



HAL
open science

A Multimedia Ontology Driven Architecture framework (MODA) for networked multimedia systems

Jorge Gomez, Myriam Lamolle, Ernesto Expósito

► **To cite this version:**

Jorge Gomez, Myriam Lamolle, Ernesto Expósito. A Multimedia Ontology Driven Architecture framework (MODA) for networked multimedia systems. 2009 First International Conference on Networked Digital Technologies (NDT), Jul 2009, Ostrava, Czech Republic. pp.411-416, 10.1109/NDT.2009.5272081 . hal-03580979

HAL Id: hal-03580979

<https://hal.science/hal-03580979v1>

Submitted on 18 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Multimedia Ontology Driven Architecture framework (MODA) for Networked Multimedia Systems

Jorge Gomez^{1,2,4}, Myriam Lamolle³, Ernesto Exposito^{1,2}

¹LAAS-CNRS, 7 av. du Colonel Roche, Toulouse FRANCE

²Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France

³LIASD/LINC, IUT de Montreuil, Université Paris VIII

⁴ Facultad de Matemáticas, Universidad Autónoma de Yucatán, México

{jorge.gomez, ernesto.exposito}@laas.fr

m.lamolle@iut.univ-paris8.fr

Abstract

Heterogeneous multimedia environments make difficult the automatic deployment and interoperability of networked multimedia systems. Diversity of languages, protocols, and hardware platforms lead to major incompatibility issues. Moreover, dynamic multimedia systems instantiation and configuration guided by user and community requirements and preferences are not easy to be performed in this context. This work introduces an ontology framework for automatic networked multimedia systems deployment and configuration. Based on well-known standards, this framework integrates multimedia, distribution and platform concepts. A study case illustrating how networked multimedia systems can be dynamically deployed is presented in order to demonstrate the advantages of using this framework.

1. Introduction

The accelerated development of Internet and the high diversity of networked devices available in the market have led to the fast development of networked multimedia applications (e.g. IPTV, VoIP, VoD, visio-conferencing, etc.). These applications are usually statically deployed. Moreover, users of these applications belong to very well defined groups or communities (e.g. users of an internet provider receiving IPTV or workmates of a project establishing a videoconference).

With the explosion of social networks popularity, new requirements for integrating heterogeneous software and hardware architectures as well as new interfaces allowing spontaneous exchange of multimedia information between ad-hoc and mobile communities of users have recently appeared.

These new requirements lead to two major challenges: interoperability and dynamicity. Firstly, current multimedia systems are highly heterogeneous. In fact, multimedia applications are implemented in various languages (e.g. Java, C/C++, .net), software environments (e.g. J2ME, Android, iPhone, JMF) and a diversity of protocols are used to

control, represent and exchange multimedia data. Likewise, network characteristics such as security, firewall, NAT, mobility and quality of service have an important impact in the time-constrained multimedia sessions and need to be taken into account in order to guarantee final users satisfaction. Secondly, configuration and deployment of distributed multimedia systems should be driven by the users' requirements in order to facilitate the spontaneous establishment of multimedia sessions allowing the exchange of information between the members of the social network.

Several research studies have been carried out in order to propose standard frameworks and platforms for designing and developing multimedia systems. Organizations such as ITU, IETF or W3C have promoted working groups leading multimedia standardization activities. Likewise, a large diversity of development platforms for fixed and mobile devices is being constantly released.

However, even when relevant efforts of standardization have been carried out, a generic multimedia framework aimed at integrating the basic distributed multimedia concepts has not been defined yet. Such generic multimedia framework should be able to represent the various configurations of distributed multimedia systems and should help to automatically configure and deploy new systems driven by the user and community requirements and preferences.

Our work targets the specification and development of the Multimedia Ontology Driven Architecture framework (MODA). This framework is based on well-known standards integrating multimedia, distribution and platform specific concepts. MODA also includes an automatic process based on the use of these ontologies allowing dynamic translation between user/community requirements and multimedia components deployment. Deployment specifications produced by the framework can also be used to self configure the communication system by deploying adequate communication components at middleware, transport or network layer, i.e. in order to provide the quality of service required to guarantee the multimedia session requirements.

This paper is organized as follows. Second section introduces driven oriented technologies and multimedia frameworks providing the basis of the MODA framework. Section 3 presents MODA framework, the basic ontologies and the driven oriented process for multimedia components deployment specification. Section 4 presents a case study aimed at illustrating how networked multimedia system can be dynamically deployed. Finally, section 5 presents the conclusions and perspectives.

2. State of the art

In this section the state of the art providing the basis of the MODA framework is introduced. The first subsection is aimed at introducing model driven methodologies intended to be integrated within the framework. The second subsection is intended to introduce the standards, protocols and implementations in the area of distributed multimedia systems.

2.1. Model Driven Engineering

The objective of a Model Driven Engineering (MDE) approach is to shift the complexity of the implementation of an application to the specification of this one. It is then a question of making an abstraction of the programming language using an abstract modeling process focused on the use of several standards such as MOF, OCL, UML and XMI.

Model Driven Architecture [1, 2] is a specific field of the MDE, which specifies three levels architecture:

- Representation of Computation Independent Model (CIM) from a business model. CIM describes the context in which the systems will be used.
- Representation of Platform Independent Model (PIM) from CIM. It describes the system itself without any details of its use or its platform. A PIM will be suited for one or several real architectural platforms.
- Representation of Platform Specific Model (PSM) from PIM. At this level, the environment of implementation platforms or languages is known.

MDA allows designing a workflow based on different mappings from CIM to PIM and from PIM to PSM. The mappings can be automated, particularly when a model type mapping specifies the mapping rules in a meta-model.

In this manner, an MDA approach increases the interoperability in heterogeneous environments and provides a method for systems integration using generic mapping engines.

In recent years, the W3C Semantic Web Best Practices and Deployment working group has proposed to extend the MDA methodology by using semantic models or ontologies (e.g. RDF and OWL languages). This extension has been defined as the Ontology Driven Architecture (ODA) methodology [3]. ODA is aimed at extending MDA by providing representation of unambiguous domain vocabularies (e.g. requirements, constraints, services, properties, etc.), model consistency checking and validation as well as to enable new automatic software engineering capabilities. Several works have been carried out in order to illustrate how an ontology-based approach can be used in the design, development and management of distributed systems. For instance, in [4], an ontology-based application server has been designed allowing the design, development and management of software components. This ontology is aimed at describing the properties, relationships and behaviors of the software components. Based on this ontology, the components can be queried, preloaded, checked and composed during development as well as during run time. Another interesting application of ontologies within the MDA process is presented in [5], where ontologies are used to represent knowledge about platform diversity and how this information is used to perform safe configuration of refinement transformation between the platform models.

In order to facilitate the deployment of reusable components, Service Oriented Architecture (SOA) approach can be used for defining applications based on existing or new services. The Service Component Architecture (SCA) is a set of specifications, which support the development of applications based on SOA. SCA proposes organizing in components the development and deployment of applications. Each component may offer services or need references in order to accomplish the offered services. The assembly of the components by interconnecting (wiring) services and references indeed allows building applications whether they are distributed or not, running in one process or multiple processes. In addition, SCA allows the integration of heterogeneous components written in many languages like Java, C++, PHP, JavaScript, and BPEL [6].

While the components are the base element in SCA, composites are the elements integrated by components. Composites may offer services through the promotion of components' services, and may need other composites' references. The remote access to composite's services can be made through the use of bindings (e.g. Corba IIOP, web services, etc.). Finally, a set of interrelated composites within the same vendor's SCA implementation forms a Domain [7]. In order to describe a composite, SCA uses the Service Component Definition Language (SCDL). SCDL is a XML based formatted language

which allows characterizing the different components that form composites, and specify the relationships between them. SCDL works like a deployment descriptor for SCA applications.

2.2. Multimedia standards, protocols and implementations

During last decades, several organizations such as the ITU, W3C and IETF, have invested important efforts in standardization activities related to multimedia systems. Next paragraphs summarize several of these standards.

ITU-T recommendation F.700 (ITU-T F.700)

F.700 proposes a framework for characterizing multimedia services. From a functional point of view this standard provides a methodology for the development of multimedia services considering needs of both final users and service providers. The approach proposed in F.700 is based on a four-level model, in a top-down order: Application, Service, Communication Task, and Media Component levels. The Application level describes the functional characteristics from a user point of view. The Service level includes services or tools that satisfy the functional requirements of the application level. Services like QoS, security, or intercommunication are defined in the service level. According to the model, the construction of services is done by combining communication tasks and coordinating their interactions. Downwards at the Communication task level, communication tasks are defined as functional entities of multimedia services, and they handle media components in order to transport information. Functions like transfer, storage, and switching are defined in this level. Finally, at the bottom of the model, the media component level deals with the multimedia aspects of the services by describing the monomedia components such as audio, video, etc, of user information and by establishing functions like capture, coding, presentation, etc. Regarding control activities, ITU-T F.700 proposes a control and processing plane, which interacts with the service, communication tasks, and media component levels through middleware service elements.

In a top-down approach F.700 suggests the decomposition of a multimedia service into communication tasks controlled by user and/or service providers. In a bottom-top approach a communication task can be viewed as the means of gather the media components required for multimedia service. Being communication tasks the means for composing multimedia services and for gathering the media components make their description particularly important. In order to do so, ITU-T F.700 proposes three attributes: communication configuration, control entity, and

information flow to describe generic communication tasks like sending, conversing, conferencing, distributing, collecting, and receiving.

In other words, ITU-T F.700 gives the possibility to describe and to construct multimedia services in an automatic manner, for instance by developing an ontology of F.700 which describes multimedia services composition. This ontology being computer readable facilitates automatic configuration and/or construction of the multimedia services. Further, by integrating other standards along with ITU-T F.700, one can describe/construct multimedia services considering simultaneously aspects like QoS (ITU-T X.641), and/or user requirements for delay and information lost (ITU-T G.1010).

IETF MMUSIC and AVT standards

The Multiparty Multimedia Session Control (MMUSIC) working group of the IETF has been responsible for the specification of the more widely used session protocols for multimedia systems. The Session Description Protocol (SDP) provides a common representation for expressing media and session descriptions. SDP proposes an entirely textual data format to maximize portability among distributed applications. Recently, a new effort for session description called SDPng has been proposed to increase interoperability and extensibility by using XML-based syntax. Other protocols standardized by the MMUSIC work group are the very well known SIP, RTSP and SAP session control protocols:

- The Session Initiation Protocol (SIP) is widely used for interactive applications (e.g. VoIP, video-conferencing, etc.).
- The Real Time Streaming Protocol (RTSP) is used as the session control protocol for multimedia sending applications (e.g. IPTV, video on demand, etc.).
- The Session Announcement Protocol (SAP) is used for assisting the advertisement of multicast multimedia conferences and other multicast sessions.

Another protocol proposed by the IETF is the Real Time Transport Protocol (RTP). RTP has been proposed by the Audio Video Transport (AVT) working group as the standard protocol for the representation of streaming multimedia flows. RTP is network and transport-protocol independent, though it is often used over UDP. Currently, RTP is used as the standard for both unicast and multicast streaming network services.

Software implementation platforms

A large diversity of multimedia frameworks aimed at developing distributed multimedia applications has been released in the last years. Most of them have been implemented in Java, C/C++ and .Net environments. Examples of these frameworks are:

- Java Media Framework API (JMF): JMF extends the Java 2 Platform, Standard Edition (J2SE) by providing a cross-platform technology for time-based media processing (e.g. capture, playback, streaming and transcoding).
- Java Micro Edition (Java ME): provides a standard environment for applications running on mobile and other embedded devices (e.g. mobile phones, PDAs, TV set-top boxes, etc.). The Mobile Media API (MMAPI) is an optional package within the Java ME platform aimed at providing a standard API for processing time-based media.
- Android: Android is Java-based platform for mobile devices including an operating system, middleware layer and key services and applications. Android environment provides support for building applications integrating multimedia-processing functions.
- iPhone: the iPhone operating system also proposes an environment to develop rich multimedia applications. This environment mainly uses the Objective-C programming language, an extension of the ANSI C language.
- Other: platforms such as the .NET / Windows Mobile OS also propose frameworks for developing multimedia services.

This non-exhaustive list of specific software platforms shows the diversity of languages and environments available today for developing distributed multimedia systems. This diversity of software platforms associated to the various multimedia standards and protocols has motivated us to propose a Multimedia Ontology Driven Architecture framework (MODA) for Networked Multimedia Systems.

3. MODA framework

In the previous section, a state of the art of model driven oriented methodologies as well as service component oriented architectures has been introduced. Likewise, standard multimedia frameworks as well as legacy multimedia protocols have been presented. Moreover, the diversity of languages and developing environments for the implementation of multimedia applications has also been presented. This section introduces the Multimedia Ontology Driven Architecture framework (MODA). MODA is intended to be used for automatic networked multimedia systems deployment and configuration. Based on well-known standards, this framework integrates multimedia, distribution and platform specific concepts in order to allow dynamic translation between the user/community requests and the components deployment specification. Deployment specifications produced by the framework can also be used to self configure the communication system (e.g. firewall,

NAT, etc.) and even to deploy adequate communication components at middleware, transport or network layer in order to provide necessary quality of service to guarantee the multimedia session requirements. Due to space limitations, in this paper only the framework basis guiding the automatic generation of deployment specifications will be introduced. Next paragraphs present the ontologies composing the MODA framework as well as the process allowing the automatic generation of the deployment specifications.

3.1. Ontologies

The ITU multimedia services ontology

In this section we describe our ontology based on recommendation F.700. As we mentioned before, an important part of F.700 is the description and construction of multimedia services based on their communication tasks. The F.700 ontology is focused on the description of communication tasks through their attributes. In this manner, a communication task has:

- Communication Configuration. In order to express if the communication is point-to-point, point-to-multipoint, multipoint-to-point, or multipoint-to-multipoint.
- Symmetry of information flow. To specify the direction in which the information is sent.
- Transmission control entity. It allows saying who is controlling the transmission of the information (e.g. source and/or sink).
- Communication delay. The type of delay supported by the communication task (e.g. real time)
- Media. The media or medium, mandatory or optional ones, transmitted by the communication task; as well as the quality level that the media has.
- Media Interrelation. It allows specifying if there is some synchronization between media (e.g. lips or subtitles synchronization), or symmetry in order to indicate bidirectionality of the same media type, or conversion between media in order to indicate when a media is converted into another type of media, i.e. when graphics are converted into still pictures.

In the figure 1 we can observe the relations between the main classes of our ITU-T F.700 ontology.

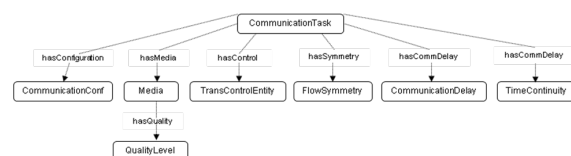


Figure 1. ITU multimedia services ontology

In this ontology the communication tasks sending, conversing, conferencing, distributing, collecting,

and receiving are added as subclasses of CommunicationTask. We also define all the relations (object properties) of the CommunicationTask class and its subclasses so we can describe individuals which correspond properly to any of these communication tasks with their attributes.

IETF protocols ontology

Based on the various protocol standards specified by the IETF, a multimedia protocols ontology integrating session control, session description and transmission protocols has been elaborated (Figure 2). Individuals of this ontology are represented by applications implementing RTSP, SIP or SAP session control protocols in order to negotiate and establish the required multimedia session described by the implementation of the SDP protocol. For most of these applications, the RTP/RTCP protocol will be used for streaming the multimedia data between the distributed system components (e.g. server to client for a VoD system).

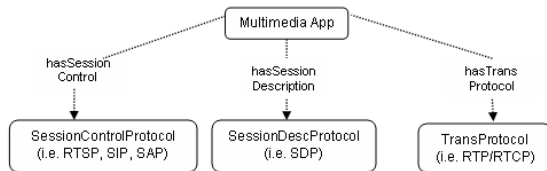


Figure 2. IETF protocols ontology

Service Component Architecture ontology

The SCA Assembly Model defines the configuration of SCA domains using composites, components and the artifacts that allow describing how they are connected or linked. As a matter of fact there are no many concepts used in the SCA Assembly model in order to provide the SCA's programming model. So we have developed a SCA ontology integrating all of these concepts, their relationships and attributes. Figure 3 depicts our SCA ontology with its main classes.

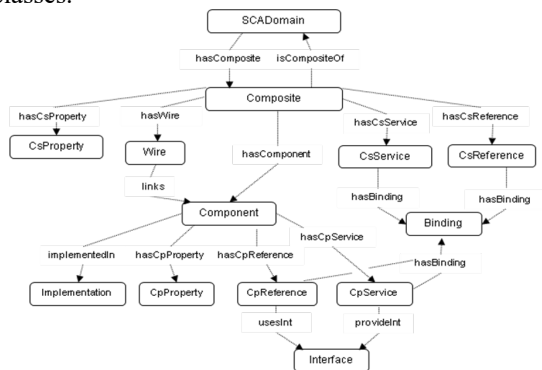


Figure 3. SCA ontology

Software implementations ontology

In order to facilitate the automatic translation to platform specific models using an ontology driven approach, a multimedia specific platform ontology

describing the components of generic multimedia systems has been elaborated (Figure 4). This ontology mainly specifies the two basic components to be deployed in every host participating in the multimedia session: the session and the media controller. The former is suited to describe, negotiate, establish and terminate the multimedia session between the participants. The second is in charge of the multimedia processing functions (i.e. capturing, coding/decoding, streaming, receiving, recording and presenting). In order to incorporate a specific platform to the MODA framework, individuals of that specific platform needs to be instantiated. For instance, in order to integrate the JMF implementation, one JMF individual related to one or several session control components (e.g. RTSP server, RTSP client, SIP agent, etc.) and to one or several media controller components (e.g. Player, Processor, etc) has to be added.

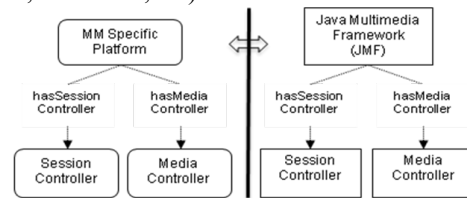


Figure 4. Software implementations ontology

3.2. MODA Process

The MODA process includes the three levels recommended by OMG. In our case, our proposition of CIM presents networked multimedia systems taking into account the different standards' ontologies. The PIM level uses a Service Component Architecture (SCA) model to obtain a generic description of networked multimedia systems from CIM. For the PSM level, implementation ontologies can be used to translate our PIM of networked multimedia systems.

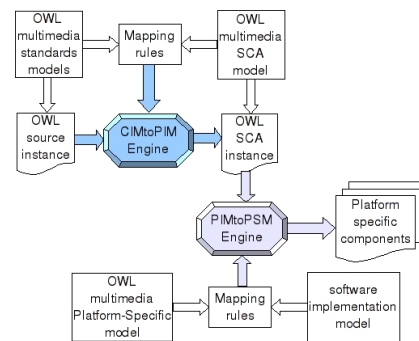


Figure 5. MODA Process

Figure 5 shows the MODA process to build automatically a multimedia networked system. The first step of our workflow is the transformation of OWL multimedia source instance through CIMtoPIM engine to produce the SCA Instance. To

make this transformation, the engine uses declared mapping rules which expresses how data in one format should be converted to another format. In the second step, PIMtoPSM engine takes as input the SCA instance and produces the deployment of specific platform components taking into account the mapping rules between OWL multimedia specific platform (Figure 4) and software implementation (e.g. JMF or Java ME).

4. Case study. Video on Demand (VoD)

We have evaluated the MODA framework for the automatic deployment of a VoD system. In order to characterize such type of application, the MODA ontologies have been used to describe the communication tasks involved. Figure 6 shows the CIM individual describing our VoD application.

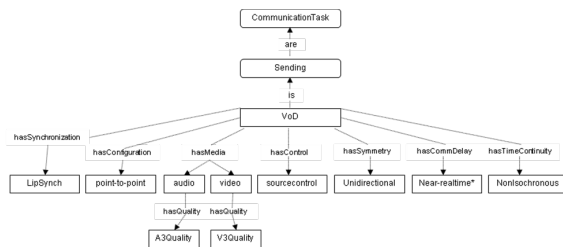


Figure 6. VoD study case

The MODA process (CIMtoPIM) allows generating from the Sending communication task a SCA domain including both server and client composites. For each composite, the SessionController and MediaController components are inferred. Following the MODA process (PIMtoPSM), the adequate session controller (i.e. RTSP server and client) as well as the media controller implementations (i.e. JMF media controllers) have been selected. Figure 7 depicts the PSM SCA individual for the VoD system.

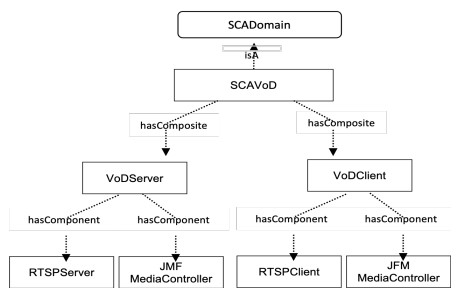


Figure 7. SCA description of the VoD system

XSL templates used to implement the mapping rules guiding the MODA engines are detailed at [8].

5. Conclusions and perspectives

In this paper, the Multimedia Ontology Driven Architecture framework (MODA) has been introduced. This framework is intended for automatic networked multimedia systems deployment and configuration. This framework is based on well-known standards and proposes a set of ontologies integrating multimedia, distribution and platform specific concepts. Furthermore, this framework also integrates an automatic process aimed at guiding the dynamic generation of multimedia system deployment configurations. The use of this framework in order to allow dynamic translation between the user/community requests and the components deployment specification has been illustrated by the instantiation of a Video on Demand system. MODA is still in a prototyping phase and its development is in progress within the framework of the Feel@Home project [9], but the preliminary results allow us to realize the benefits of following such approach to automatically deploy multimedia services for community of users. Next steps aim at generating dynamic user interfaces for final users (to drive the deployment process) and enhancing deployment specifications produced by MODA in order to be used for self configure the communication system (e.g. firewall, NAT, etc.). We also intend to deploy the adequate communication components at middleware, transport or network layer in order to provide the quality of service required to guarantee the multimedia session requirements. Due to space limitations, in this paper only the framework basis guiding the automatic generation of deployment specifications have been introduced. Further information about the MODA framework can be found at [8].

6. References

- [1] "MDA Guide Version 1.0.1", 12 June 2003, document number: omg/2003-06-01, Editors : Joaquin Miller, Jishnu Mukerji.
- [2] J. D. Poole, "Model-Driven Architecture: Vision, Standards And Emerging Tech.", in Workshop on Metamodeling and Adaptive Object Models, ECOOP 2001
- [3] "Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering", <http://www.w3.org/2001/sw/BestPractices/SE/ODA/>
- [4] D. Oberle et al, "Developing and Managing Software Components in an Ontology-based Application Server", ACM/IFIP/USENIX 5th International Middleware Conference, Toronto, Ontario, Canada, volume 3231 of LNCS, pp. 459-478.
- [5] D. Wagelaar, "Platform Ontologies for the Model-Driven Architecture", Thesis dissertation, Faculty of Science of the Vrije Universiteit Brussel, 2008
- [6] G. Barber, "What is SCA", August 2007, <http://www.osoa.org>
- [7] D. Chappell, Introducing SCA, <http://www.davidchappell.com>

[8] Multimedia Ontology Driven Architecture framework,
[http://www.laas.fr/~eexposit/pmwiki/pmwiki.php/MODA/
Main](http://www.laas.fr/~eexposit/pmwiki/pmwiki.php/MODA/Main)

[9] Feel@home project website, [https://rd-projet-
feelathome.rd.francetelecom.com/public](https://rd-projet-feelathome.rd.francetelecom.com/public)