Multimedia Systems (2008) 14:327–340
DOI 10.1007/s00530-008-0134-0

REGULAR PAPER

Canonical processes of semantically annotated media production

Lynda Hardman · Željko Obrenović · Frank Nack · Brigitte Kerhervé · Kurt Piersol

Published online: 16 July 2008

Abstract While many multimedia systems allow the association of semantic annotations with media assets, there is no agreed-upon way of sharing these among systems. As an initial step within the multimedia community, we identify a small number of fundamental processes of media production, which we term canonical processes. We specify their inputs and outputs, but deliberately do not specify their inner workings, concentrating rather on the information flow between them. We thus identify a small set of building blocks that can be supported in semantically aware media production tools. The processes are identified in conjunction with a number of different research groups within the community who supply, in the companion papers, descriptions of existing systems and a mapping to them. We give a basic formalisation of the processes and discuss how this fits with other formalisation endeavours. We present a number of frequently asked questions during the development of the model and this special issue.

Lynda Hardman is also affiliated with the Technical University of Eindhoven.

L. Hardman · Ž. Obrenović (✉)
CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
e-mail: zeljko.obrenovic@cwi.nl

F. Nack
University of Amsterdam, Kruislaan 419,
1098 VA Amsterdam, The Netherlands

B. Kerhervé
Département d'Informatique, UQAM, C.P. 8888,
Montreal, QC, Canada

K. Piersol
Ricoh Innovations, Inc., 2882 Sand Hill Road/115, Menlo Park,
CA, USA

1 Introduction

There is substantial support within the multimedia research community for the collection of machine-processable semantics during established media workflow practices [4,5,10,12,14]. An essential aspect of these approaches is that a media asset gains value by the inclusion of information about how or when it is captured or used, and how it is manipulated and organised. For example, metadata captured from a camera on pan and zoom information can be later used for supporting an editing process [3]. Though the combination of description structures for data, metadata and work processes is promising, current approaches share an essential limitation, namely that the descriptions are not shareable. The problem is that each approach provides an implicit model for exchanging information that serves the particular functionality and process flow addressed by a particular environment.

To address the issues of metadata capture, preservation and exchange, we propose an approach to improving interoperability of (semantically-rich) multimedia systems based on a model of canonical processes of media production. Our hypothesis is that the existence of such a model can facilitate interoperability among software built by the multimedia community. We have chosen the term *canonical*[1] to indicate that a canonical process represents the highest level of abstraction for description of a process that can be shared among systems. Non-canonical processes can be constructed by combining canonical ones.

The desired outcome of this special issue is to gain community agreement on a small number of very basic processes involved with capturing, interpreting and annotating media

[1] Canonical: "reduced to the simplest and most significant form possible without loss of generality", see http://wordnet.princeton.edu/perl/webwn?s=canonical.

and publishing. Our aim is to establish clear interfaces for the information flow across processes among distinct production phases so that compatibility across systems from different providers can be achieved. We see this as a first step towards a longer term goal—namely, to provide agreed-upon and rigorous descriptions for exchanging semantically annotated media assets among applications.

While there exist many standards that try to facilitate the exchange between the different media process stages [15], such as MXF (Media Exchange Format), AAF (Advance Authoring Format), MOS (Media Object Server Protocol), and Dublin Core, our goal was to capture and model the whole media production process, starting from the early ideas of authors. Our model is therefore not a replacement for, but complementary to existing models, where we aim at defining the framework where the existing standards can be integrated as detailed specialization or instances of particular concepts of the model of the canonical processes of media production.

Our goal is to encourage system creators to provide the outputs we identify when the processes are supported within their system. We hope that in this way the multimedia community will be able to strengthen itself by providing not just single process tools, but enabling these to belong to a (global) suite of mix and match tool functionality. Another important role of identifying canonical processes is providing a clear definition of concepts so that researchers can coordinate their efforts and instructors can explain concepts to students. As articles in this special issue show, authors often use different terminology to describe similar functionality.

The canonical processes should not be viewed as prepackaged and detailed, ready to be implemented. Our goal is rather to analyse existing systems to identify and generalise functionality they provide and, on the basis of the processes supported within the system, determine which inputs and outputs should be made available.

In the next section, we discuss the process of constructing the model. Then, we briefly describe the companion system papers and the application areas they cover. We discuss our requirements for describing a single canonical process. The main section of the paper, the definition of the canonical processes, identifies and describes a number of processes we see as being canonical to media production. We then discuss what we have achieved and still need to achieve with the current state of the canonical process descriptions.

2 Building the model of canonical processes as a community process

This special issue focusses on the identification of canonical processes and their mapping to real systems. In this paper we identify and define a number of canonical processes of media production and give an initial formal description. The other papers in the special issue describe existing systems, the functionality they support and a mapping to the identified canonical processes. These companion system papers are used to validate the model by demonstrating that a large proportion of the functionality provided by the systems can be described in terms of the canonical processes, and that all the proposed canonical processes are supported in more than a single existing system. We do not claim that the canonical processes are able to describe all desired functionality, and discuss issues with extending the canonical processes later in the paper.

This special issue is the result of discussions with many researchers and practitioners in the community. This was initiated in a workgroup on "Multimedia for Human Communication" at a Dagstuhl seminar 05091[2] with a follow-up workshop at ACM Multimedia 2005 [3] on "Multimedia for Human Communication—From Capture to Convey". Our goal with these discussions and the open call for papers for the special issue has been to establish community agreement on the model before presenting it here.

As one important message of this special issue, we want to emphasise that building a useful model must always include the agreement of a significant part of the community. The agreement on particular elements of the model and discussion about usability of the model for a particular domain are equally, or sometimes even more important, than detailed and rigorous formalisation. In this special issue we present not "yet another model", but an approach to building such a model, and the benefits of having a model that a significant part of the community agrees on. Our model is the result of long discussions in the multimedia community, and we present not only the result of this discussion, but also a number of system papers that discuss the model in a particular domain. The system papers, therefore, are not simple instances of the model, but an important part of the broader process of building the model. They contain valuable contributions that demonstrate benefits and problems of using the model in particular domains. The model itself has also developed during the preparation of this special issue based on comments from contributing authors.

3 Companion system papers

Papers in this special issue come from very diverse areas, namely feature extraction systems, professional news productions systems, new media art, hyper-video production, photo book production, non-linear interactive narratives, systems for production of media abstracts, and ambient

[2] http://www.dagstuhl.de/en/programm/kalender/semhp/?semnr=05091.

[3] http://www.cwi.nl/~media/conferences/mhc05/mhc05.html.

multimedia systems with complex sensory networks. The main contribution of each system paper is in mapping already existing media production processes to identified canonical processes of media production. Each paper also discusses pros and cons of using canonical processes for its specific domain, giving a more practical context about how to relate canonical processes to concrete systems.

Hyowon Lee et al. describe their system for structuring, searching and browsing personal image collections. The main goal of the system is to locate important or significant events in a person's life. The system takes as input images created by *SenseCam*, a small wearable personal device which automatically captures up to 2,500 images per day. The system processes these using feature analysis, allowing the user to browse the collection of images organised according to their novelty using different granularities, e.g. today, this week, this month. They identify three phases related to capturing, structuring and displaying SenseCam images.

Philipp Sandhaus et al. describe the *CeWe Color Photo Book* software system. This system allows users to design a photo book on a home computer and have it printed by commercial photo finishers. The system uses low-level feature analysis to select and group pictures while allowing the user to override the selection and suggested layout. The authors identify the different steps related to capturing, annotation and organisation of photo book layout, that are necessary before the photo book is ready to be printed.

Erik Mannens et al. describe the news production system implemented at the *Flemish Radio and Television* (Vlaamse Radio en Televisie, VRT). Their system supports a highly parallel production process, optimized for short cycle times and driven by the dynamic nature of news data and metadata. They present the operations from a professional news system, and explain how the mapping to canonical processes has helped them to get a better picture of the different information needs during the production process.

Stephan Kopf and Wolfgang Effelsberg describe a video adaptation application called *Mobile Cinema*, which optimises video for viewing on mobile devices based on features such as screen resolution, bit rate, and color depth of a display. Their system is based on common logistical production processes, supported by an extensive data model. The authors argue that the definition of canonical processes makes the comparison of two video adaptation applications easier and improves the exchange of modules between different systems.

Brigitte Kerhervé et al. illustrate how canonical processes can be used to describe the different stages in the life cycle of a new media artwork. Using the adaptive video artwork *The Man of the Crowd* as a case study, the authors discuss general processes involved in the production of this artwork, such as artwork design, media acquisition, artwork production and exhibition. Their work is a first step towards a general modeling framework for the description and documentation of new media artworks.

Fabrizio Nunnari et al. present a character-based guided tour system, called *Carletto the spider*. The system uses dramatisation to make presentations more engaging and improve the reception of the content by the user. The key technical issue of their system is segmentation of the presentation into audiovisual units and their annotation that facilitates editing on-the-fly in a way that guarantees dramatic continuity of interaction.

Dieter Van Rijsselbergen et al. present *FIPA*, a manufacturing system for the production of drama television and motion picture programmes. This professional production facility implements a production workflow based on common industrial manufacturing processes, guided by detailed annotation of individual aspects of the drama production process. The authors argue that using canonical processes helps in identifying core functionality, so that process implementations can be simplified and input and output from different processes can be coordinated for better integration with external systems.

Michael Hausenblas describes *NM2*, an authoring suite for the creation and testing of non-linear narratives. The system has been used in the production of non-linear narratives for domains such as news and sport, documentaries and interactive drama.

Ansgar Scherp presents *SemanticMM4U*, a component framework for creating personalised semantically-rich multimedia presentations. The framework supports the creation chain of semantically rich multimedia content, based on a detailed analysis of today's approaches and systems for authoring, personalising, and semantically enriching multimedia presentations.

Olivier Auburt et al. describe *Advene*, a framework for active reading of audiovisual documents. The authors describe active reading as an interactive activity, where users through interaction and advanced visualisation consume but also produce metadata, encapsulated within multimedia objects called hypervideos. The authors relate Advene active reading processes to the canonical processes.

4 Describing canonical processes

A process is defined in terms of its inputs and outputs and is independent of whether the process can, or should, be carried out by a human or a machine. This allows for a gradual shift of the processing burden from human to machine as technology develops.

To assist in describing the processes we use the Unified Modeling Language (UML[4]). Although UML has limitations,

[4] http://www.uml.org/.

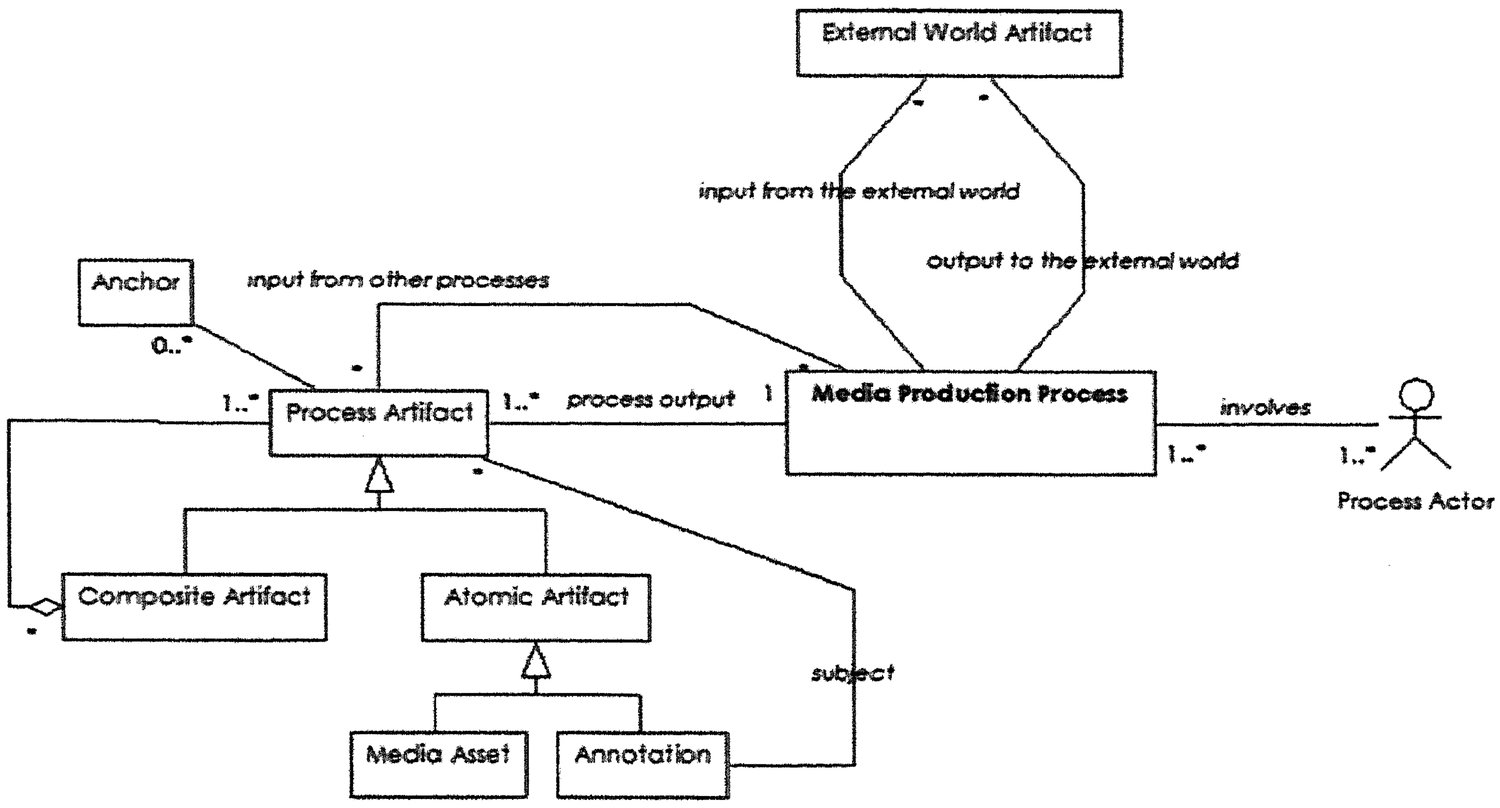

Fig. 1 The metamodel of canonical processes of media production

it is a widely adopted standard, familiar to many practitioners, widely taught in undergraduate courses, and supported by many books and training courses. In addition, many tools from different vendors support UML.

In this section we describe a metamodel that defines a vocabulary of modelling primitives used to describe the media production processes. We then introduce a number of UML extensions based on the metamodel, which are used to describe the canonical processes in the following section.

4.1 Metamodel of canonical processes

Figure 1 shows the metamodel where we formally describe basic concepts of media production processes. A *media production process* can be complex or basic. A *complex process* is composed of several basic or complex processes. A *basic process* is represented as a unit that cannot be decomposed into other processes. Each media production process is defined by:

- Input that it receives from the real world, such as thoughts of the authors, or user input.
- Input that it receives from other processes, such as existing annotations or captured media.
- Process and/or real-world artifacts it produces.
- Actors that it involves, such as editor, operator, or designer. An actor can also be some other system or a piece of software.

The output of each process can be the input for other processes, while some processes, such as premeditation, receive input only from the real world. An artifact produced by processes can be atomic, or composite. An atomic artifact defines an artifact that, for a given level of abstraction, cannot be decomposed. We define an *atomic artifact* as:

- Media asset,[5] such as captured video or an image. Usually it contains "raw" data recorded with some sensor technology, such as a camera or a microphone, but it can also be a product of editing or drawing programs, or the result of transformation of existing media assets.
- Annotation,[6] any pertinent information, whether denotative or connotative, such as a description of what is

[5] This is equivalent to the Dexter [7] and Amsterdam Hypermedia Model (AHM) content of the atomic component, http://www.cwi.nl/~lynda/thesis/A1.pdf.

[6] This is equivalent to the Dexter [7] and AHM attributes of the atomic component, http://www.cwi.nl/~lynda/thesis/A1.pdf.

Table 1 UML stereotypes used to describe processes of media production

Name	Type	Description
<<Process>>	Class stereotype	Describes a process.
<<Process artifact>>	Class stereotype	Describes any process artifact.
<<Media asset>>	Class stereotype	Describes a media asset as a basic artifact.
<<Annotation>>	Class stereotype	Describes an annotation as a basic artifact.
<<Composite artifact>>	Class stereotype	Describes a composite artifact.
<<Process actor>>	Class stereotype	Describes a process actor.
<<External world artifact>>	Class stereotype	Describe entities from external world.
<<Input>>	Association stereotype	Connects processes with input artifacts.
<<Output>>	Association stereotype	Connects process with output artifacts.

represented in a media asset, or how the media asset is created. The subject of the annotation can be any process artifact (or part of the artifact identified by an anchor [7, 8]), including other annotations. Optionally, an annotation could be based on terms defined by some schema, e.g. in MPEG-7 [13] or OWL [18]. We make no restrictions or assumptions on the semantics and format of an annotation.

A *composite artifact* is composed of other atomic or composite artifacts. We specify no further semantics of the composition.[7] Note that the terms media asset and annotation represent roles that a particular information object plays for a process. For different processes the same information object may have different roles.

4.2 UML extensions for describing canonical processes

We specify a UML profile where we introduce several UML extensions based on the proposed metamodel. With these extensions, we can describe processes of media production at different levels of abstraction, with various levels of detail. UML includes a formal extension mechanism to allow practitioners to extend its semantics. The mechanism allows us to define stereotypes, tagged values and constraints that can be applied to model elements.[8] In this paper, we describe only the stereotypes we use. Table 1 shows some of introduced UML class and association stereotypes. We use these stereotypes to describe the canonical processes in the next section.

[7] The AHM specifies two structural compositions, temporal and atemporal, which describe presentation oriented composition, http://www.cwi.nl/~lynda/thesis/A1.pdf.

[8] A stereotype is an adornment that allows us to define a new semantic meaning for a modeling element. Tagged values are key value pairs that can be associated with a modeling element that allow us to "tag" any value onto a modeling element. Constraints are rules that define the well-formedness of a model. They can be expressed as free-form text or with the more formal Object Constraint Language *OCL*.

5 Canonical processes of media production

Based on an examination of existing multimedia systems, we have identified nine canonical processes of media production. Every process introduced into our model has at least several instances in existing systems. Our model, therefore, does not contain processes that are specific for particular systems. In the following sections we describe in detail these nine processes:

- *Premeditate*, where initial ideas about media production are established.
- *Create media asset*, where media assets are captured, generated or transformed.
- *Annotate*, where annotations are associated with media assets.
- *Package*, where process artifacts are logically and physically packed.
- *Query*, where a user retrieves a set of process artifacts.
- *Construct message*, where an author specifies the message they wish to convey.
- *Organise*, where process artifacts are organised according to the message.
- *Publish*, where final content and user interface is created.
- *Distribute*, where final interaction between end-users and produced media occurs.

For each process, we give a detailed explanation and state its inputs, outputs and involved actors. A preliminary, less formal, diagram can be found at the Dagstuhl web site.[9] While we give a name to each of the processes, these are meant to be used in a very broad sense. The textual description of each one specifies the breadth of the process we wish to express.

[9] http://www.dagstuhl.de/files/Proceedings/05/05091/05091.PiersolKurt1.Slides.pdf.

In this paper we describe only basic processes. Real-life systems, described in the companion system papers, often implement composite processes that combine several canonical processes.

5.1 Premeditate

Any media creation occurs because someone has made a decision to embark on the process of creating—whether it be image capture with a personal photo camera, drawing in a drawing tool, professional news video, an expensive Hollywood film or a security video in a public transport system. In all cases there has been premeditation and a decision as to when, how and for how long creation should take place.

In all these cases what is recorded is not value-free. A decision has been made to take a picture of this subject, conduct an interview with this person, make this take of the chase scene or position the security camera in this corner. Already there are many semantics that are implicitly present. Who is the "owner" of the media to be created? Why is the media being created? Why has this location/background been chosen? Whatever this information is, it should be possible to collect it and preserve it and be able to attach it to the media that is to be created. For this we need to preserve the appropriate information that can, at some later stage, be associated with one or more corresponding media assets.

Figure 2 shows a UML class diagram of the premeditate process described in terms of the metamodel. The input to this process are ideas, decisions, and artifacts from outside the system. The output is a set of premeditate artifacts, which are typically annotations.

5.2 Create media asset

After a process of premeditation, however short or long, at some point there is a moment of media asset creation. Some device, for example, is used to collect images or sound for a period of time, be it photo or video camera, scanner, sound recorder, heart-rate monitor, MRI etc.

Note that in this process, we do not restrict creation of a media asset to only newly recorded information. Media assets can also be created in other ways. For example, images can be created with image editing programs or generated by transforming one or more existing images. The essence is that a media asset comes into existence, we are not interested in the method of creation per se. If the method is considered as significant, however, then this information should be recorded as part of the annotation.

Figure 3 shows a UML class diagram of the create media asset process described in terms of the metamodel. The input to the capture process is a collection of annotations, for example, information available from the premeditation process, and/or the message construction process. As a result of the create media asset process we have a new media asset.

5.3 Annotate

The annotate process allows extra information to be associated with any existing process artifact. The term annotation is often used to denote a single human user adding metadata to facilitate search. Here we view annotation as the broader process of adding more easily machine-processable descriptions of the artifact.

The annotations need not be explicitly assigned by a user, but may be assigned by an underlying system, for example by supplying a media asset as input to a feature analysis algorithm and using the extracted result to annotate the media asset. We make no distinction whether annotations are selected from an existing vocabulary or machine generated. If deemed relevant, the identity of the human assigner or the algorithm can be recorded in the annotation [1].

We do not prescribe the form of annotations, but require that they can be created and associated with one or more artifacts. We also do not impose limitations on the structure of annotations, due to the high diversity of annotation formats in practice. In most semantically-rich systems, however, the structure of an annotation may include a reference to a vocabulary being used, one of the terms from the vocabulary plus a value describing the media asset.

The annotation can refer to any artifact as a whole, but the annotation could also be more specific. In this case, an anchor mechanism is needed to refer to the part of the media asset

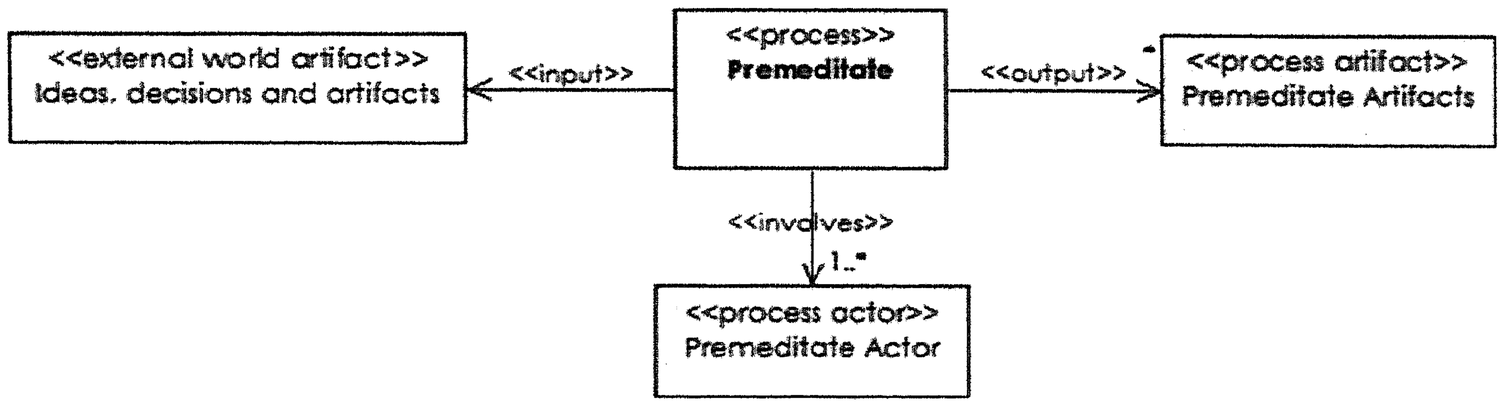

Fig. 2 A class diagram describing the premeditation process

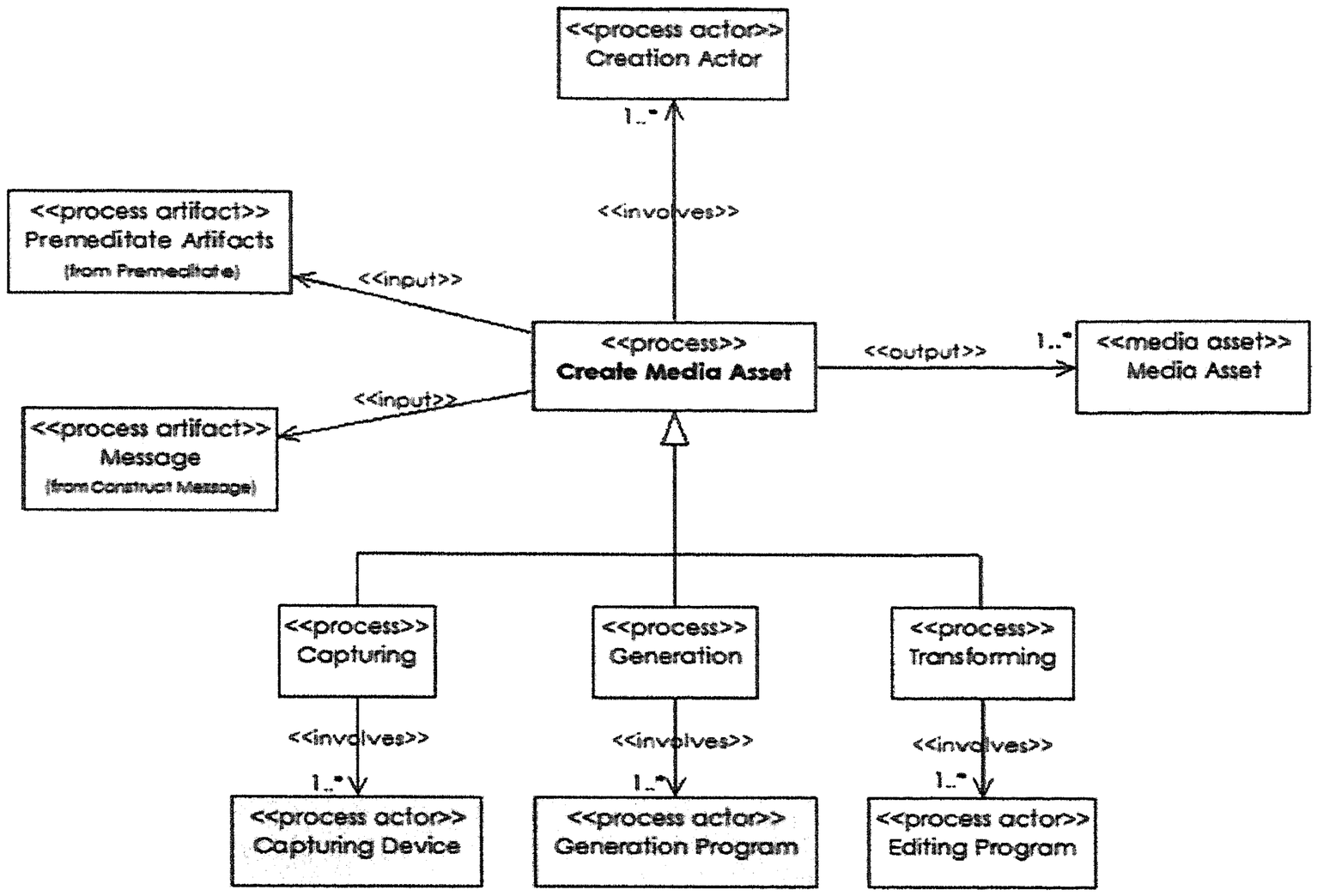

Fig. 3 A class diagram describing the media asset process

to which the annotation applies [7]. An anchor consists of a media independent means of referring to a part of the media asset and a media-dependent anchor value that specifies a part of the media asset. For example, for an image this could be an area, for an object in a film a time-dependent description of an area of the image. For further discussion on anchor specifications see [8] p. 53.

Figure 4 shows a UML class diagram of the annotate process described in terms of the metamodel. The input to this process is a process artifact plus an annotation supplied by a human or system. The output is a process artifact including the extra annotation. A semantic annotate process is a specialization of annotate process, where the annotation itself may contain a reference to a vocabulary, element and/or value used in the annotation (Fig. 5).

5.4 Package

The process of packaging provides a message independent grouping of artifacts. The output of this process is aimed at authors and developers, to help them to maintain process artifacts, and is unrelated to the final presentation organization. This process, for example, can assign a group of related media items and annotations an identity so that it can be retrieved as a unit. One of the simplest forms of packaging is putting related files in one directory, where the directory path provides an identity for the package.

We make a distinction between physical and logical packaging, where physical packaging reflects the organization of units in a database or file system, and logical packaging defines logical relations among items. For example, a SMIL presentation is logically one presentation unit, but links media components physically packaged in many files in a distributed environment. On the other hand, a multimedia database can physically be packaged in one file, but contain many logical units.

Figure 6 shows a UML class diagram of the package process described in terms of the metamodel. The input to this phase is a set of process artifacts and optional input given by user, such as an identifier for the created package. The output is a multimedia package that physically or logically groups input artifacts into one unit.

5.5 Query

The query process selects a number of artifacts from a repository of artifacts. Up until now the processes we describe concentrate on creating, storing and describing primarily media assets. These are needed for populating a media repository. Note that our definition of media repository does not necessarily imply the existence of a complex storage infrastructure, but we assume that systems have a repository where they keep media assets and other artifacts, in the most simple case a hierarchically organised file directory structure. Once there is a repository of artifacts it can be queried for

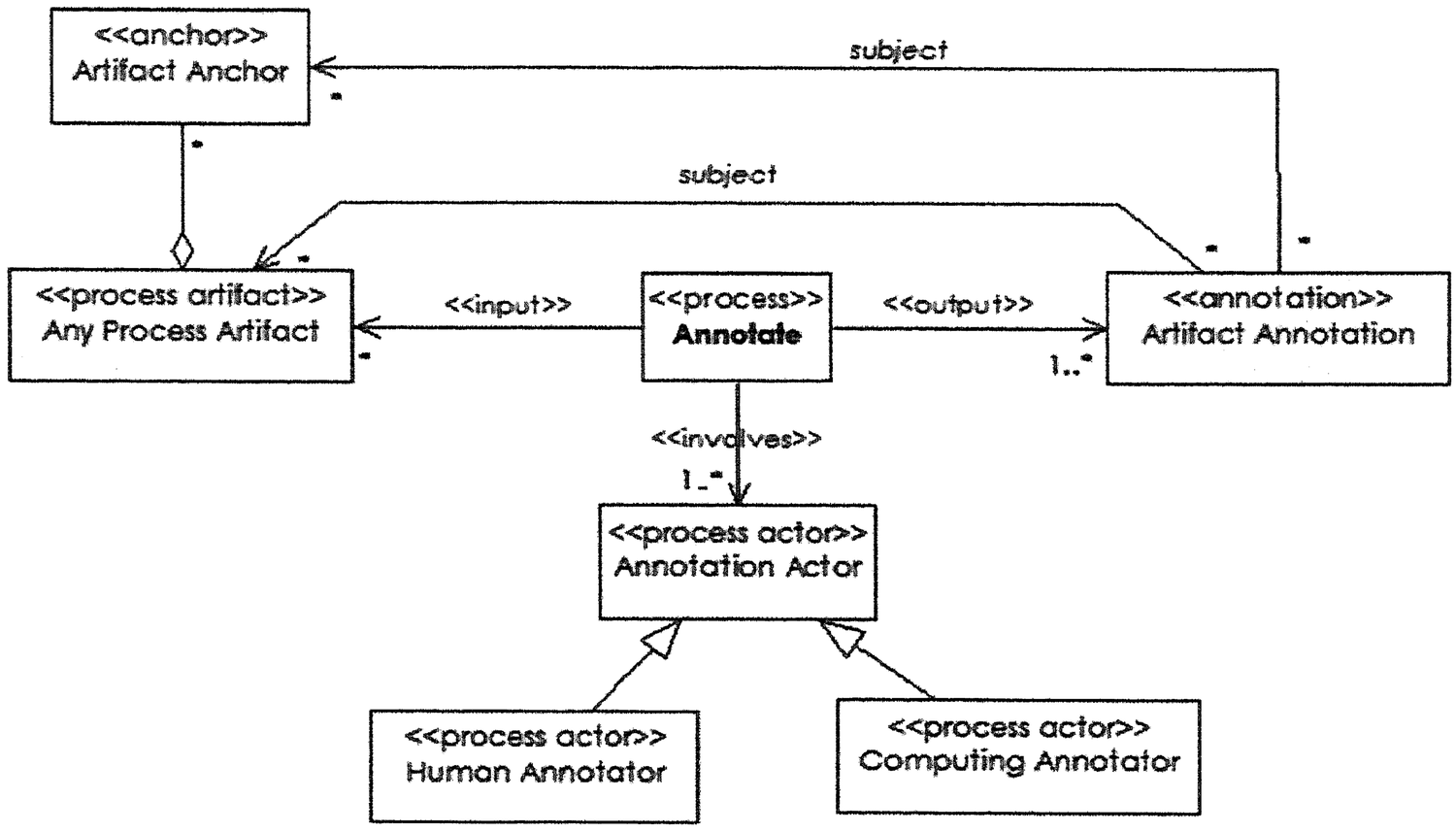

Fig. 4 A class diagram describing the annotation process

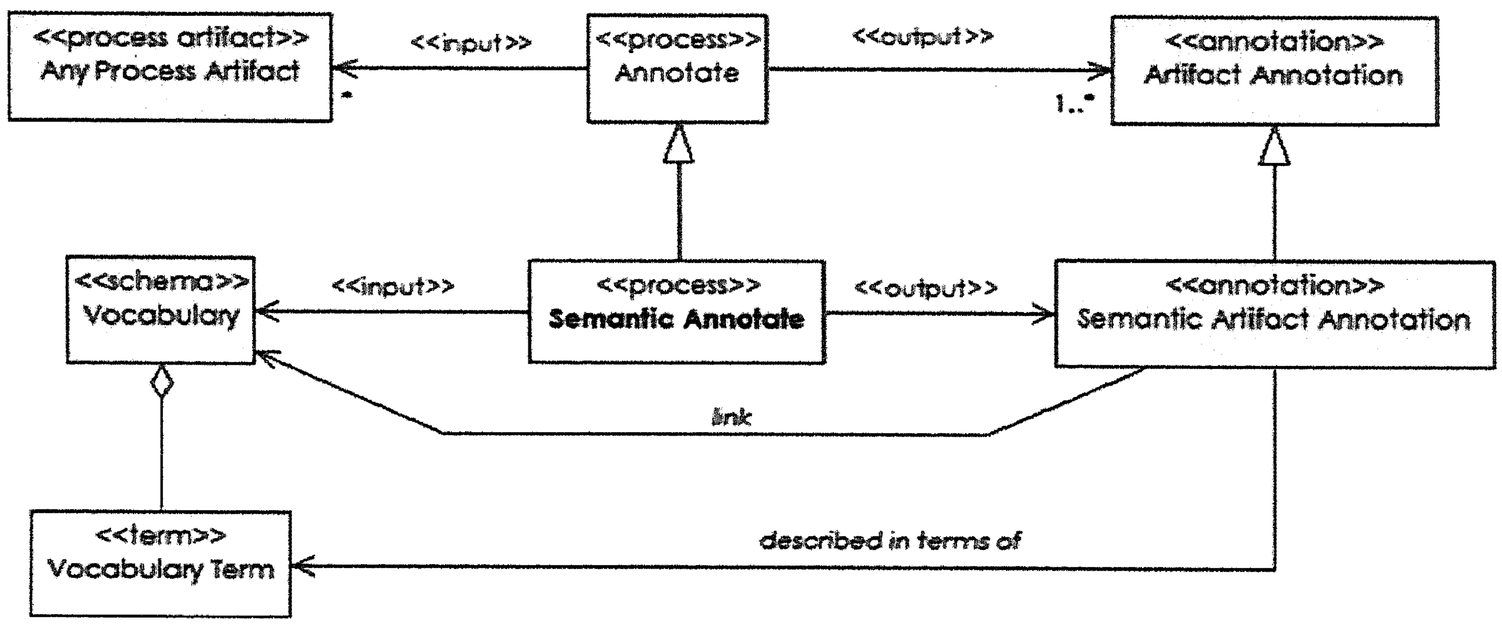

Fig. 5 A class diagram describing the semantic annotation process

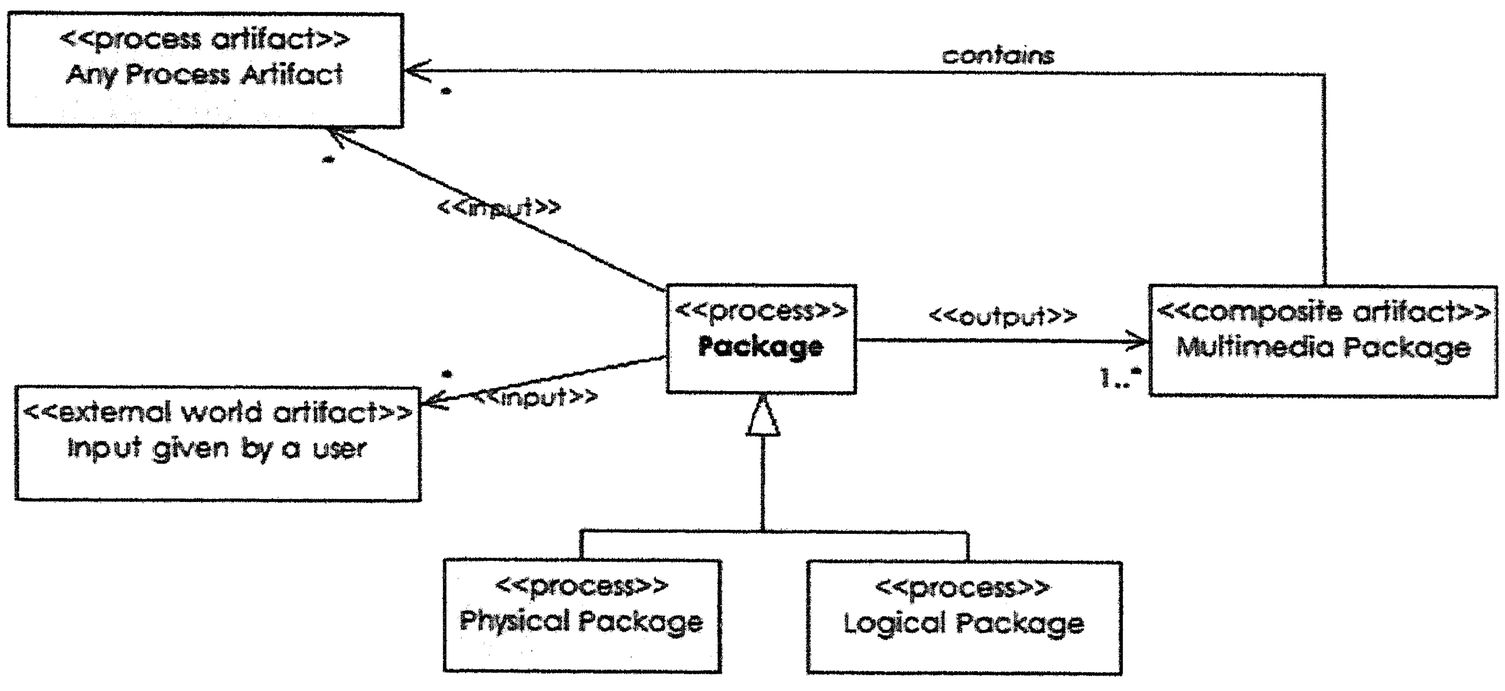

Fig. 6 A class diagram describing the package process

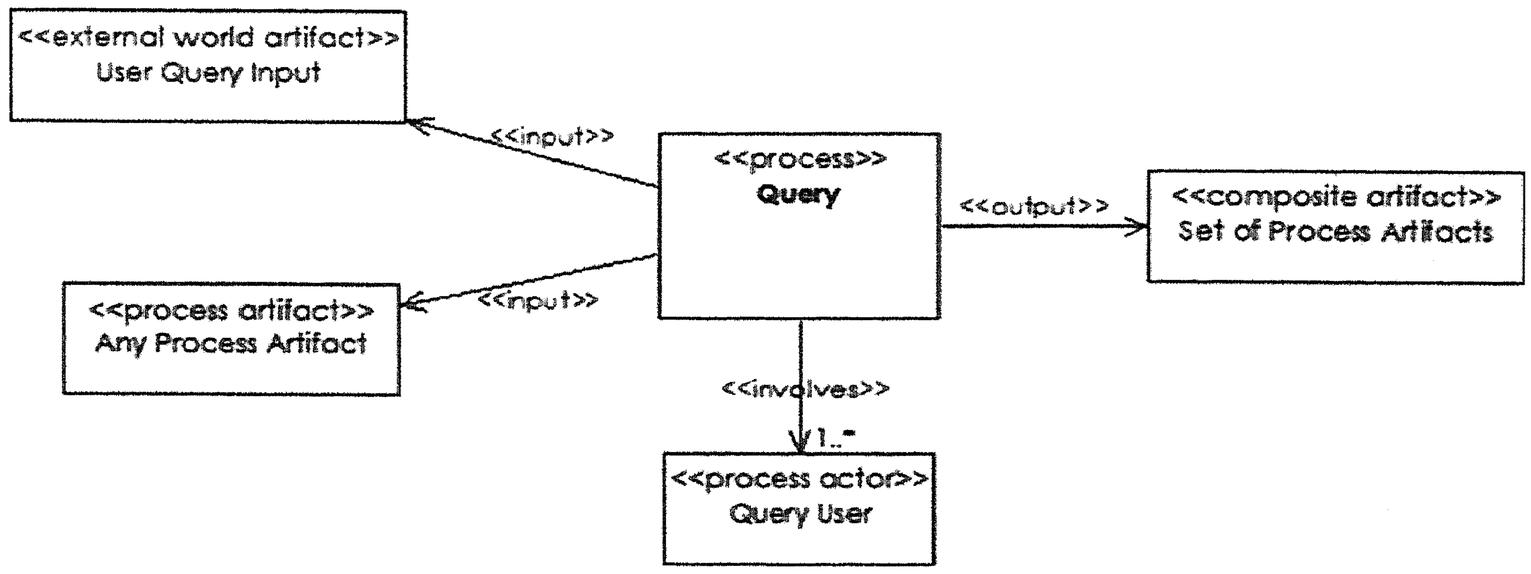

Fig. 7 A class diagram describing the query process

components whose associated media assets correspond to specified properties.

We do not wish to use a narrow definition of the term "query", but intend to include any interface that allows the artifacts to be searched, using query languages of choice or (generated) browsing interfaces that allow exploration of the content of the archive. It is worth noting that many systems that provide advanced query interfaces, also provide support for other processes. For example, browser interfaces can, in addition to a simple query interface, also organise intermediate results to present them to a user for feedback, and create temporary presentations that are then published and distributed to the user.

A query of the system may be in terms of media assets, or in terms of the annotations stored with the media assets. A query needs to specify (indirectly) the annotation(s) being used, and includes techniques such as query by example. The mechanisms themselves are not important for the identification of the process.

Figure 7 shows a UML class diagram of the query process described in terms of the metamodel. The input to the query process is a set of process artifacts plus a specification of a subset of these. The output is a (possibly empty) set of identified media components corresponding to the specification. The output is often not a set of media assets, but a structural asset that includes references to the media components that contain links to process artifacts.

5.6 Construct message

A presentation of media assets, such as a film or an anatomy book, is created because a human author wishes to communicate something to a viewer or reader. Constructing the message which lies behind the presentation is most often carried out by one or more human authors. When a viewer watches a film or a reader reads a book then some part of the intended original message of the author will hopefully be communicated. In order to give different processes access to the underlying intent, we include an explicit process which brings a processable form of the message into the system. Just as capturing a media asset is input into the system, so is the specification of the message an author wishes to convey.

In some sense, there is no input into the construct message process. However, the real input is the collection of knowledge and experience in the author her/himself. The output of the process is a description of the intended message. For example, a multimedia sketch system, such as described in [2], allows an author to gradually build up a description of the message. For the message to be machine processable the underlying semantics need to be expressed explicitly.

A query implicitly specifies a message, albeit a simple one, that an author may want to convey, since otherwise the author would not have been interested in finding those media assets. The query is, however, not itself the message that the author wishes to convey.

In general, we give no recommendation in this paper for the expression of the semantics of the message. We expect that it contains information regarding the domain and how this is to be communicated to the user, but we do not assign anything more than a means of identifying a particular message.

Figure 8 shows a UML class diagram of the construct message process described in terms of the metamodel. The input to this process are ideas and decisions from the external world, and the output is a message.

5.7 Organise

While querying allows the selection of a subset of media assets, it imposes no explicit structure on the results of one or more queries. The process of organisation create a document structure that groups and orders the selected media assets for presentation to a user. How this process occurs is, again, not relevant, but may include, for example, the linear relevance orderings provided by most information retrieval systems. It also includes the complex human process of producing

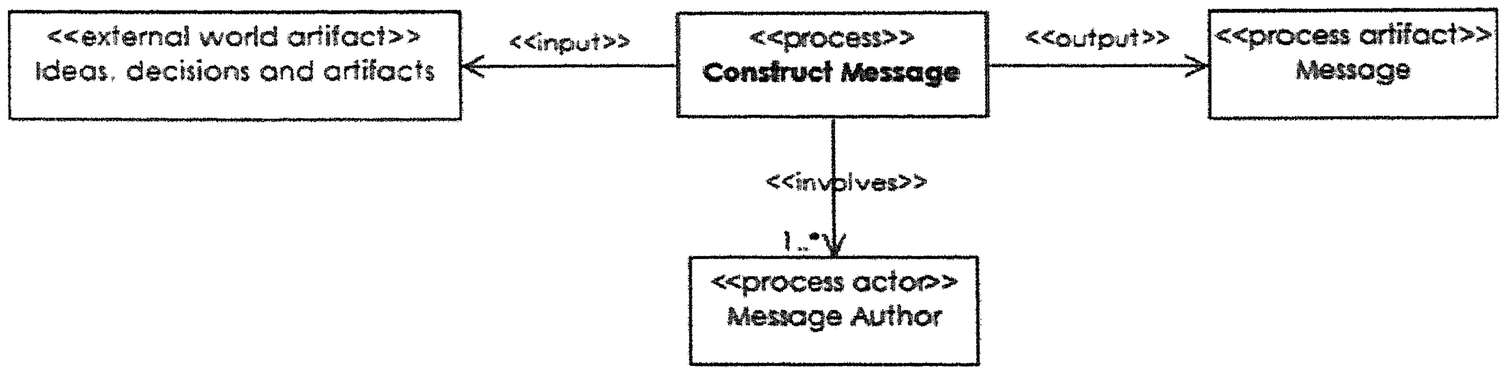

Fig. 8 A class diagram describing the construct message process

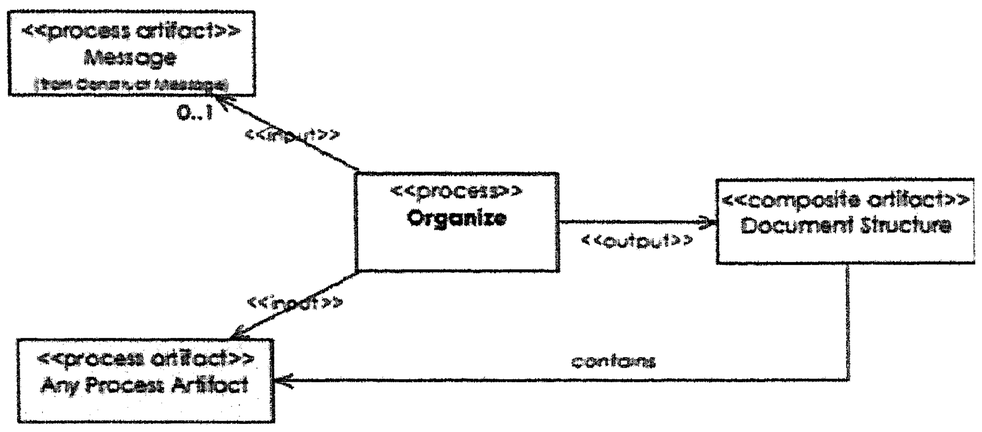

Fig. 9 A class diagram describing the organise process

a linear collection of slides for a talk; creating multimedia documents for the web; ordering shots in a film; or even producing a static 2-dimensional poster.

The process of organisation is guided by the message (the output of the construct message process). The organization depends on the message and how the annotations of the process artifacts relate to the message. For example, annotations concerning dates could be used to order assets temporally. The resulting document structure may reflect the underlying domain semantics, for example a medical or cultural heritage application, but is not required to. The structure may be colour-based or rhythm based, if the main purpose of the message is, for example, aesthetic rather than informative.

In the arena of text documents, the document structure resulting from organisation is predominantly a hierarchical structure of headings and subheadings. The document structure of a film is a hierarchical collection of shots. For more interactive applications, the document structure includes links from one "scene" to another. In a SMIL [17] document, for example, `par` and `seq` elements form the hierarchical backbone of the document structure we refer to here.

Figure 9 shows a UML class diagram of the organise process described in terms of the metamodel. The input to the organise process is the message plus one or more process artifacts. The output is the document structure, which includes the media assets associated with the substructures.

5.8 Publish

The output of the organise process is a prototypical presentation that can be communicated to an end-user. This serves as input to the publication process which selects appropriate parts of the document structure to present to the end-user. The publication process takes a generic document structure and makes refinements before sending the actual bits to the user. These may include selecting preferred modalities for the user and displayable by the user's device. The resulting presentation can be linear (non-interactive, e.g. a movie) or non-linear (interactive, e.g. web presentation).

Publication can be seen as taking the document structure from the internal set of processes and converting it (with potential loss of information) for external use. Annotations may be added to describe the published document. For example, the device or bandwidth for which the publication is destined. Annotations and alternative media assets may be removed to protect internal information or just reduce the size of the data destined for the user.

Figure 10 shows a UML class diagram of the publish process described in terms of the metamodel. The input to the publication process is a set of process artifacts and a message that guides the organization, and the output is a published document that organises input process artifacts according to the message.

5.9 Distribute

Created content has to be, synchronously or asynchronously, transmitted to the end-user. This final process involves some form of user interaction and requires interaction devices, while transmission of multimedia data to the user device goes through some of the transmission channels including the internet (streamed or file-based) non-networked medium (such as a CD-ROM or DVD) or even analog recording media (for example, film).

It is important to note that the term "distribution" in our model has a much broader meaning than in classical linear production of media. It can also be used to describe interactive non-linear productions, such as games or other interactive presentations. The resulting system would implement a complex process including query, organise and publish processes in addition to distribution (end-user interaction). For example, some systems can have a final presentation where the story line depends on user feedback. In this case,

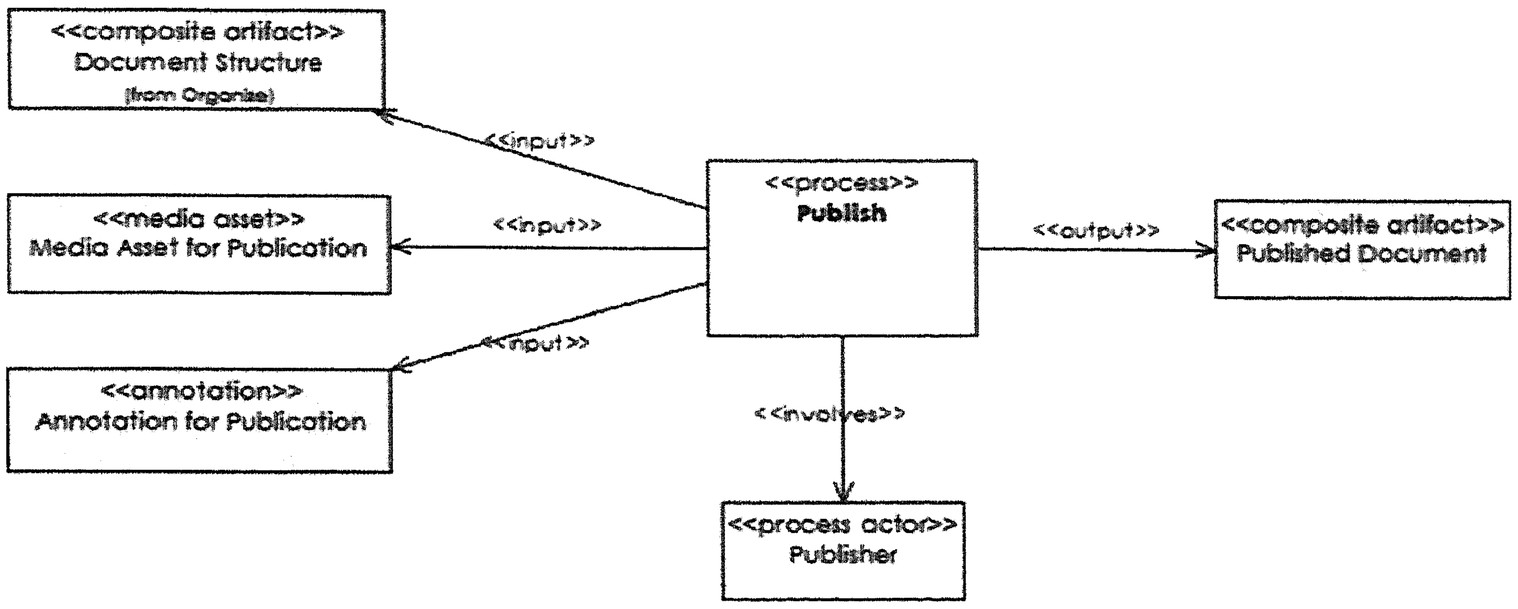

Fig. 10 A class diagram describing the publish process

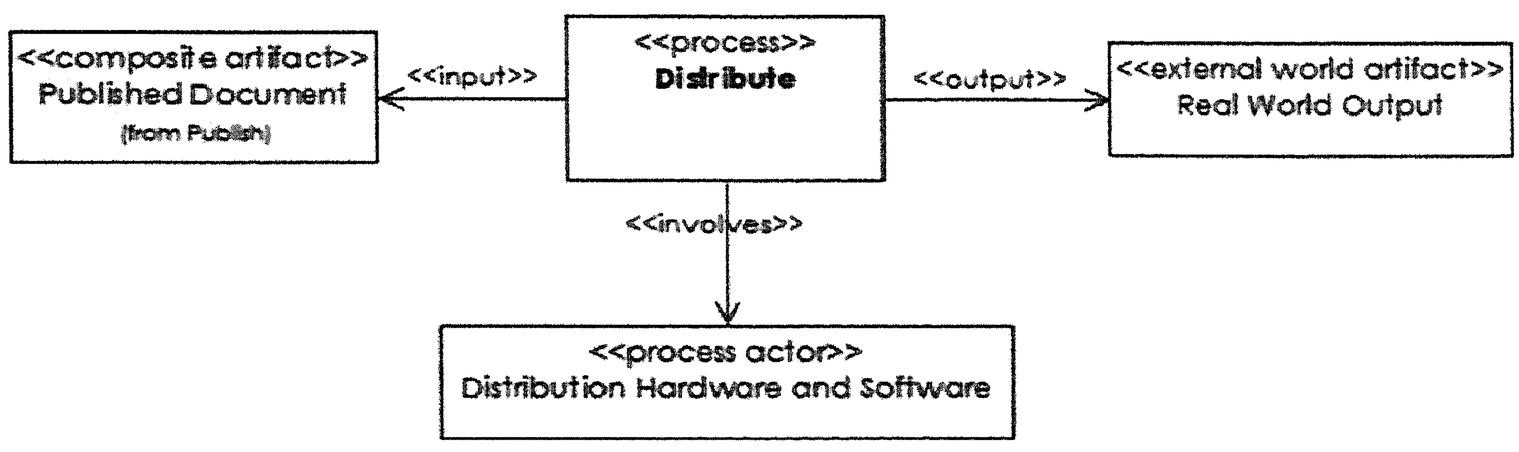

Fig. 11 A class diagram describing a distribution process

the presentation system would include the canonical processes query (to select next part of the story), organise (to organise selected items coherently), publish (to create internal document ready for presentation) and distribute (to present and expose the control interface to user).

Figure 11 shows a UML class diagram of the distribute process described in terms of the metamodel. The input to the process is a process artifact representing a published document. This process involves appropriate software/hardware for displaying/playing or interacting with the media assets. The output is the real-time display or projection of the media assets to the end-user, or the creation of a physical carrier, such as DVD, film or photobook.

6 Discussion

The canonical processes represent a distillation of our discussions on formulating them to help the multimedia community communicate about their systems at both a descriptive and computational level. During the course of our discussions we encountered questions that arose from multiple authors. We discuss these with the aim of clarifying potential misunderstandings.

We also discuss how our descriptions of canonical processes fit in with more formal descriptions, in particular foundational ontologies, and what the boundaries of our achievements are—in particular we deliberately do not describe the form of the annotations. We see this rather as parallel, but closely related, work.

The section gives an overview of advantages of specifying the canonical processes explicitly, some frequently asked questions and then how the process descriptions can be linked to more formal representations.

6.1 Benefits of descriptions in terms of the canonical processes

Most of the authors in this special issue discussed the potential benefits of canonical processes. Here we summarise the main observations from the authors of the companion papers.

6.1.1 Identifying omitted functionality

Identifying and aligning the processes implemented by a particular system with the canonical processes enables a comparison of system functionality. This allows implementors to identify functionality not currently implemented and make informed decisions as to whether to implement the missing functionality in their own system or search for an existing, compatible component. For example, in their paper, Hyowon Lee et al. noted that by aligning their system processes with

the canonical processes, they were able to better understand the process cycles of their system in the context of the more generalised process cycles of existing systems.

6.1.2 Improving interoperability

Comparing an existing system with the canonical processes also improves interoperability. Defining both application and framework functionalities in terms of canonical processes allows the decoupling of the actual abstraction steps and visualisation and avoids hard-wiring design decisions and context information.

The canonical processes also help to envisage future scenarios where some of the processes within a system could be exchanged with those from another.

6.1.3 Using annotations in different processes

Much work in the multimedia community is geared to analysing visual or audio media and extracting higher-level representations of the features found. These are often geared to facilitating the query process. By identifying a number of other processes, we are now able to discuss the role of annotations in each of these. For example, the organise process can use semantics from both the message and the results retrieved by a query for more user-friendly ways of grouping and ordering material. The publish process can also benefit from information about the media characteristics (such as bandwidth required, or aspect ratio) for selecting items for a specific platform.

6.2 Frequently asked questions

While preparing the special issue, we had several discussions with authors about how to describe particular elements of their systems in terms of the canonical processes. Here we present the most frequently asked questions.

6.2.1 Complex processes

The processes described in our model are the basic ones, and the real-world processes are usually composed from two or more of them. We encountered requests from authors of companion papers to include more complex processes in the list of canonical processes. The resolution was to keep in the model only the basic processes, and to better explain that the canonical processes constitute a model of building blocks, useful for discussing within the community, rather than a prescription for a system architecture. Any particular application may implement functionality that includes multiple canonical processes. Our goal is to make developers aware that this is what they are doing, and if there are intermediate results within the system that these be made available to other modules or systems.

6.2.2 Interaction

The relationship between interaction and the canonical processes is not clear in the first instance. While the canonical processes describe a query process and a distribute process, there is no process termed "interaction" or "user feedback". Our view is that the process of interaction takes place within the context of an application, which may implement one or more of the canonical processes, and that interaction is, and should remain, a complex process including multiple canonical processes. The new media art application discussed in [11] is an example that illustrates this. Users moving through an interactive art installation provide input that is used as a precursor to a query process. This in turn leads to processing within the system, which then takes the results from the query process and, via an organise process, distributes the result back to the user. The interaction cycle takes place on sub-second timescales, during which different canonical processes are invoked.

The process of interaction itself can also be used to contribute to an annotation process. For example, if an application records that many people select the same media asset many times, then this could be captured as an annotation and used to reorder the results on a subsequent organisation of the same query.

6.2.3 Complex artifacts and annotations can be annotated

The composite artifact representing a complex presentation resulting from the organise process can be fed back into any process that accepts a composite artifact as input. This can be used to associate annotations with the complex artifact, such as its author or a version number. For example, a film script is the result of a long premeditation process with a complex structure whose semantics can only be captured with difficulty. The complete script represented by an atomic artifact, however, is a process artifact with which annotations can be associated. An annotation is itself an atomic artifact, hence a process artifact with which annotations can be associated. For example, the author who assigned the annotation.

6.2.4 Further specification of canonical processes

While the canonical processes provide a small set of identified processes common across systems, some authors felt the need to specify more details of a number of the processes for particular situations. While we see the need for agreement on a further level of detail in some of the processes, our goal for this special issue is to obtain community wide agreement on the identification of a small number of processes. A further

stage of evolution could be to encourage different communities to agree upon more detailed specifications in specialised areas. We see this as similar to the community process of describing agreed-upon XML schemas or domain-specific ontologies.

6.3 Canonical processes and workflows

There are several established workflows in multimedia production[10,11]. The relationship of the canonical processes of media production to workflow patterns is that we propose basic building blocks that can be coordinated using different workflow patterns. In other words, we are not interested in defining a workflow of media production, but only the building blocks where workflow patterns can play a role of a higher-level coordination language for our canonical processes. We do not propose or limit usage of any of the workflow patterns.[12]

6.4 Towards a more rigorous formalisation of the model

The focus of this special issue is not on rigorous formalisation of canonical processes, but on their identification and mapping to real systems. We provide a UML description of the processes. While a text-only description is insufficiently precise, we do not want to exclude a large audience by using unfamiliar formal languages. We do, however, see the need for future work to link our current descriptions into a higher-level ontology, and to specify the structure of annotations more precisely.

6.4.1 Relationship to foundational ontologies

A foundational ontology, or upper-level, ontology, describes a domain independent vocabulary that explicitly includes formal definitions of foundational categories, such as processes or physical objects. It provides a hierarchy of entities and associated rules (both theorems and regulations) that describe general entities that apply across domains. Describing the processes using a foundational ontology, such as DOLCE [6], provides a solid modelling basis and enables interoperability with other models. DOLCE provides description templates (patterns) for the specification of particular situations and information objects. Figure 12 illustrates the relation of the model of canonical processes of media production and specific models and systems, and with upper ontologies.

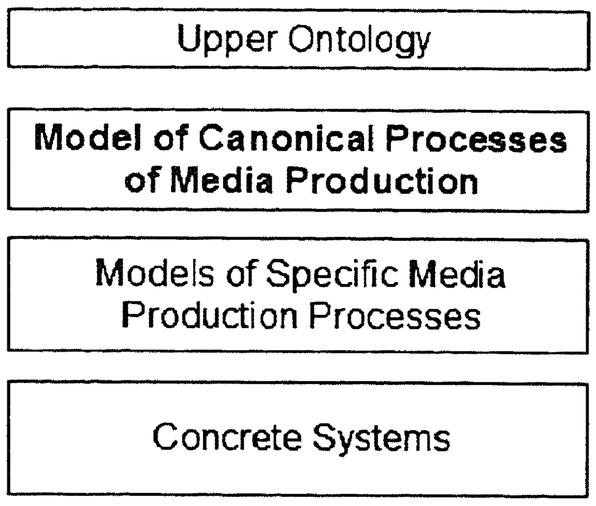

Fig. 12 Relation of the model of canonical processes of media production with upper ontologies and specific models and systems

A next step in the specification of the ontologies would be to express these as specialisations of the DOLCE model, in particular, the DOLCE situations.

6.4.2 Semantics of annotations

Although out of the scope of this special issue, our effort to identify canonical processes is part of a broader effort to create a rigorous formal description of a high quality multimedia annotation ontology compatible with existing (semantic) web technologies. In particular, we are involved with work on specifying the structure of complex semantic annotations of non-textual data. This has resulted in COMM—a core ontology for multimedia[13] [1] based on the MPEG-7 standard [9] and expressed in terms of OWL [18].

7 Conclusion

In this paper we identify a number of canonical processes that are fundamental to different applications of multimedia. This model forms an initial step towards the definition of data structures, which could be accessed and produced by systems created within the multimedia community. In the rest of this special issue we present a number of companion papers that describe existing systems in terms of the canonical processes.

While the canonical processes have been developed within the multimedia community, the semantic web community is also investigating ways to resolve the problem of keeping data and its associated annotations together in everyday desktop environments—a so-called semantic clipboard. Initial work on this has been carried out [16], and we see the canonical processes as being helpful in providing a framework for researchers to investigate particular solutions and adapt solutions from neighbouring fields.

[10] http://www.sgi.com/pdfs/3477.pdf.

[11] http://www.davidausterberry.com/dam.html.

[12] The site http://www.workflowpatterns.com/ provides an overview of literature and systematisation of the workflow patterns.

[13] http://multimedia.semanticweb.org/COMM/.

The companion papers illustrate the breadth of applicability of the processes, but are not intended to restrict its scope. Many other application areas of semantically annotated media would potentially benefit from their consideration during the design process. For example, universal design,[14] advocates *the design of products, services and environments to be usable by as many people as possible regardless of age, ability or circumstance*, is an area that could benefit to a great extent from more explicit management of multimedia and associated annotations. For example, annotations from a premeditate process, such as a script, can provide a basis for creating audio descriptions of movies for blind users, or textual descriptions of audio effects for deaf users. Information about organization of the content can make easier navigation for blind users.

While we have identified a set of canonical processes for semantic multimedia, these have not been developed in isolation. As described in the discussion section, COMM provides a model for describing the annotations mentioned, but not specified, in the canonical processes. Other relevant work on semantically annotating images on the web has been carried out by the W3C Incubator Group on Multimedia Semantics.[15] We see these different threads stimulating and complementing each other towards open web-based data structures and software components for describing and sharing semantically annotated media assets among different platforms.

Acknowledgments This paper was inspired by Dagstuhl meeting 05091 "Multimedia Research—where do we need to go tomorrow" organised by Susanne Boll, Ramesh Jain, Tat-Seng Chua and Navenka Dimitrova. Members of the "Multimedia for Human Communication" working group were: Lynda Hardman, Brigitte Kerhervé, Stephen Kimani, Frank Nack, Kurt Piersol, Nicu Sebe, and Freddy Snijder. We would also like to thank all the authors of the companion system papers, in particular Susanne Boll and Ansgar Scherp, for their feedback on the model. In addition we would like to thank Raphaël Troncy and Jacco van Ossenbruggen for their valuable feedback on this paper. Parts of this research were funded by the Dutch national BSIK MultimediaN e-Culture, ToKeN2000 CHIME and European ITEA Passepartout and IST K-Space projects.

References

1. Arndt, R., Troncy, R., Staab, S., Hardman, L., Vacura, M.: COMM: Designing a well-founded multimedia ontology for the Web. In The Semantic Web—ISWC/ASWC 2007, vol. 4825 of Lecture Notes in Computer Science, pages 30–43, Busan, Korea, 11–15 November 2007
2. Bailey, B.P., Konstan, J.A., Carlis, J.V.: Supporting multimedia designers: towards more effective design tools. In: Proceedings of Multimedia Modeling: Modeling Mutlimedia Information and Systems (MMM2001), pp. 267–286. Centrum voor Wiskunde en Informatica (CWI) (2001)
3. Bocconi, S., Nack, F., Hardman, L.: Using rhetorical annotations for generating video documentaries. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME) (2005)
4. Davis, M.: Active capture: integrating human–computer interaction and computer vision/audition to automate media capture. In: ICME '03, Proceedings, pp. 185–188 (2003)
5. Dorai, C., Venkatesh, S.: Bridging the semantic gap in content management systems—computational media aesthetics. In: Dorai, C., Venkatesh, S. (eds) Media computing—computational media aesthetics, pp. 1–9. Kluwer, Dordrecht (2002)
6. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A.,Schneider, L.: Sweetening ontologies with DOLCE. In: EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, pp. 166–181. Siguenza, Spain, 01–04 October 2002. Springer, Heidelberg
7. Halasz, F., Schwartz, M.: Grønbæck, K., Trigg, R. (eds) The dexter hypertext reference model. Commun ACM **37**(2), 30–39 (1994)
8. Hardman, L.: Modelling and Authoring Hypermedia Documents. PhD thesis, University of Amsterdam, 1998. ISBN: 90-74795-93-5, also available at http://www.cwi.nl/~lynda/thesis/
9. ISO/IEC: Overview of the MPEG-7 Standard (version 6.0). ISO/IEC JTC1/SC29/WG11/N4980, Pattaya (2001)
10. Jain, R.: Experiential computing. Commun ACM **46**(7), 48–55 (2003)
11. Kerhervé, B., Ouali, A., Landon, P.: Design and production of new media artworks. In: Proceedings of the ACM Workshop on Multimedia for Human Communication—From Capture to Convey (MHC 05) (2005)
12. Kosch, H., Böszörményi, L., Döller, M., Libsie, M., Schojer, P., Kofler, A.: The life cycle of multimedia metadata. IEEE MultiMedia, (January–March) pp. 80–86 (2005)
13. Nack, F., Lindsay, A.T.: Everything you wanted to know about MPEG-7: part 1. IEEE multiMedia, pp. 65–77 (1999)
14. Nack, F., Putz, W.: Designing annotation before it's needed. In: Proceedings of the 9th ACM International Conference on Multimedia, pp. 251–260. Ottawa, Ontario, Canada, 30 September–5 October, 2001
15. Pereira, F., Vetro, A., Sikora, T.: Multimedia retrieval and delivery: essential metadata challenges and standards. Proc IEEE **96**(4), 721–744 (2008)
16. Reif, G., Morger, M., Gall, H.: Semantic clipboard—semantically enriched data exchange between desktop applications. In: SemDesk2006 Workshop at the 5th International Semantic Web Conference (ISWC2006), Athens, GA, USA, November (2006)
17. W3C.: Synchronized Multimedia Integration Language (SMIL 2.0) Specification. In: Cohen A (ed) W3C Recommendation, 7 August 2001
18. W3C.: Web Ontology Language (OWL)—overview. W3C recommendation, 10 February 2004

[14] http://en.wikipedia.org/wiki/Universal_design.

[15] http://www.w3.org/2005/Incubator/mmsem/.