed environment by using a combination of web services. These approaches include BPEL4WS, OWL-S, Web-components, π-calculus, Petri nets and Model checking/Finite state machines.

BPEL4WS is an abbreviation of Business Process Execution Language for Web Services. It is often shortened to 'BPEL'. It is an XML based orchestration language, standardized by the Organization for the Advancement of Structured Information Standards (OASIS). BPEL supports a process-oriented form of service composition (Curbera, Khalaf, Mukhi, Tai, & Weerawarana, 2003). Process orientation denotes that each BPEL composition is a business process or workflow that interacts with a set of Web services to achieve a certain goal (Curbera, Khalaf, Mukhi, Tai, & Weerawarana, 2003).

A BPEL composition interacts with a set of web services to achieve a given task (Milanovic & Malek, 2004). In BPEL the composition result is called a process, participating services are partners, and message exchange or intermediate result transformation is called an activity (Milanovic & Malek, 2004).

OWL-S is a part of the Semantic Web vision (Milanovic & Malek, 2004). OWL-S is an ontology built on top of Web Ontology Language (OWL) by the DARPA DAML program. It replaces the former DAML-S ontology (http://www.w3.org/Submission/OWL-S/) OWL-S is a services ontology that enables automatic service discovery, invocation, composition, interoperation and execution monitoring (Milanovic & Malek, 2004). Using OWL-S the composition process can be executed without human involvement.

This approach treats services as components, a web component packages together elementary or complex services and presents their interfaces and operations in a consistent and uniform manner in the form of a class definition (Yang & Papazoglou, 2002). The main idea is to encapsulate composite-logic information inside a class definition, which represents a web component (Milanovic & Malek, 2004). A web component's public interface can then be published and used for discovery and reuse (Milanovic & Malek, 2004). Web components support basic software development principles as reuse, specialization and extention (Yang & Papazoglou, 2002).

π-calculus is an algebraic method to describe processes, π-calculus is described by Milner (Milner, 1991). In π-calculus the basic entity is a process, types of process are an empty process, a choice between several I/O processes and their continuations, a parallel composition, a recursive definition or a recursive invocation (Milanovic & Malek, 2004). IO operations can be input (receive) or output (send). When compositions are described using pi-calculus it is possible to reason about composition correctness (Meredith & Bjorg, 2003).

A Petri net is a directed, connected and bipartite graph in which nodes represent places and transitions and tokens occupy places (Milanovic & Malek, 2004). When there is at least one token in every place connected

to a transition, that transition is enabled (when modeling web services this would mean it is possible to invoke a web service because all input requirements are met). An enabled transition might fire by removing one token from every input place, and depositing one token in each output place (Milanovic & Malek, 2004).

Model checking is used to formally verify finite-state concurrent systems. In case of model checking, the system specifications are described using temporal logic, then the model is used (traversed forward and back) to determine whether the specification holds (Maximilien & Singh, 2004).

A composition approach is the core part of a composition method. Several evaluations of compositions approaches can be found in literature (Beek, Bucchiarone, & Gnesi, 2006; Milanovic & Malek, 2004). Milanovic and Malek compare these approaches using the criteria: 1. Service connectivity, 2. Non-functional properties, 3. Composition correctness, 4. Automatic composition and 5. Composition scalability. Beek et al. (2006) use the following criteria: 1. Connectivity and Non-functional Properties, 2. Composition Correctness, 3. Automatic Composition, 4. Composition Scalability, 5. Exception Handling and Compensations and 6. Tool Support. The criteria used by both evaluations are similar, Beek et al. extend the set of criteria used by Milanovic and Malek by adding exception handling and compensations and by including tool support.

Table 1: Comparing service composition approaches (based on Milanovic & Malek, 2004).

	Service connectivity	Non-functional properties	Composition correctness	Automatic composition	*Composition scalability*
BPEL	√				*Average*
OWL-S	√	√			*Average*
Web components	√		√		*Low*
π-calculus	√		√		*Good*
Petri nets	√		√		*Low*
Model checking/FSM	√		√	√	*N/A*

All composition approaches offer connectivity, without connectivity if would be impossible to integrate a set of services. Connectivity is an absolute minimal requirement for a composition approach.

The ability to handle non-functional properties (or Quality of Service attributes) of services is desirable, for example to assess the speed or error probability of a proposed composition. Most composition approaches neglect specification of non-functional Quality of Service (QoS) properties such as security, dependability, costs, response time, reliability and scalability. Only OWL-S lets users define non-functional properties, but that capability has yet to be fully specified (Milanovic & Malek, 2004).

When the correctness of a composition is verified, the service and composition specifications are used to assess whether the composition will behave as required under different circumstances. The more formal approaches (Web components, π-calculus, Petri nets and Model checking/FSM) all provide some support for verification tasks. BPEL and OWL-S do not provide any way to check on specification conformation .

Automated composition is a part of the semantic web vision. Some kind of intelligent composition engine processes the specifications of the services and the requirements of the composition to be designed. A composition is then generated without any human involvement.

The composition scalability column indicates whether the composition approach is suitable for larger compositions. When the number of services used increases and the complexity of the composition approach increases at a higher rate, the scalability of the method is low.

The approaches listed in table 1 assume that all services are well-defined and there are unambiguous selection requirements and there is a clear, undisputed, goal about the performance. In fact, these approaches can be better denoted using the term *representational formalism* than using the term composition approach.

A composition approach as representational formalisms is not sufficient in the field of e-government. None of the composition approaches support the orientation and negotiation phase. In short, the evaluation approaches focus on easy to measure criteria, whereas other requirements are equally or even more important. As such there is a need for an evaluation approach that is able to deal with this more difficult to measure requirements.

3. Case study

In this section we elaborate the composition problem using a case study to be able to develop a comprehensive evaluation method. First, we present a case study from the police force to analyze the composition problem in this section. We will use this case study to derive e-government related requirements on composition methods.

Dutch police officers make use of existing state of the art mobile computer equipment to check data regarding persons and vehicles. The current system consists of a number of regional, national and international data sources provided by different organizations like the Inland Revenue and the Regional police forces. These sources can be queried in a unified way. The results of the queries is made visible on a mobile terminal (PDA) using a regular cell phone and a Bluetooth connection. This system is currently available to all 26 Dutch police regions, on a subscription basis provided by the ISC. The ISC (ICT-Service Centrum Politie, Justitie en Veiligheid U.A.) is the ICT supplier for the entire law enforcement and security (OOV) chain in the Netherlands. ISC specializes in and focuses exclusively on effective, reliable and fitting ICT solutions for the police, the judiciary and their partner organizations in the law enforcement and security sector.

If a police officer wants to check data of a car and a driver, the composite service consists of a the following sub services; checking the license plate and owner information at the RDW (Rijksdienst voor het wegverkeer, the Dutch road traffic department), checking if the taxes have been paid at the Inland Revenue, checking if all fines have been paid in the police databases and validating the name and address information of the driver in a municipal database. Each municipality has its own database containing citizens information within their geographical boundaries. The services required for this example are listed in table 2.

Table 2: Actors and the services they offer.

Actors	Services
RDW	Check license plate Check owner
Inland Revenue	Check tax
Regional Police Force	Check fines
Municipality	Log on Check name and address Display results

Figure 3 displays an overview picture of the complete system. The different cooperating organizations are clearly visible. These organizations are highly autonomous, the feasibility of the composition will depend on the degree the different interests of these organizations are met while cooperating.

During the service composition process a composition is realized by making use of the already existing services provided by different service providers. During this processes, decisions has to be made concerning which alternative service offerings to include in the composition. Different parties have often different requirements, opinions and interests, this will influence their choices made during the service selection process. The involved parties will differ in their opinion about the optimal composition. A police officer on the street for example likes to receive an immediate response to a license plate query, because of this requirement the query execution times at the RDW have to be rather small. The RDW has to pay for the computer capacity to meet this requirement. The RDW has to balance this expense with other expenses for other projects. The final composition needs to get the support from all parties, otherwise it likely will be impossible to realize the composition.

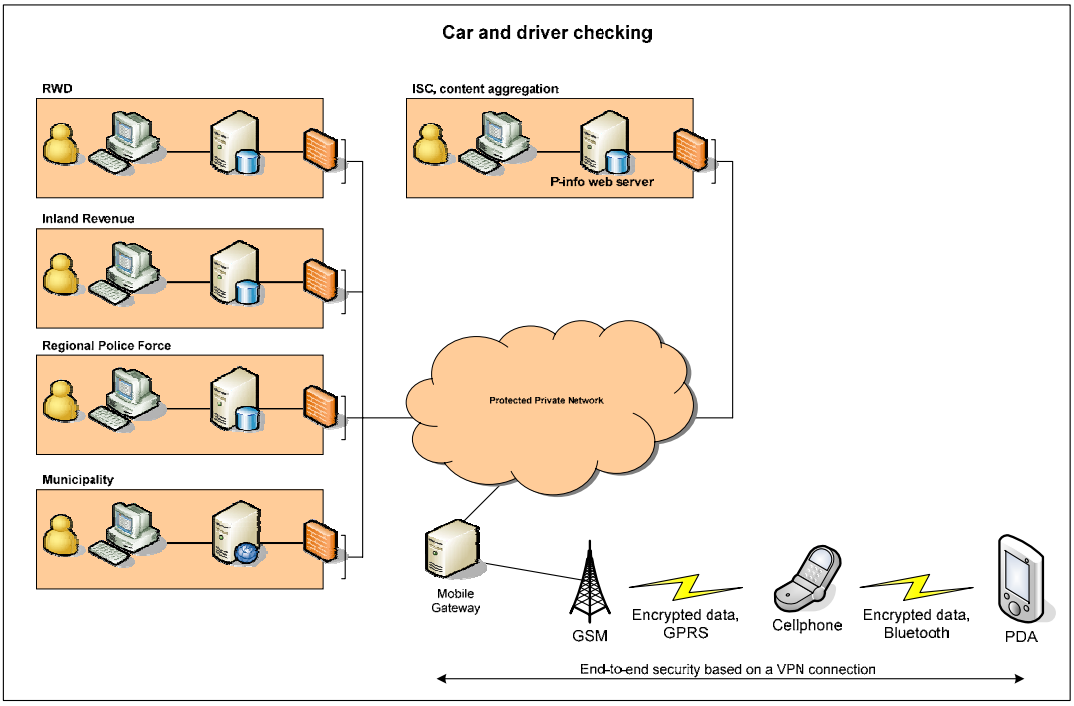

Figure 3: Elements of car and driver checking composition.

3.1 Requirements for composition methods

The essence of a service composition method is to support the process of connecting several services from different suppliers together in order to execute a specific task. During the composition process decisions have to be made which service of which supplier to include. We analyzed the composition process based on the problem structure and the views of actors and found that the process of composition design in essence exists of three stages. Each stage poses different requirements on compositions methods.

During the *orientation* phase parties will get the impression that cooperation is both feasible and beneficial (for example because of the possible reuse of existing cervices). At this stage, the cooperation is described only in general terms, there is often no overview available of services that might be used and the composition problem is not structured.

After the orientation phase, the *negotiation* phase follows where parties decide - in cooperation –which services are needed, who will deliver which services and what the specifications of these services will be. This is an essential phase as the unstructured problem is detailed to a clear problem and all the actors should have a common understanding of the problem.

At the moment agreement is reached regarding the services to be used, and regarding the specifications of these services, the composition can be realized. Subsequently the usage phase in which the composition is used starts. The usage phase ends at the moment when one of more participating actors decide to end their cooperation to the composition. Figure 4 shows schematically the three main phases of the composition process.

Reaching agreement regarding a composition is a crucial step in the process. It has to be clear for all parties what can be expected. In case a composition fails at a critical moment – for example in case of a disaster, when one depends on the correct functioning of the composition – the impact can be high and catastrophic. Furthermore, this might result in the asking of critical questions in Parliament and might even result in a political crisis.

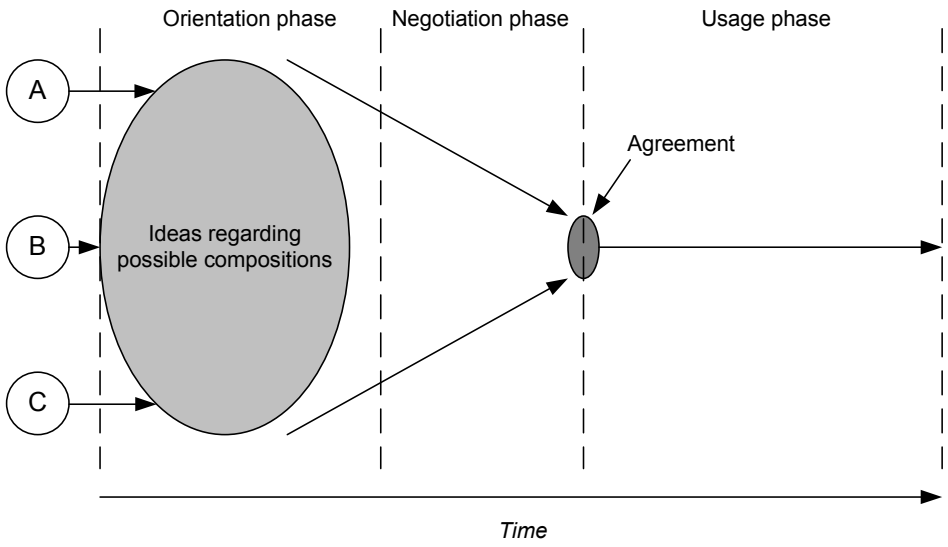

Figure 4: The stages of the composition process using services delivered by parties A, B and C.

During the composition process several technical-oriented problems might occur, for example one might have to deal with the integration of legacy systems, connecting these systems can be difficult when suitable interfaces are unavailable. It can be unclear which (web/software) services an organization can offer and what the properties of these services are. A lack of standardization, or indistinctness regarding dependencies, can complicate the design of a composition to a large extent. The agreement phase might not be reached in that case.

In e-government multiple parties/actors with possible conflicting interest try to cooperate which might result in actor-oriented problems. During this process the consequences of decisions are not directly clear, various problems regarding the composition may arise and the risks linked to the failure of the composition (not conforming to the specifications) are substantial. As a result, the characteristics of the composition problems lead to an important set of requirements for the composition method. These requirements can be seen as criteria when evaluating a set of composition methods. The multi-actor type of problems seem to be hardly considered by existing composition methods, although these type of problems determine to a large extent success and failure. In the next subsections we discuss four categories of composition problems. These problems will be used to derive criteria for evaluating composition methods.

3.1.1 Multi-actor perspective

The decisions regarding the composition of a service are taken within a network containing different actors. All stakeholders (commercial or non commercial, single departments or large organizations) have a certain unique view to the service composition problem. For example a police officer will focus on user friendliness of the interface used and on the battery life of the devices, they have to use the designed system on the streets. The application administrator will focus on scalability, and a line manager will be concerned about the costs and reliability of a composition.

The actors have often different interests and are dependent on each other. Because of these dependencies no single actor can solve the problem autonomously. They are dependent on the others to create a solution. Furthermore, none of the actors can impose a particular solution to the others. Often a solution is a compromise that has to combine several contradictory goals and interests.

The actors in the network will need to cooperate to a certain extent in order to realize a solution that at the same time helps realizing the common goal and pays enough attention to their specific interests. The decision making process is effective only if it leads to commonly taken decisions (De Bruin & Ten Heuvelhof, 1998). The decision making process that leads to a service composition will therefore also need to deliver a set of rules to which every actor involved will comply.

3.1.2 Non-functional requirements (express, evaluate)

When designing a composition, at first it is important to specify the requirements of this proposed composition (Dym, Agogino, Eris, Frey, & Leifer, 2005). A common divisions of requirement types is by making a difference between functional and non-functional requirements. The functional requirements all relate to the purpose of the composition, to the tasks it has to accomplish. The non-functional requirements relate to the QoS aspects. This part of the requirements specifies for example the overall reliability, availability or execution speed of a composition.

In order to express requirements, some level of agreement on what the objectives of the composition are must exist among the participating actors. Mostly the composition will be designed to assist a human user during a specific task. During this task the user interacts with the service composition in order to solve a problem (for example in order to check a license plate of a car).

In our case study the tasks a composition has to perform were not entirely clear at the beginning of the composition process. The process starts with a global idea and the purpose of the new service was not pre-defined and was not defined during the process. Moreover, a completely different composition might result from the process than initially expected. For example, during the process ICT-architects were involved that could draw the attention to a service that was not known in advance. Then a number of compositions have to be made, and only during tests in practice and interviews and observations in real life, the successful and unsuccessful compositions are identified. This requires understanding that the service composition process is an iterative and interactive process, in which the composition is more and more refined or changed according to current insights.

3.1.3 Insight into alternatives

A service composition consists of a number of services. If one service fails this will likely have an impact on the overall performance of the composition. If no alternative service is available as failsafe option, the performance will drop inevitably. It is even possible a composition fails entirely; in that case no response is given after the composition is executed. To avoid this alternative compositions should be available, which can be used in case of failure. As such, a composition method should be evaluated on the ability to include guidelines for its use. A composition method has to support failure analysis, or ‘what if ‘scenario’s.

3.1.4 Planning support (shared view)

Lack of a shared view among actors hinders both communications and might block the iterative development cycles of the composition process. As the composition problem is not clear at once our interviewees indicated the a number of interactions are necessary to refine the composition problem. A lack of iterations will result in identifying only a limited number of alternatives. It will also limit the number of alternative compositions that can be evaluated in a certain time, simply because the composition design process becomes a tedious process.

The composition method has to provide the possibility to identify services that have to be changed and services that are not available yet (without implementation). The composition method should provide the network partners to plan the realization of the composition.

The objectives are related to both the composition process (the steps to be performed by the stakeholders) as to the resulting composition. The process should facilitate reuse, dependability and planning. The result should reuse services as much as possible, contain an overview of functional and non-functional specifications and be accompanied by a planning.

4. Towards an approach for evaluating composition methods

In our case study, we found that composition is largely driven by human decision-makers who are using or following a certain method. Moreover none of the existing evaluation approaches focus on the evaluation from a multi-actor perspective. Consequently, we develop an evaluation method which takes into account different actors in this section. This method evaluates the compositions methods using a workshop setting, as in a workshop participants from various organizations can be included. In this way the multi-actor aspects can be evaluated. As such our methods goes beyond the hard evaluation of the composition method as done by the evaluation method discussed in the preceding subsection. This section discusses the design of the service composition workshop, the way of testing and the aspects the test is focused on.

4.1 Overview of the evaluation approach

To solve the described problems related to the design of service compositions an evaluation approach has been designed that specially addresses the fact that multiple parties are designing a composition together. Our case study shows that a multi-actor composition method is partly a negotiation process. To evaluate composition methods, our idea is to compare not only the methods with each other, but also with the current (ad-hoc) way of working.

To assess the impact of the newly designed multi-actor method, different methods have to be used to solve exactly the same case. In this way the conditions remain the same and the difference in outcomes can be contributed to the different composition methods. A workshop approach is chosen because it enables the comparison of different composition approaches within a controlled environment and without requiring log term efforts from busy police professionals.

A number of workshop participants designs the compositions for several scenario's within a relatively short time (about 4 hours). The scenarios consist of an objective (a description of the composition that has to be realized), an overview of the supplied services and an overview of the actor specific demands. The workshop resembles the real situation where multiple parties (actors) have to decide on a composition by means of negotiation. The workshop requires only a short amount of time and offers the possibility to vary both the composition method used and the available ICT support (the software) for the method. The test focuses on three main aspects at different moments in time as shown in Figure :

V1: Test to what extent the earlier identified hurdles are solved by using a specific method and tool
V2: Make a comparison of the results of the multi-actor composition method to the existing semantic method, what differences occur regarding the solving of the hurdles?
V3: Make a comparison of the composition methods used during the workshops and the current (ad-hoc) method.

Two types of workshops will be held, using exactly the *same* scenario as input to ensure that the methods are evaluated using the same circumstances. To test the aspects listed above, measurements will be taken on four different moments.

At moment **M1** the current way of working will be described. Which process is currently used to design compositions? Which problems arise during this process? The current way of working will be analyzed by conducting a number of interviews with employees of the police forces.

M2: At the start of the workshop participants are asked (by means of a survey) whether they expect several hurdles to occur. This survey also checks if the participants understood the scenario. All scenarios have to contain several clear hurdles.

The hurdles are clearly stated in the scenario descriptions for the workshop participants. During the workshops the comparison of two composition methods is the main objective, not the validation of the hurdles.

M3. At the end of the workshop the opinion of the participants regarding the method and the provided support is asked. To check to what extent the hurdles are solved by the method chosen.
Measuring the opinion of the participants, regarding the multi-actor composition method at the start and the end of the workshop, is insufficient to be able to draw conclusions regarding the effects of the composition method. Comparison material must be available.

For this reason compositions are made for exactly the same scenario using the semantic method in a separate workshop. Exactly the same surveys are used in both workshops. At this point is becomes possible to compare the results of the surveys in order to draw conclusions regarding the effects of the multi-actor composition method.

M4. Moreover it is also possible to discuss the results of the workshops (the compositions made) with a panel of experts. In this way a comparison can be made between compositions realized with different methods.

The graph below shows the testing process. Moments M1, M2, M3 and M4 are the moments of measurement on which an inquiry or interview is carried out.

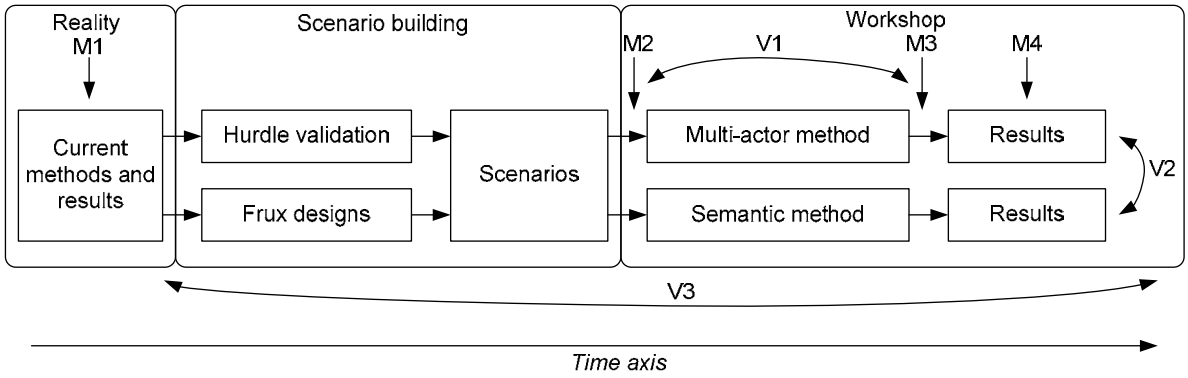

Figure 5: The composition evaluation approach.

The measurements are subsequently used to answer the questions V1, V2 and V3. The measurements M2 and M3 are used to determine whether the identified hurdles are solved by the composition method used (V1). The compositions made during the workshops are compared to analyze any differences in the results obtained using different methods (V2). By comparing the results of the interviews to the results of the composition workshops, it is determined whether the hurdles discovered in other projects also occur at the police, and what the differences are in the obtained results (V3). The hurdles define the four main evaluation criteria, these criteria are used to compare the different composition methods:

- Does the method provide support for the multi-actor perspective?
- Does the method provide the ability to express and evaluate non-functional requirements? Does the method still support the composition process when service properties are unknown?
- Does the method provide insight into alternative compositions, for example in the case a part of a composition fails?
- Does the method support the planning or the creation of a shared view on the composition? For example does it provide an overview of missing services, or services that have to be changed?

Next to the quantitative comparison of the method, also the behavior of the workshop participants will be taken into account. The conversation and interaction of participants will be reviewed. The evaluation of the composition methods consist of both a qualitative and a quantitative part.

4.2 The facilitator role

The composition workshops are sessions with a large agenda / many agenda steps. A group of participants have to design composition for a number of scenarios. During these tasks the participants are supported by a Group Support System and by a specially developed composition tool. The facilitation of this workshop is a complex task, to avoid unwanted influences to the group caused by the facilitator; it is advisable to use an external facilitator. Because of the complexity of the supportive tools and the scenario's, it is however rather complicated to transfer all knowledge within a reasonable amount of time to an external person. Consequently, one researcher will facilitate all workshops during this research project.

5. Conclusion

Current composition evaluating methods focus on the service usage phase and assume that all web services are well-described using standardized interfaces and have similar kind of performance and quality. Some of the approaches take into account the performance differential between web services, however only to a minor extent. All of them ignore factors like trust in web service and service provider, scalability and the not-invented-here syndrome. These aspects can be viewed as essential requirements that should be considered in the orientation and negotiation phases. Not considering them will easily result in failure.

In our literature review, it was found that current composition approaches neglect QoS attributes, assume all services already exist, do not provide support for multi-actor evaluation, do not provide planning or overview, do not take into account that actors may have different interests, assume a single rationality, provide little room for experiments, propose a single step to implementation, neglect dependencies / do not provide sufficient methods to track dependencies. In short, a mismatch exists between the characteristics of the problem and the current methods used to try to solve it as found in the literature. Furthermore, the methods are supply-driven instead of user-driven.

In order to evaluate composition methods within a multi-actor setting, a novel approach based on human participation in workshops, and pair-wise comparison of methods has been developed and presented in this paper. In these workshops the participants are asked to take on different roles and create a composition based on a set of web services. At the start of the workshop, participants are asked using a survey whether they expect several hurdles to occur. At the end of the workshop the participants are asked to indicate to what extent the hurdles can be solved by the method chosen. Finally, the results of the workshops, i.e. the created compositions, are evaluated by experts.

References

Beek, M. H. t., Bucchiarone, A., & Gnesi, S. (2006). A Survey on Service Composition Approaches: From Industrial Standards to Formal Methods: Technical Report 2006-TR-15, ISTI, Consiglio Nazionale delle Ricerche.

Beynon-Davies, P., & Williams, M. D. (2003). Evaluating electronic local government in the UK. Journal of Information Technology, 18(2), 137 - 149.

Curbera, F., Khalaf, R., Mukhi, N., Tai, S., & Weerawarana, S. (2003). The next step in Web Services. Communications of the ACM, 46(10), 29-34.

De Bruin, J. A., Ten Heuvelhof, E. F. (1998). Procesmanagement, over procesontwerp en besluitvorming. Amsterdam: Academic Service.

Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., & Leifer, L. J. (2005). Engineering Design, Thinking, Teaching and Learning. Journal of Engineering Education, 94(1), 103-120.

Feenstra, R. W., Janssen, M. F. W. H. A., & Wagenaar, R. W. (2006). Designing a Service Composition Framework to Support Multi Actor Networks. Paper presented at the EURO mGOV 2006, Brighton, UK.

Fremantle, P., Weerawarana, S., & Khalaf, R. (2002). Enterprise services. Examine the emerging files of web services and how it is intergraded into existing enterprise infrastructures. Communications of the ACM, 45(20), 77 - 82.

Hazlett, S. A., & Hill, F. (2003). E-government: the realities of using IT to transform the public sector. Managing Service Quality, 13(6), 445 - 452.

Linthicum, D. S. (2004). Next Generation Application Integration Addison Wesley.

Maximilien, E. M., & Singh, M. P. (2004). A Framework and Ontology for Dynamic Web Services Selection. IEEE Internet Computing, 8(5), 84 - 93.

Meredith, L. G., & Bjorg, S. (2003). Contracts and types. Communications of the ACM, 46(10), 41-47.

Milanovic, N., & Malek, M. (2004). Current solutions for Web service composition. IEEE Internet Computing, 8(6), 51 - 59

Milner, R. (1991). The polyadic-calculus: a tutorial: University of Edinburgh.

Papazoglou, M. P., & Georgakopoulos , D. (2003). Service-oriented computing: Introduction Communications of the ACM, 46(10), 24-28.

Provan, K. G., & Milward, H. B. (1995). A preliminary theory of network effectiveness: A comparative study of four community mental health systems. Administrative Science Quarterly, 40, 1-33.

Yang, J., & Papazoglou, M. P. (2002). Web Components: A Substrate for Web Service Reuse and Composition. Paper presented at the Proc. 14th Conf. Advanced Information Systems Eng. (CAiSE 02).

