

Towards Building Large Scale Multimedia Systems and Applications: Challenges and Status

Klara Nahrstedt
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
klara@cs.uiuc.edu

Wolf-Tilo Balke
University of Hannover and L3S Research Center
Hannover, Germany
balke@l3s.de

ABSTRACT

This paper is a position statement of the co-chairs for the First Workshop on Multimedia Service Composition on specific challenges in the area of multimedia service composition. The goal is to present and discuss problems that occur when considering building large scale multimedia systems via service composition. Today the realization of multimedia systems still heavily relies on building monolithic systems. Hence, building complex large scale multimedia systems is always a difficult, costly, time-consuming and challenging problem. Service-based architectures and the possibility to flexibly compose basic services to implement more complex workflows (or rather execution flows), as proposed in the Web and Grid communities, can provide a possible solution to this problem. However, due to the special characteristics of multimedia applications and the rich semantic structure of multimedia data and workflows, Web or Grid-based research results still cannot be readily applied. In this introduction-paper, we summarize challenges that need to be addressed and present a snapshot of the current state of the art towards building large scale multimedia systems.

Categories and Subject Descriptors

D.2.12 [Software Engineering]: Interoperability – *Interface definition languages*.

General Terms

Algorithms, Languages, Design.

Keywords

multimedia service composition, service-oriented architectures.

1. INTRODUCTION

Being a Brave New Topic at ACM Multimedia 2004 Conference [1], the topic of multimedia service composition has sparked considerable attention within the multimedia community. Composing basic building blocks in service-oriented architectures promise to introduce a maximum of flexibility and reusability of components into building even advanced multimedia applications. The first workshop on multimedia service composition in conjunction with ACM Multimedia 2005 provides a platform to investigate the necessary concepts in more detail and points to some related re-

search, composition frameworks and prototypical multimedia applications. This paper is intended as a positional statement of the workshop co-chairs on the specific challenges that will have to be addressed by the multimedia community and for that purpose showcases some relevant related work.

Service-oriented computing and service-oriented architectures are concepts strongly discussed and researched in the Web and Grid communities today. With the advent of frameworks and languages to build and manage Web services and protocols to enable conversations between them, a lot of work (mostly driven by industry alliances) has been invested in standardization. Generally speaking Web applications can already now be flexibly modeled using services as basic building blocks and the market – especially in B2B interactions – is constantly growing. Beside the efficient provisioning and improved reusability of components, the move from data-driven to service-driven Web architectures promises to open up a whole new field of value-adding applications. These applications can be built on top of existing components and thus reuse individual services to form new and increasingly complex workflows in a time- and cost-aware manner. Moreover, new innovative business models for content-, service- and network-providers can be employed and used for mutual benefit.

Given the enormous development costs for large scale applications also the multimedia community is currently on the move from monolithic multimedia applications to more flexible solutions. Extensive solutions are in the domain of data semantics. The multimedia community already provides sophisticated standards for media coding accompanied with meta-data descriptions (e.g. MPEG-7, MPEG-21). Nevertheless, useful concepts from Web services research on dynamically building complex applications and execution flows using semantically well-defined descriptions did not make a broad impact on multimedia systems development yet. On the other hand, Web-based models, concepts and constructs are invariant to new data types that are being heavily explored in the multimedia community. Therefore, the Web community tries to solve a much more general problem domain leading to a lot of problems that could possibly be avoided, if the problem space is limited down to a concrete domain. Therefore, the benefit of bringing together novel Web-based service-oriented concepts and the sophisticated handling and processing of multimedia data and annotations will be mutual.

In this introduction-paper we will outline some of the challenges for bringing service-oriented concepts into the multimedia domain and the status we see in this integration. In Section 2 we briefly present the multimedia application model and requirements on large scale multimedia systems. In Section 3 we discuss the system challenges and status, and Section 4 presents the semantic data challenges and provides an overview of the current state of the art. We conclude in Section 5 with a discussion of future directions in this area.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MSC'05, November 11, 2005, Singapore.

Copyright 2005 ACM 1-59593-245-3/05/0011...\$5.00.

2. COMPOSITION REQUIREMENTS

Multimedia service composition is a **composition process**, where multiple services (e.g., retrieval, transcoding, display services), processing multimedia data,

- are connected via functional and data dependencies to create a new multimedia service (e.g., a video-on-demand service), and
- span over heterogeneous network and distributed system infrastructures.

To pose clear requirements on the composition process, first we need to have a well defined multimedia service model, which then will provide the atomic functional unit in the overall composition process.

2.1 Multimedia Service Model

Multimedia applications are generally flow-based applications, since their data usually are continuous streams (e.g., video and audio streams), i.e. dependent in time and space. This data-time and space dependency puts stringent timing and spatial constraints on the functional services that assist in processing and communication of the multimedia data in distributed environments. Moreover, quality constraints often need to be taken into account, adding another dimension. Hence, due to the rich semantic relations of multimedia data, and their time and space dependencies, functional services end up with rich dependencies, and it makes the building of large scale multimedia applications and systems truly challenging.

In summary, a **multimedia service** is a functional entity that assists processing and communication of multimedia data in timely and space-aware fashion. Each service includes the concept of time, space and dependency relation to other services that precede or follow the application service. The time, space, data and functional dependency relations among individual multimedia services form a **service graph**, which yields new multimedia services. To compose independently developed services, each service needs have a clear description of its timing, spatial, semantic data and functional capabilities. This service description is expressed via **meta-data** and published in order for other services to be discovered and used.

Multimedia services within the service graphs can be divided into **input, output and intermediate/transformational** services. An example of an input service is a video capturing service that captures video data from a camera and prepares the data in digital form (e.g., at 30 frames per second, with frame size of 640x480 pixels, 8 bits per pixel) for further processing and communication. An example of an output service is a display service that takes the video and displays its bitmap on the hardware display. An example of an intermediate/transformational service is a transcoding service that takes MPEG-4 encoded video and transforms it into H.263 coded video.

Service descriptions, expressed via metadata, can be categorized into **media-specific and functional descriptions**. Media-specific metadata describe multimedia data characteristics and their related Quality of Service (QoS) specifications such as frame rate, frame size, jitter, end-to-end delay, throughput, loss rate. Functional metadata describe functions embedded in services such as encoding transformation, retransmission, or filtering functions.

2.2 Requirements on System Infrastructure

To build a large scale distributed multimedia application, the underlying system infrastructure must provide a strong support across multiple protocol and service layers for the overall service composition process. The service composition process consists of four phases such as the **service synthesis, discovery, selection and execution**, and the requirements on system infrastructure manifest themselves as demands of each of the service composition phase onto the various processing and communication protocol and service layers:

- **Demands of Service Synthesis:** Large scale composed applications form abstract dependent service graphs, and the synthesis of these graphs may be created off-line via high-level programming tools. If this is the case, then the underlying system infrastructure needs to accommodate two types of mapping:
 - (1) If the placement of physical multimedia services is already given through proxy service providers (e.g., IBM administers its own service proxy network [9]), then the service request needs to trigger mapping between the abstract service graph and the physical service network [9, 18].
 - (2) If the placement of physical multimedia services is not given, i.e., services are stored in a central service repository (e.g., Gaia smart room uses a central service repository [44]), then the services in service graphs need to be requested, mapped and uploaded into the physical service network infrastructure.
- **Demands of Service Discovery:** In case a requested service is not available, discovery protocols of substitutable services, and eventually replication and/or customization of services may be needed. Furthermore, service discovery will require scalable content-addressable network [2], scalable lookup services [3], search for service paths [6], mapping of media-specific QoS requirements onto their own (e.g., transport packet-specific) system/network QoS representations, fitted towards system-based processing and communication services and data (e.g., connection setup service, flow control service, scheduling service) [10], and other protocols and services.
- **Demands of Service Selection:** In case multiple services of the same functional description are found, service selection is needed [6]. The selection needs to be then guided by the media-specific metadata and their corresponding system/network QoS metrics since they need to match across composed system services (e.g., rate, data format), and if they don't match, intermediate multimedia services (e.g., transcoding) need to be requested and invoked to make the end-to-end service composition holistic. Moreover, even between services with exactly the same capabilities a selection has to be performed considering e.g. statistical parameters like the services' expected availability.
- **Demands of Service Execution:** Timely multimedia service delivery can only be achieved, if the underlying systems and networks support resource management mechanisms, protocols and policies for performance-related Quality of Service (QoS) metrics such as deadlines, throughput, jitter, loss rate, and other time- and space-related metrics. These QoS metrics are part of the media-specific meta-data descriptions (e.g., end-to-end video delay, video jitter).

2.3 Requirements on Semantic Data

Due to the rich semantics of multimedia (e.g., MPEG-7 and MPEG-21 standards introduced a large set of metadata to allow for content-rich query in multimedia databases), large scale multimedia applications will end up with large amount of multimedia streams of different qualities and characteristics, hence with a rich set of metadata. The media-specific metadata must satisfy the following requirements:

- The multimedia metadata needs to be expressed in an easily readable (and machine understandable) form, such that services can address it, program it and manipulate it.
- The multimedia metadata needs to be organized, so that easy management and efficient searches can be executed.
- The multimedia metadata needs to be compatible so that other services such as Web services can utilize it for its inclusion and processing.

3. SYSTEM INFRASTRUCTURE CHALLENGES AND STATUS

Large scale multimedia applications and their service composition process will run on heterogeneous network and system platforms, hence we split the system challenges into network challenges and system challenges.

3.1 Network Challenges

To support service composition, future networks will need to assist in service synthesis, discovery and selection via an appropriate **service path/graph establishment process**, and in service execution via **timely data delivery process**. Within this composition framework, we will discuss two major challenges dealing with quality of service:

Challenge 1: QoS Mechanisms for Service Composition

The biggest challenge to support establishment and execution of QoS-aware service graphs is the **inclusion of QoS-aware resource management mechanisms** into the **Internet protocol stack**¹. The QoS-aware service graph establishment needs QoS mechanisms through the whole protocol stack. For example, the MAC layer needs priority mechanisms or time division multiplexing (TDMA) approaches, the network layer would benefit from QoS-aware routing, the transport layer could use rate-based flow control, selective retransmission, service and data differentiation, and the session should have timing and adaptive coding service.

Furthermore, even if some QoS mechanisms exist, often they are not accessible to higher layers such as middleware and application services (cross-layer design). For example, TCP/IP protocol stack hides fully any QoS mechanisms in physical and MAC layers, hence many researchers conduct end-to-end QoS measurements to estimate possible network resource availability and implicitly availability for multimedia service graphs on top of these networks [19,20].

¹ It is important to stress that multimedia service composition uses Internet protocol stack across wired and wireless networks. The wireless networks, especially 802.11 networks, represent a very difficult infrastructure to support deterministic or statistical QoS guarantees for multimedia service delivery [37, 38].

The current status is that some QoS mechanisms, such as priority MAC scheduling, jitter control via queue management, exist at MAC and routing layers and are being utilized (e.g., CANS [12]), but many mechanisms and policies are not available in the lower network layers, e.g., QoS routing [17], or are not accessible to the higher protocol layers for its usage and control. Moreover, trade-offs within the QoS mechanisms have to be considered while managing multiple instances of multimedia data. There is a direct relation between the content flowing through a network and the available services managing that flow (especially in the presence of user-dependent trade-offs), as e.g. discussed in [31].

Challenge 2: QoS-aware Policies for Service Composition

The establishment and execution of service graphs deals with many **dynamic situations** since multimedia data changes its content over session time (e.g., during two hour movie viewing) and hence it changes its throughput, loss rate, and delay QoS characteristics. Handling this type of dynamic traffic requires QoS-aware adaptive policy management which is currently not present in the Internet protocol stack. The **QoS-aware adaptive policy management** must provide assistance in selection of physical service graphs, media types, new intermediate services, tradeoffs in case of resource shortage, and other assistance.

The current adaptive policy management frameworks are still mostly part of individual research projects. Most advances of the adaptive policy frameworks have been done in the area of **Wide Area overlays and Peer-to-Peer networks** [4, 5], where resources and services are being traded when finding service paths [6] and finding servers [7]. Adaptive and dynamic service composition frameworks have been explored in the CANS framework [12], OverQoS [13], SpiderNet [8, 14], and others. However, a lot of **tradeoff policy management** work is still missing in the wireless and pervasive environments, although some initial results are coming up [21].

3.2 System Challenges

The service composition operations (synthesis, discovery, selection and execution) also rely on system resources such as processors, memory, and disk that need to be appropriately allocated and coordinated in order to assist in timely service composition. We discuss three major challenges.

Challenge 1: Broad Availability of Multimedia OS

For multimedia services to perform according to their media-specific descriptions, each computing node should have multimedia operating systems that would monitor, allocate, schedule and manage local resources in a time and space-aware fashion. This means that to deliver multimedia streams in timely fashion, we need **deadline-based scheduling algorithms** at the processor and disk level. Furthermore, we need **time and space-based monitoring, prediction and management algorithms** to deal with the dynamic characteristics of multimedia streams that are processed and communicated at the various computing nodes.

The current status is that various multimedia scheduling algorithms for processors and disks have been explored (e.g., [39, 40, 41]), but none of them are part of current operating systems (e.g., Linux or Windows XP). The benefit analysis shows clearly that it would be of great performance advantage to have any of the researched soft-real-time scheduling algorithms, however, due to significant cost of embedding them into the general purpose OS and due to relatively small video traffic on our computing nodes, inclusion of deadline-based schedulers will have to wait.

Challenge 2: Automated Service Graph Establishment

One of the major difficulties with current composed multimedia services is that one has to *manually*

- (a) setup all physical service components in the distributed infrastructure or in a central repository,
- (b) provide a static service dependency path among physical services,
- (c) ensure that sufficient resources are available for composed service, and
- (d) invoke the appropriate physical service path for timely multimedia data delivery.

So the challenge is to automate the overall composed service graph establishment or at least part of this process. This means that there is a strong need to

- (1) provide **automated high-level programming tools** that would assist in creation and synthesis of abstract service graphs,
- (2) provide **automated service discovery and selection**,
- (3) provide **automated mapping and matching** between abstract service graphs and physical distributed service infrastructure, and
- (4) provide **automated QoS-aware service routing** and fault-tolerant invocation of service graphs.

The current state is that pieces of the establishment process have been automated. For example, there are limited programming tools that allow for abstract service synthesis, and creation of service graphs such as the QoS-Talk tool [42, 47]. There is also an extensive body of work on automated service discovery and selection including Chord P2P lookup service [3], media proxy finding service [6], QoS-aware discovery service [43, 16], and others. Assistance for service mobility, multicast, anycast and overall service composition can be obtained via the Internet Indirection Infrastructure (i3) framework [50, 51]. The automated mapping and QoS-aware service routing have been explored in SpiderNet [14], in service multicast framework [10, 18], and via the QoS Compiler framework [48, 49]. Fault-tolerant invocation of service has been explored in [15].

Challenge 3: Understanding and Dealing with Heterogeneous Devices

The large scale multimedia applications will run on very diverse devices which differ in processor power, memory capacity, network throughput, network connectivity, energy efficiency, distance accessibility, mobility, security, and other attributes. Many of these devices are connected via 802.11, Bluetooth, or 3G networks that differ in their range, MAC protocols, QoS support, and other characteristics. Many of these devices range from running a single service (e.g., sensors, iPAQs) to multiple services (e.g., laptops, PC servers). The integration of these different devices is not very well understood. Hence, the multimedia service composition challenges are

- (a) **scaleable algorithms** to manage a large number of devices (hundreds of sensors or mobile devices),
- (b) **dynamic addressing** of devices and content in case of mobility,

- (c) **fast hand-shake** in case of service discovery and selection,
- (d) **timely and scalable delivery**, and many others.

The current state is that a lot of research has been done in smart rooms and other ubiquitous environments where many small and mobile devices reside. However, little has been done in integrating the multimedia pervasive computing research into large scale wide area distributed infrastructures. Few examples show interesting results in some of the settings: scalable and mobile delivery in smart rooms was explored in the Gaia middleware system [44, 45], scalable content-addressable network is discussed in [2], and seamless hand-shake is presented in [46].

In summary, the research community explored some of these system challenges in simulated or controlled environments on community networks such as smart rooms or Planetlab, however, unless these research results are integrated with Web or Grid system services, which do have much broader usage due to large commercial or defense backups, multimedia service composition will have difficulties when building large scale systems.

4. SEMANTIC DATA CHALLENGES AND STATUS

The ultimate goal of Web services is to provide interoperability for a possibly large number of applications by providing a generic syntax and interface to service components. Standardized languages like SOAP and WSDL [27] for communication between services and the description of service interfaces are based on XML and also rely on XML for representing the data types involved. In this section we will consider multimedia service composition challenges from their multimedia semantic data and service description point of view.

4.1 Modeling Compositions

While for simple interactions or conversations with services the SOAP and WSDL standards already provide a good foundation, the problem of composition is somewhat harder. Compositions deal with the implementation of complex applications that are in turn offered as new composite services. The component services that are invoked in this application are generally different (atomic or composite) services usually offered by multiple providers. The sequence and conditions in which a Web service invokes other services to perform a certain task together is often referred to as orchestration. As we already discussed before, the basic problems in performing such compositions occur in different steps during the composition process:

- Service synthesis
- Service discovery
- Service selection
- Service execution

These service composition operations apply to multimedia service composition as well as discussed in Section 2.1, and Section 3. From the semantic modeling point of view they yield four distinct modeling challenges.

Challenge 1: Modeling of Service Synthesis

The first step in the service composition process is the service synthesis, which builds from basic and independent components the synthesis of a suitable invocation flow; a task very similar to

specifying an intended **workflow** describing the application. Though sometimes results from artificial intelligence research like **goal planning** (see e.g. [33]) might be applicable, for most applications the synthesis has still to be performed manually (e.g. by specifying sequence or activity diagrams of alternative invocation flows) or at least supervised semi-automatically. An example for such a specification of alternative invocation flows is the **composite services description language (CSDL)** used in the eFlow system [11]. Here a process schema for a composite process is modeled by a graph, which defines the order of execution among the nodes in the process. Composition graphs in eFlow can include service, decision, and event nodes, where service nodes represent invocations of services; decision nodes specify alternatives and rules controlling the execution flow, while event nodes are used to send and receive notifications with respect to other services. If different steps can be composed to a single service satisfying a subgoal, this subgoal can of course also be used by different composition schemes not necessarily having the same overall goal.

Challenge 2: Modeling of Service Discovery

After one or more correct invocation flows for the application have been determined, suitable services for composition have to be found during the service discovery. While it is a general problem to figure out what functionality a service generally provides (also state of the art standards like **UDDI** [28] only amount to simple keyword matching during discovery), there are more questions to address. For a running application it is essential to ensure that services involved will be able to interact properly, in other words that they are compatible with each other (see e.g. the discussion in [32]). Services can be incompatible for a variety of reasons. First, there is the general **semantic incompatibility** of the functionality (e.g. an encoder service obviously cannot perform scheduling). Here it is important to notice that services could also perform more specialized tasks only (and thus would need additional services in the composition), or might be able to even perform more complex tasks, whose functionality may not be fully needed, but does not hurt either. Since in restricted domains there might at least exist a common understanding or even a standardized classification of what will be expected from a specific service, this problem can sometimes be put aside.

Second, incompatibility might arise from **mismatches in interfaces** (as e.g. defined by **WSDL**) or the type of messages they can exchange. Usually also this problem can be quite easily checked and obvious mismatches could be avoided. More challenging is a mismatch in the dynamic behavior of services, e.g. possible deadlocks during execution given certain message exchanges. Formal representations of a services behavior as given by e.g. Petri nets or state machines, thus have to be reasoned about to guarantee a correct application (see e.g. [25]). Compatibility is also closely related to another problem in flexibly composing applications, **substitutability**. Substituting a previously used service by a new one is often necessary, for instance when a specific service is unavailable due to network or server problems.

Challenge 3: Modeling of Service Selection

Service selection mainly poses the problem of choosing adequate services that have been discovered in the previous step. There is a **difference between trade-offs** induced by limitations in the capabilities of the set of services to choose from (these trade-offs usually do already occur at discovery time and sometimes are handled cooperatively with respect to the user respect, e.g. [23]),

or differences in the functionality of individual services even if all services support the basic task. There may be differences in many characteristics like service costs, quality of service guarantees, or expected service availability. Although this decision for the individual service can often be made based on an individual **user's preferences**, see e.g. [24], or a **group profile** for a certain application incorporating rules like e.g. always choosing those services offering the best quality of service guarantees, the impact on the composition is hard to assess. Choosing specific services e.g. optimizing costs at some stage in the composition process can lead to problematic situations later in the invocation flow. The problem thus becomes a multi-objective problem that has to be solved before the instantiation of a specific composition can be offered. Solving this problem is, however, usually possible in acceptable time, because there will only be a limited number of discovered services and their possible characteristics. On the other hand, dynamically putting together compositions (e.g. if unavailable services have to be substituted) is often difficult to facilitate given restrictive quality of service constraints.

Challenge 4: Modeling of Service Execution

The main problem in the execution is usually in the controlling and monitoring of the application and its characteristics. Applications may be on a simple best effort basis where the failure of individual components might not amount to serious problems, but can also range to commercial services that will have stronger demands, for example due to putting penalties on quality of service violations. Thus also the execution models range from simple frameworks trying to substitute failed services as quickly as possible, to full-fledged transaction models, e.g. the XML-based model discussed in [29]. Defining an adequate set of control parameters, monitoring throughout the composite application (even if certain components should be replaced dynamically) and proactively managing undesired situations during service execution will thus demand a lot of attention. Current implementations like the business process execution language for Web services (**BPEL4WS**), see e.g. [35], do already allow for a limited amount of exception handling [34], but are still far from what is needed to control complex multimedia applications.

4.2 Meta-data for Compositions

As we have pointed out in the previous section, the production of viable compositions basically can be managed as a planning problem. Though some basic composition patterns could be designed manually, given the variety of different technical devices and thus different implementations of services will definitely need some automation. If Web service compositions have to be performed automatically, a general understanding of the terms involved (e.g. descriptions of service capabilities or the compatibility of certain data types) has to be shared. This sharing of common vocabularies and the benefit brought by machine-understandable meta-data has evolved into a large research area within the Web community, called the **Semantic Web**.

Semantic Web technologies focus on managing structured collections of information, often together with sets of inference rules that can be used for automated reasoning. The challenge is to provide a language that expresses both data and rules for reasoning about the data and that allows rules from any existing knowledge-representation system to be exported. Any composition engine that has to compare or combine information across two or more services to be combined, needs to know the exact meaning of terms used e.g. for description and if they refer to the same or at least similar concepts. A powerful solution to this problem is

given by **ontologies** providing structured collections of information and formal definitions of the relations among terms. Advanced types of ontologies provide a **taxonomy** and a set of **inference rules**. The taxonomy defines classes of objects and relations among them. For multimedia service composition we identify two major challenges that will need to be solved to make service descriptions easily expressible, organized and compatible.

Challenge 1: Multimedia Service Taxonomies

Suitable taxonomies are a basic problem that has to be solved independently for each application domain, although some basic concepts might be transferable. In multimedia applications there are already some first approaches towards building taxonomies, see e.g. [1]. However, it remains to be seen, if the sophisticated media-specific descriptions of multimedia data types as e.g. given by the MPEG-7 standard can provide a suitable taxonomies or how they have to be extended. Furthermore, also important concepts in multimedia applications like QoS parameters and their interdependence needs to be modelled. Ontologies thus can enhance the functioning of composition engines by improving the accuracy of service capabilities looking for precise concepts instead of ambiguous keywords. More advanced applications will use ontologies to relate the information on services or data types to the associated knowledge structures and inference rules.

Challenge 2: Semantic Ontology Multimedia Language

The development of an ontology for Web services using the **Semantic Web ontology language OWL** based on the DAML+OIL standards [26], has led to the creation of the Ontology Web Language for Services (OWL-S, formerly DAML-S). OWL-S [36] is a Web service ontology developed in OWL, a description logic-based language for describing content. OWL-S has well-defined semantics and can be used to describe the process model of a Web service. The challenge is if this type of description logic-based language could lead to a Semantic Ontology Multimedia Language to allow for multimedia content description and how it will mesh with the current MPEG-7 and MPEG-21 standards that do define media-specific metadata. Unlike descriptions in WSDL that provides no means to represent the semantics of defined operations and associates messages, OWL-S provides a language for specifying functional descriptions in the form of preconditions and effects of operations together with semantic types for both input and output values of the service. The definitions of all semantic concepts used (what for instance an effect is meant to be) then can be made available using a **uniform resource identifier** (URI) and can be shared (and more or less understood) by different services. Another important aspect is that in this way also the output (types) of a service can be correctly interpreted. OWL-S is an OWL ontology featuring three parts:

- a profile,
- a process model,
- and a grounding.

The profile refers to the service capabilities, whose description is needed prominently for discovering services that are capable of performing a requested task in compositions. How to match different descriptions of essentially similar or even identical capabilities, however, remains a largely unsolved problem. The process model provides an insight into how the service works and thus enables the invocation (and to a certain degree also monitoring and recovery) in actual compositions. The grounding finally maps constructs of the process model to detailed specifications of mes-

sage formats, protocols, etc. For a more detailed description of the profile, process model and grounding sub-ontologies see e.g. [36].

However, even considering such promising frameworks like OWL-S, the Semantic Web community is still a long way from the goal of automated Web services composition, and the same applies to automated multimedia service composition. Beside the **fundamental planning problem** (that has already been researched in AI for quite some time, too), the representation is still not rich enough to suffice for complex compositions in composite processes. It is also questionable, if the concepts of preconditions and effects are sufficient for deriving service guarantees like needed in most multimedia applications. On the other hand, especially in terms of the planning building multimedia applications seems generally not as hard a problem as composing all purposed business processes out of arbitrary Web services. Having strongly typed data, structured descriptions and quite often to some degree predefined workflows, it remains to be seen, if current multimedia standards can be used to augment service ontologies strong enough to tackle the composition problem for large scale real world applications.

5. SUMMARY

In this paper we have outlined service composition challenges that need to be solved for large scale multimedia applications to become reality. We have addressed system infrastructure challenges as well as semantic data challenges. From the current state of the art it follows that the multimedia community has already progressed far in terms of understanding the underlying system infrastructure. This includes topics like multimedia streams setup and delivery, time and space-aware (QoS) data specification, timely delivery services and protocols, as well as monitoring and management services that assist in handling of independent resources. That means the community can handle single services quite effectively and has sufficient means to control their execution.

However, we are still missing a lot of service-based models, frameworks and implementations that would provide timely dependency management during the four main steps of dynamic interaction with services for composition tasks: synthesis, discovery, selection, and execution of composed services, and many other capabilities especially in terms of controlling a composed execution flow (especially with respect to quality of service) and dynamically adapting to failures, e.g. the problem of timely substitution of failed services. Given that the challenges can be overcome service composition could become a broadly used concept and software engineering pattern in building even large scale multimedia applications.

The presented challenges clearly outline many future directions that service composition research needs to explore. We conclude with two further samples of future questions that may be of interest to the community:

- (a) Can services decouple and express QoS-aware adaptive policies for external management? If yes, how do we match different adaptive policies to enforce stable multimedia delivery? How do we coordinate different adaptive techniques (e.g., linear control, fuzzy control) in end-to-end composed services over heterogeneous devices and networks?
- (b) Can MPEG-7 be used to define an upper ontology for content retrieval purposes? What other upper ontologies are needed e.g. for QoS guarantees, service capabilities, capabilities of technical devices, etc.?

6. ACKNOWLEDGEMENTS

As co-chairs of the workshop on multimedia service composition (MSC'05) the authors would like to thank ACM SIGMM for sponsoring the workshop in conjunction with ACM Multimedia 2005. We are also grateful to all members of the program committee, Christian Becker, Hao-Hua Chu, Peter Dolog, Xiaohui Gu, Dejan Milojicic, Nalini Venkatasubramanian, Matthias Wagner, Klaus Wehrle, Xing Xie, Dongyan Xu, and the keynote speaker Roger Zimmermann who helped to make the workshop possible.

7. REFERENCES

- [1] K. Nahrstedt, W.-T. Balke. A Taxonomy for Multimedia Service Composition. ACM Multimedia Conference, New York, USA, ACM, 2004
- [2] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker: A scalable content-addressable network, ACM SIGCOMM 2001, San Diego, USA, 2001.
- [3] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, ACM SIGCOMM 2001, San Diego, USA, 2001.
- [4] Gnutella, <http://gnutella.wego.com/>
- [5] Napster, <http://www.napster.com/>
- [6] D. Xu, K. Nahrstedt: Finding Service Paths in a Media Service Proxy Network, SPIE/ACM Multimedia Computing and Networking Conference (MMCN'02), San Jose, USA, 2002.
- [7] R. Carter, M. Crovella: Server Selection using Dynamic Path Characterization in Wide-Area Networks, ACM Multimedia (Multimedia Middleware Workshop), Ottawa, Canada, 2001.
- [8] X. Gu, K. Nahrstedt, R. Chang, Ch. Ward: QoS-assured Service Composition in Managed Service Overlay Networks, IEEE ICDCS 2003, Providence, USA, 2003.
- [9] X. Gu, K. Nahrstedt, R. Chang, Z. Shae: An Overlay Based QoS-Aware Voice-Over-IP Conferencing System, IEEE Intern. Conf. on Multimedia and Expo (ICME2004), Taipei, Taiwan, 2004.
- [10] J. Jin, K. Nahrstedt: Large-Scale Service Overlay Networking with Distance-Based Clustering, ACM/IFIP/USENIX Middleware 2003, Rio de Janeiro, Brazil, 2003.
- [11] F. Casati, S. Ilnicki, L. Jin, V. Krishnamoorthy, M. Shan: Adaptive and Dynamic Service Composition in eFlow, Intern. Conf. on Advanced Information Systems Engineering CAiSE'00, LNCS 1789, Stockholm, Sweden, 2000.
- [12] X. Fu, W. Shi, A. Akkerman, V. Karamcheti: CANS: Composable, Adaptive Network Services Infrastructure, USENIX Symp. on Internet Technologies and Systems, San Francisco, USA, 2001.
- [13] L. Subramanian, I. Stoica, H. Balakrishnan, R. Katz: OverQoS: Offering QoS using Overlays", Workshop on Hot Topics in Networks (HotNets-I), Princeton, USA, 2002.
- [14] X. Gu, K. Nahrstedt: A Scalable QoS-aware Service Aggregation Model for P2P Computing Grids, IEEE HPDC 2002, Edinburgh, UK, 2002.
- [15] E. Cohen, S. Shenker: Replication Strategies in Unstructured Peer-to-Peer Networks, ACM SIGCOMM, Pittsburgh, USA 2002.
- [16] J. Guo, B. Li: Distributed Algorithms in Service overlay Networks: A Game Theoretic Perspective, IEEE ICC 2004, Paris, France, 2004.
- [17] Z. Wang, J. Crowcroft: QoS Routing for Supporting Resource Reservation, IEEE Journal on Selected Areas in Communications (JSAC), vol. 14(7), 1996.
- [18] J. Jin, K. Nahrstedt: Source-based QoS Service Routing in Distributed Service Networks, IEEE ICC 2004, Paris, France, 2004.
- [19] B. Melander, M. Bjorkman, P. Gunningberg: A new End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks, Global Internet Symposium, San Francisco, USA, 2000.
- [20] M. Jain, C. Dovrolis: End-to-End Available Bandwidth Measurement Methodology, Dynamics and Relation with TCP Throughput, ACM SIGCOMM, Pittsburgh, USA, 2002.
- [21] J. Smith, R. Mohan, C. Li: Scalable Multimedia Delivery for Pervasive Computing. ACM Multimedia, Orlando, USA, 1999.
- [22] T. Bultan, X. Fu, R. Hull, J. Su: Conversation Specification: A New Approach to Design and Analysis of E-Service Composition. WWW'03, Budapest, Hungary, 2003.
- [23] W. Balke, M. Wagner: Cooperative Discovery for User-centered Web Service Provisioning. ICWS'03, Las Vegas, USA, 2003.
- [24] W. Balke, M. Wagner: Towards Personalized Selection of Web Services. WWW'03, Budapest, Hungary, 2003.
- [25] S. Narayanan, S. McIlraith: Simulation, Verification and Automated Composition of Web Services. WWW'02, Honolulu, Hawaii, 2002.
- [26] D. Connolly et al.: DAML+OIL Reference Description. W3C Note, 2001.
- [27] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana: Web Services Description Language (WSDL) 1.1. <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>, 2001.
- [28] UDDI. The UDDI Technical White Paper. <http://www.uddi.org>.
- [29] P. Pires, M. Benevides and M. Mattoso. Building Reliable Web Services Compositions. WS-RSD'02, Erfurt, Germany, 2002.
- [30] M. Paolucci, N. Srinivasan, K. Sycara, T. Nishimura: Towards a Semantic Choreography of Web Services: From WSDL to DAML-S. ICWS'03, Las Vegas, USA, 2003.
- [31] W. Kellerer, M. Wagner, W. Balke, H. Schulzrinne. Preference-based Session Management for IP-based Mobile Multimedia Signaling. Europ. Trans. on Telecommunications, Vol. 15(4), Wiley, 2004.
- [32] L. Bordeaux, G. Salaun, D. Berardi and M. Mecella: When are two Web Services Compatible? VLDB-TES'04, Toronto, Canada, 2004.

- [33] M. Vilain. Getting Serious about Parsing Plans: a Grammatical Analysis of Plan Recognition. AAAI-90, Boston, USA, 1990.
- [34] F. Curbera, R. Khalaf, F. Leymann, S. Weerawarana: Exception Handling in the BPEL4WS Language. Business Process Management, Eindhoven, The Netherlands, 2003.
- [35] R. Khalaf, N. Mukhi, S. Weerawarana: Service-Oriented Composition in BPEL4WS. WWW'03, Budapest, Hungary 2003
- [36] D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, N. Srinivasan, K. Sycara: Bringing Semantics to Web Services: The OWL-S Approach. SWSWPC'04, San Diego, CA, USA 2004.
- [37] S. Shah, K. Chen, K. Nahrstedt: Dynamic Bandwidth Management in Single Hop Ad Hoc Wireless Networks, Kluwer MONET Journal, Vol. 10, No.1-2, Feb. 2005.
- [38] Y. Yang, R. Kravets: Distributed QoS Guarantees for Real-time traffic in Ad Hoc Networks, IEEE SECON, Santa Clara, CA, 2004.
- [39] Hao-hua Chu, Klara Nahrstedt: CPU Service Classes for Multimedia Applications, IEEE International Conference on Multimedia Computing and Systems (ICMCS'99), Florence, Italy, June 1999.
- [40] M. Jones, D. Rosu, M-C. Rosu: CPU Reservations and Time Constraints: Efficient, Predictable Scheduling of Independent Activities, ACM Symposium on Operating Systems Principles, Saint-Malo, France, October 1997.
- [41] W. Yuan, K. Nahrstedt, K. Kim: R-EDF: A Reservation-based EDF Scheduling Algorithm for Multiple Multimedia Task Classes, IEEE Real-Time Technology and Applications Symposium, Taiwan, 2001.
- [42] X. Gu, D. Wichadakul, K. Nahrstedt: Visual QoS Programming Environment for Ubiquitous Multimedia Services, IEEE International Conference on Multimedia and Expo (ICME), Tokyo, Japan, 2001.
- [43] D. Xu, K. Nahrstedt, D. Wichadakul: QoS-aware Discovery of Wide-Area Distributed Services, IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid), Brisbane, Australia, 2001.
- [44] M. Roman, Ch. Hess, R. Cerqueria, A. Ranganathan, R. Campbell, K. Nahrstedt: Gaia: The Middleware Infrastructure to Enable Active Spaces, IEEE Pervasive Computing Magazine, Oct-Dec, 2002.
- [45] S. Chetan, J. Al-Muhtadi, R. Campbell, D. Mickunas: MobileGaia: A Middleware for Ad-hoc Pervasive Computing, IEEE Consumer Communications and Networking Conference (CCNC'05), Las Vegas, NV, USA, 2005.
- [46] Yi Cui, K. Nahrstedt, D. Xu: Seamless User-level Hand-off in Ubiquitous Multimedia Service Delivery, Kluwer Multimedia Tools and Applications Journal, 2004.
- [47] D. Wichadakul, X. Gu, K. Nahrstedt: A Programming Framework for Quality-aware Ubiquitous Multimedia Applications, ACM Multimedia 2002, Juan Les Oins, France, 2002.
- [48] K. Nahrstedt, D. Wichadakul, D. Xu: Distributed QoS Compilation and Run Time Instantiation, IEEE/IFIP International Workshop on Quality of Service (IWQoS), Pittsburgh, PA, USA, 2000.
- [49] D. Wichadakul: Q-Compiler: Meta-data QoS-Aware Programming and Compilation Framework, PhD Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2003.
- [50] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, S. Surana: Internet Indirection Infrastructure, ACM SIGCOMM, Pittsburgh, PA, USA, 2002.
- [51] K. Lakshminarayanan, I. Stoica, K. Wehrle: Support for Service Composition in i3, ACM Multimedia (Position Paper), New York, NY, 2004.