



**HAL**  
open science

## Towards user-centric configuration and deployment of multimedia services: A semantic framework

Jorge Gomez-Montalvo, Ernesto Expósito, Myriam Lamolle

### ► To cite this version:

Jorge Gomez-Montalvo, Ernesto Expósito, Myriam Lamolle. Towards user-centric configuration and deployment of multimedia services: A semantic framework. Workshops (ICUMT), Oct 2009, St. Petersburg, France. pp.1-7, 10.1109/ICUMT.2009.5345421 . hal-03430704

**HAL Id: hal-03430704**

**<https://hal.science/hal-03430704>**

Submitted on 16 Nov 2021

**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.

# Towards user-centric configuration and deployment of multimedia services: A semantic framework

Jorge Gomez-Montalvo<sup>1,2,4</sup>, Ernesto Exposito<sup>1,2</sup>, Myriam Lamolle<sup>3</sup>

<sup>1</sup> CNRS ; LAAS ; 7 av. du Colonel Roche, F-31077 Toulouse, France

<sup>2</sup> Université de Toulouse; UPS, INSA, INP, ISAE; LAAS ; F-31077 Toulouse, France

<sup>3</sup> LIASD/LINC, IUT de Montreuil, Université Paris VIII

<sup>4</sup> Facultad de Matemáticas, Universidad Autónoma de Yucatán, México

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

m.lamolle@iut.univ-paris8.fr

**Abstract**—New communication service capabilities of mobile devices have boosted the apparition of ubiquitous multimedia scenarios where users could easily create ad hoc multimedia sessions by friendly providing their requirements and preferences. In order to dynamically create and deploy such ubiquitous multimedia systems, end-users require frameworks which allow them in a spontaneous manner to generate the required multimedia sessions. This work introduces an ontology based framework aimed at the configuration and deployment of multimedia services driven by user preferences. A case study is presented in order to illustrate the use of this framework in the context of extended home networks.

**Keywords;** *Ontologies; Multimedia Services; MODA*

## I. INTRODUCTION

The explosion of the mobile devices with communication service capabilities has contributed to the apparition of ubiquitous multimedia scenarios where users could easily create ad hoc multimedia sessions. Besides, the role of end-user in networked multimedia systems has been evolved in the last years; it has changed from being just the consumer to the producer of the media content like happens in social networks. In the future ubiquitous multimedia scenarios we envisage frameworks that take into account user's preferences in the construction and configuration of multimedia sessions. Indeed it will be necessary frameworks which allow configuring and deploying spontaneously multimedia sessions in order to met new ubiquitous multimedia scenarios.

Even when relevant efforts have been made in order to propose frameworks for the configuration and deployment of services/application, a generic multimedia framework aimed at user-centric configuring and deploying of multimedia services has not been defined yet. Such multimedia framework should be able to take into account ubiquitous multimedia system characteristics as well as user preferences in order to guarantee end-user satisfaction. In this sense, such multimedia framework should be able to characterize the potential configurations of the multimedia services as well as their sessions. Furthermore the framework should allow taking advantage of the semantic of such characterization in

order to enable spontaneous generation of multimedia sessions.

In this paper we propose the Multimedia Ontology Driven Architecture (MODA) framework. The MODA framework targets to session generation and configuration, in a spontaneously manner, of ubiquitous multimedia systems and applications. MODA is based on well-known standards which include multimedia service and session concepts. This work also illustrate how can be integrated other ontologies to the framework in order to considering potential session configuration regarding aspects like e.g., user preferences or QoS. The framework allows to producing and deriving the information required to create session and deployment descriptors driven by user preferences.

This paper is organized as follows. Second section introduces the state of art on multimedia service standards and protocols as well as the service component architecture which provide the basis of our MODA framework. Section 3 presents the basic ontologies that integrate the MODA framework. Section 4 presents a case study intended to illustrate how MODA framework can be used to spontaneously create and configure multimedia services. Finally Section 5 presents the conclusion and perspectives.

## II. STATE OF ART

This section presents the state of the art which provides the basis of the MODA framework. It summarizes the effort that organizations like IETF and ITU have done in order to propose multimedia standards. Also, it is presented the work that organizations like the W3C and OSOA have done towards the development and deployment of service oriented applications.

### A. ITU-T F.700 Recommendation (ITU-T F.700)

ITU-T F.700 recommendation [1] 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 ITU-T 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 also 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. At this level functions like capture, coding, presentation, etc. are established. 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 ITU-T 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. Since communication tasks are the means for composing multimedia services and gathering the media components, their description is particularly important. ITU-T F.700 proposes three attributes: communication configuration, control entity, and information flow in order 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, when integrating other standards along with ITU-T F.700, one can describe/construct multimedia services considering simultaneously aspects like QoS (ITU-T X.641) [2], and/or user requirements for delay and information lost (ITU-T G.1010) [3].

### *B. Session Description Protocol (SDP) and SDP New Generation (SDPng)*

The IETF Multiparty Multimedia Session Control (MMUSIC) Working Group has proposed the Session Description Protocol [4]. SDP is intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. In order to do so, SDP has to carry information like media details, transport addresses, and session description metadata to participants of the session (e.g. streaming, videoconference, VoIP sessions). Indeed, the idea of using SDP is to be able to describe the session information in a standard format.

As of now, examples of the use of SDP information include:

- With the Session Initiation Protocol (SIP) for creating, modifying, and terminating sessions particularly in VoIP.
- With the Real Time Streaming Protocol (RTSP) in order to controlling on-demand delivery of data with real-time characteristics.
- With the Session Announcement Protocol (SAP) for distributing session description information to potential participants in multicast sessions.

The diversity of use of SDP has led to requirements for which SDP was not originally designed. In order to fill these gaps several extensions have been proposed e.g., the offer/answer model with the SDP [5] and grouping of media lines in the SDP [6].

The MMUSIC Working Group has proposed SDPng as the SDP successor. SDPng considers besides the session description information, the dynamic aspects (e.g. parameters and configuration) of interactive sessions. One important technical characteristic of SDPng is that it uses an XML-based syntax in order to increase the expressiveness required for achieving these tasks.

We think that taking SDP and SDPng information as the basis for developing a session description model based on ontologies enables not only to describe the session information (format aspect) but also allows to integrate the meaning of information (semantic aspect). Further in an ontological framework which integrates for example semantic information of the session (SDP ontology) and semantic of multimedia services (ITU-T F.700 ontology) relationships between both domains can be defined in order to facilitate the spontaneous creation and configuration of interactive multimedia session.

### *C. Service Oriented Architecture (SOA)*

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) [7] is a set of specifications, which support the development of applications based on Service Oriented Architecture. SCA proposes organizing in components the development and deployment of applications. Each component offers services and may need references (services of other components). 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.

While the components are the base element in SCA, the composites are the elements which are integrated by components. Composites offer services through the promotion of components' services, and they 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. In order to describe a composite, SCA uses the Service Component Definition Language (SCDL). SCDL is a XML based formatted language which allows characterizing components and composites as well as specifying the relationships between them. SCDL works like a deployment descriptor for SCA applications.

#### D. Web Ontology Language (OWL)

OWL [8] has been proposed by the W3C as a language which OWL was designed to allow representing concepts and their relationships within a domain of knowledge. The OWL capability to express the meaning and semantics of concepts and relationships has made of OWL one of the most used languages to develop ontologies. Besides, OWL ontologies are machine interpretable which allow the use of applications, for example reasoners which process the information in order to derive implicit information from the ontology.

Considering the aforementioned characteristics of OWL, we decide use it for developing the MODA framework. The richness of OWL for describing concepts and relationships as well its semantic expression capability allowed us to cope with the MODA semantic requirements.

### III. THE MODA FRAMEWORK

In the previous section, it has been introduced the state of the art of multimedia standards and protocols as well as service component oriented architectures. This section introduces the MODA framework which aims to facilitate the spontaneous creation of ubiquitous multimedia systems and applications driven by user preferences (like application and/or user priorities).

#### A. ITU-T F.700 Ontology

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

- **CommunicationConf** (Communication Configuration). This class is used to express if the communication is point-to-point, point-to-multipoint, multipoint-to-point, or multipoint-to-multipoint.
- **FlowSymmetry** (Symmetry of information flow). It allows specifying the direction in which the information is sent.
- **TransControlEntity** (Transmission Control Entity). It allows saying who is controlling the transmission of the information (e.g. source and/or sink).
- **CommunicationDelay**. 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 of the media.
- **TimeContinuity**. This class allows expressing if the communication task is buffer capable or not.

- **MediaInterrelation**. 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.

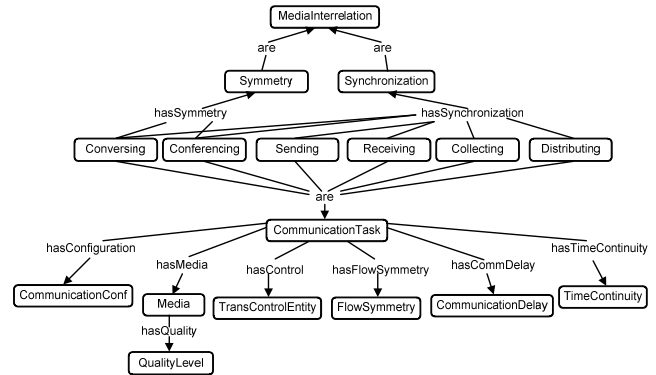


Figure 1. ITU F.700 multimedia services ontology

In the ITU-T F.700 ontology the communication tasks sending, conversing, conferencing, distributing, collecting, and receiving are subclasses of CommunicationTask class. We define all the relations (object properties) of the CommunicationTask class and its subclasses so we can describe individuals which correspond properly to any of the communication tasks.

#### B. Session Description Ontology

We have developed a session description ontology (SDO) based on the SDP and SDPng. This ontology makes use of the already presented ITU-T F.700 ontology in order to generate standard multimedia service descriptions. Also, this ontology utilizes our user/application priorities ontology in order to integrate user preferences in the session. The figure 2 illustrates the main classes of the SDO.

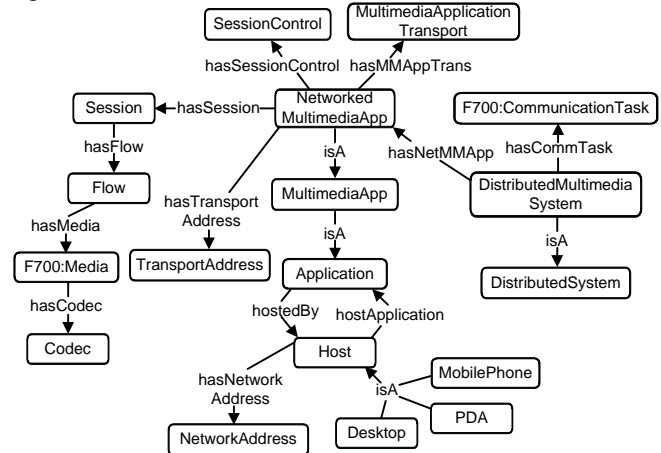


Figure 2. Session Description Ontology

The following paragraphs describe in detail the purpose of the main classes of the session description ontology:

- **Application.** This class characterizes the applications. Application class is used by the user/application priorities ontology (figure 3).
- **MultimediaApp.** This class characterizes multimedia applications which are a subclass of the Application class.
- **NetworkedMultimediaApp.** Since multimedia applications may be local or networked this class characterizes networked multimedia applications (clients, servers or peers).
- **Session.** This class describes some components of a networked multimedia session. For example, using the session class the media flows of the networked multimedia applications are described.
- **SessionControl.** Individuals of this class are session control protocols like RTSP, SIP or SAP. Indeed, the SessionControl class allows knowing the control session protocol used as well as its characteristics.
- **MultimediaApplicationTransport.** This class describes the transport protocol used to transfer the media data. For most of the networked multimedia applications, the RTP/RTCP will be used to accomplish the multimedia data transportation. The MODA framework allows integrating information required by multimedia application transport protocols like RTSP through its corresponding ontology. It is important to remark that the information required to characterize the transport protocol (e.g. RTSP) is available in this ontology e.g. network address, flow information, etc.
- **Flow.** This class characterizes the flow created by the multimedia application during the session.
- **F700:Media.** This class belongs to the ITU-T F.700 ontology (see figure 1).
- **Codec.** This class describes the codec and its parameters used to transmitting the multimedia data. Clearly the codec instance depends on the media in the flow.
- **DistributedMultimediaSystem.** This class characterizes a distributed multimedia system that is composed by two or more networked multimedia applications.
- **Host.** This class describes the device that is hosting the application. It includes a number of subclasses representing several devices like mobile phones, PDAs, and Desktop. Also, information about the characteristics of the device is considered (e.g. display resolution).

### C. User/Application Priorities (UAP) Ontology

We have developed the user/application priorities ontology in order to express the priorities between users and applications. This ontology is aimed at allowing users to describe their priority preferences between home users

and/or applications using shared network resources. UAP ontology is depicted in the figure 3.

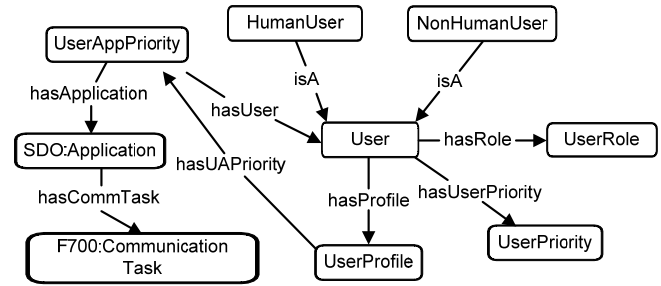


Figure 3. User/Application Priorities (UAP) Ontology

The following are the main classes of UAP ontology:

- **User.** This class permits the description of the user. There are two types of user: human (HumanUser) and non human (NonHumanUser) users. A human user can have a name, username, and e-mail as properties.
- **UserPriority.** It allows a user (e.g. home network administrator) to express priorities between users.
- **UserProfile.** This class allows a user (any user) to predefine within profiles the priorities between his applications considering different situations (e.g. application's priorities while working, playing, etc.).
- **UserRole.** This class represents the different roles a user can play. For instance: administrator or home user.
- **UserAppPriority.** It allows a user to define the priorities for his applications. So the user can express the maximum and minimum numerical priority values for his applications as well as the numerical application priority itself.
- **SDO:Application.** This class belongs to the session description ontology (see figure 2).

### D. Service Component Architecture ontology

The SCA Assembly Model defines the configuration of SCA domains using composites, components and the artifacts which 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, relationships, and attributes. Figure 4 depicts the main classes our SCA ontology.

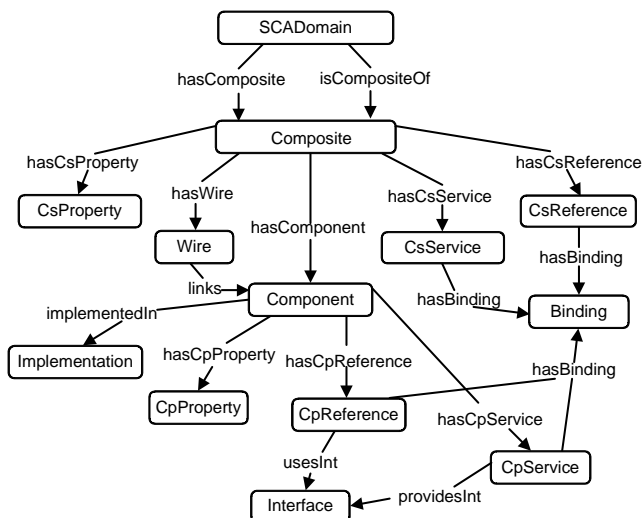


Figure 4. Service Component Architecture ontology

The purpose of developing the SCA ontology is to be able to create deployment descriptors using the Service Component Definition Language. Such descriptors facilitate the deployment of multimedia services when the SCA programming model is used to develop the multimedia services.

#### IV. CASE STUDY

In this section a case study is presented in order to show some of the advantage of using the MODA framework. The MODA framework is used in the context of Feel@Home project [9]. The Feel@Home project focuses in scenarios where Feel@Home users easily share with other users multimedia contents whether they are inside or outside home. Using the Feel@Home context we want to illustrate how the MODA framework allows configuration and deployment of spontaneous multimedia sessions to be done.

The first part of the case study includes the characterization of the multimedia services. Let's suppose that the home user wants to execute a Video on Demand (VoD) application. Using the MODA framework is it possible to characterize a specific VoD application and then configured using a friendly graphic user interface. This characterization is as follows:

- **Communication Configuration.** The ontology only allows a point-to-point configuration to be established because it is a sending task. With this information we can obtain/infer that exist a source and sink host (both of them having parameters like IP address and ports).
- **Communication Delay.** Since it is a sending task, the GUI enables the options near-real-time, real-time, non-real-time, specified-time. However, due to the user is defining a VoD application, he has the options: "wait a few seconds but have better quality" (near-real-time); "start immediately (real-time); or "watch it 2hrs from now" (specified-time).
- **Symmetry of information flow.** By definition a sending task is unidirectional regarding information

plane, and then VoD application has unidirectional symmetry of information flow. This means the audio and video will be sent just in one direction (from source to sink).

- **Transmission Control Entity.** The only option is "source control" because of the definition of sending communication task.
- **Time continuity.** Here there are two options: isochronous (i.e. if the user terminal equipment has no buffering capabilities, or the capturing device does not tolerate interruptions or variation in transmission speed), and non-isochronous. However, time continuity is related to communication delay so for instance for a VoD service the appropriate configuration would be an isochronous transmission with a near-real-time delay.
- **Media.** The user can set audio and video flows for a VoD application. According to ITU-T F.700 recommendation, so the ontology allows the user to set the quality level of both audio and video media (e.g. "a speech quality", equivalent to an A3 quality level (MPEG4 codec), and "High definition video quality", equivalent to a V4 quality level (High Definition video quality)).
- **Media Interrelation.** Within a VoD application with audio and video flows there is a relationship of synchronization between them. Through the ontology it is possible to configure the type of synchronization e.g. lips synchronization (audio and video flows), or subtitles synchronization (video and text), or both of them lips and subtitles synchronization (audio, video and text).

Once the multimedia services are characterized, the MODA framework can generate the session description by instantiating the proposed session description ontology as follows:

- **NetworkedMultimediaApp.** The instance of this class is actually the VoD application (one instance for the client and one for the server) that the home user will execute. The VoD application has associated a transport address (port), a session control protocol which in this case would be RTSP for the VoD, and multimedia application transport protocol which in this case are RTP/RTCP.
- **Session.** The instance of this class includes more session information. Specifically it allows describing the flows created by the VoD application. It also may include session name, session information, and email of the responsible of the session.
- **SessionControl.** As mentioned before, for the VoD application the session control protocol is RTSP.
- **MultimediaApplicationTransport.** The multimedia application transport protocols for the VoD application are RTP/RTCP. The information required by RTP is taken from the session instantiation of the VoD application. This class describes the transport protocol used to transfer the

media data. For most of the networked multimedia applications, the RTP/RTCP will be used to accomplish the multimedia data transportation.

- Flow. For the VoD will be created two flows: audio and video flow. Besides, the audio and video instances links to more information about the flows like for example the multimedia application protocol used, in this case RTP, the transport port(s) to which the media flow is sent.
- F700:Media. Audio and Video instances will be created corresponding to audio and video flows for the VoD application. Each media instance has also information as the format of the media e.g. the AVI format.
- Codec. This individual characterizes the media codec used during the session if any e.g. MPEG.
- DistributedMultimediaSystem. This instance represents the all multimedia system itself. In this case the all VoD application. This individual uses in turn the ITU-T F.700 sending task instance to create the component of the VoD application.
- Host. This instance can be a type of mobile phone including its characteristics.

Once the session description is done, the home user can set priorities for them, and in this case for the VoD example. In order to do so, MODA framework allows using a GUI where the user can set his multimedia services priorities as follows:

- User. An individual of user class has a name John Doe. John Doe has the username jdoe, and the email account john.doe@home.com.
- UserRole. Given that John Doe owns the home network, the ontology GUI allows him to set his role as an administrator user.
- UserPriority. John creates two more users Alice and Bob Doe. John defines Alice and Bob priorities, so John defines the minimum priority for a user as 0 and the maximum priority 10. John also defines that Alice's priority is 7, while Bob's priority is 4. John sets 10 for himself as his priority.
- Application. As applications the user can select one of the applications previously instantiated using ITU-T F700 ontology e.g. the VoD. In this manner, John, Alice and Bob can also instantiate conversing tasks representing applications such as Skype.
- UserAppPriority. The idea of having a user's application list is that the user be able to define the priorities between his applications, so these priorities can be take into account for QoS provisioning. So for example John has defined in his application list for VoD applications a priority value of 5, while his conversing applications have a value of 9, within a scale ranging from a minimum priority of 0 to a maximum of 10. Alice has defined for her VoD applications a priority value of 6, and for her conversing applications a priority of 10 within the same range from 0 to 10. Finally, Bob has defined

for his VoD applications a priority value of 5 and for his conversing applications a priority value of 5 within the same scale range.

- UserProfile. John, Alice and Bob may want to define several profiles for their applications, for example John defines a profile "watching films" where his VoD applications have a higher priority than any other application; and John also defines a profile "conversing for work" where his conversing application have a higher priority than any other application. Alice and Bob can create the same kind of profiles.

Finally, in order to create the deployment descriptor of the VoD application the SCA ontology is instantiated as depicted in the figure 5.

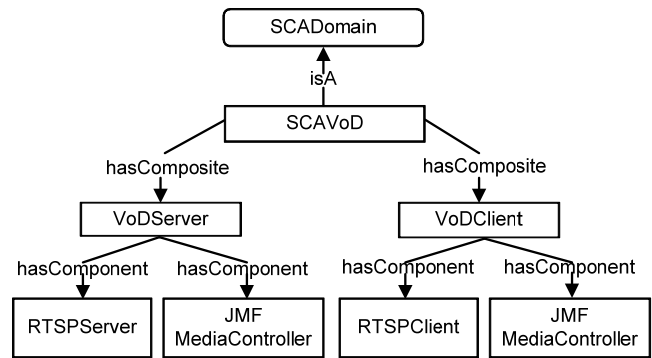


Figure 5. SCA descriptor of the VoD application

Because of the ontology's interrelations of the MODA framework, it is possible to define rules in order to have suitable session configurations and also to take decisions in a more intelligent manner when it is necessary chose between potential session configurations. In [10] is presented the process for the automatic deployment of multimedia services using the MODA framework.

## V. CONCLUSION AND PERSPECTIVES

In this paper the Multimedia Ontology Driven Architecture (MODA) has been presented. The MODA framework is intended to enable the configuration and deployment of multimedia services driven by user preferences. Also, the MODA framework allows exploiting the semantic of the session description as well the user preferences. Despite of MODA framework is still under developing, preliminary results obtained in the context of the Feel@Home project show us the advantages of using the framework for the session configuration and deployment of multimedia services in extended home networks. Future work aims at the integration in the MODA framework of QoS and QoE aspects. We believe that by integrating semantic description of QoS characteristics and parameters will provide a common QoS vocabulary for both service clients and service provider.

## REFERENCES

- [1] ITU-T F.700. Framework Recommendation for Multimedia Services. <http://www.itu.int/rec/T-REC-F.700-200011-I/en>
- [2] ITU-T X.641. Information Technology – Quality of Service: Framework.
- [3] ITU-T G.1010. End-user multimedia QoS categories. <http://www.itu.int/rec/T-REC-G.1010-200111-I/en>
- [4] Handley, M. and Jacobson, V. 2009. *Sdp: Session Description Protocol*. RFC4566bis.
- [5] Rosenberg, J. and Schulzrinne, H. 2002. *An Offer/Answer Model with Session Description Protocol (Sdp)*. RFC3264
- [6] Camarillo, G., Eriksson, G., Holler, J., and Schulzrinne, H. 2002. *Grouping of Media Lines in the Session Description Protocol (Sdp)*. RFC3388
- [7] OSOA. Service Component Architecture, Assembly Model specification V1.00. 2007. <http://www.osoa.org/>
- [8] W3C. Web Ontology Language. <http://www.w3.org/2004/OWL/>
- [9] Feel@Home project website, <https://rd-projetfeelathome.rd.francetelecom.com>
- [10] Gomez J., Lamolle M., Exposito E., " A Multimedia Ontology Driven Architecture framework (MODA) for Networked Multimedia Systems", NDT2009, to be presented in Ostrava, June 2009, in press.