Relating the primitive hierarchy of the PREMO standard to the standard reference model for intelligent multimedia presentation systems

D.J. Duke [a,*], I. Herman [b,1], T. Rist [c,2], M. Wilson [d,3]

[a] *University of York, Heslington, York YO1 5DD, UK*
[b] *Center for Mathematics and Computer Sciences (CWI), Kruislaan 413, Amsterdam 1098 SJ, Netherlands*
[c] *German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany*
[d] *Council for the Central Laboratory to the Research Councils (CCLRC), Rutherford Appleton Laboratory, Chilton, Didcot, Oxon OX11 0QX, UK*

Abstract

The need for a suitable classification of media types arises for several reasons when building or comparing multimedia systems. Within an Intelligent Multimedia Presentation Systems (IMMPS), it is necessary to formulate and encode design knowledge for decision making on the appropriate medium in which to present information and for the generation of the presentation. It is also required in order to specify interfaces to and between system components which will be employed to run a generated presentation before the user's eyes. This task is reflected in the Standard Reference Model (SRM, see this volume) for IMMPS by the Presentation Display Layer. However, the SRM does not instantiate this layer in detail, but instead refers to the Presentation Environment for Multimedia Objects (PREMO) ISO/IEC standard which provides a reference model for a presentation runtime environment for multimedia. PREMO already contains a set of basic structures, the so-called PREMO Primitive Hierarchy, to describe different media types. Thus the question arises, as to how far the PREMO Primitive Hierarchy could serve as a media classification for the SRM in general. In particular, this would support consistency between the design and presentation layers of the SRM if PREMO were used to instantiate the presentation layer. In the current paper, we first point to a number of typical problems with generating classifications of media types. We then provide a brief introduction to PREMO and its Primitive Hierarchy. Finally, the benefits and costs of using the PREMO Primitive Hierarchy for the SRM are discussed. © 1997 Elsevier Science B.V.

Keywords: Multimedia; Multimedia modelling; Knowledge representation; PREMO; Geometric primitives; Standards

1. Introduction

Bordegoni et al. [1] present a Standard Reference Model (SRM) for Intelligent Multimedia Presentation Systems (IMMPS). In outline, this model consists of a layered pipeline to generate presentations, and which can call on knowledge servers to control its decisions. The Content Layer contains a Media

* Corresponding author. E-mail: duke@minster.york.ac.uk
[1] E-mail: ivan@cwi.nl
[2] E-mail: rist@acm.org
[3] E-mail: mdw@inf.rl.ac.uk

Allocation component, and the Design and Realization Layers of this pipeline contain media specific Design and Realization components, each of which can draw on centralized design knowledge from the Design Expert in the Knowledge Server. The SRM reveals the need for a classification of media types for at least two reasons: for the encoding of design knowledge, and for the specification of interfaces through which information on media objects can be exchanged.

The way in which design knowledge is encoded in an IMMPS depends on the internal structure of a system's components which correspond to the SRM'S Media Allocation, Design and Realization components. While early approaches were based on rules or templates, approaches which rely on hierarchical planning, or constraint solving, as well as combinations of these approaches have become popular within the last years (e.g., see the system descriptions in this volume). Accordingly, design knowledge is coded in rules, constraints or planning operators. The captured design information may be based on empirical findings (e.g., see Ref. [2]), or just reflect rules of thumb such as that natural language is better for presenting information containing quantification, negation or temporal information, while graphics are better for displaying spatial information (after Ref. [3]). In all cases, such design information includes a reference to presentation media or specific properties of media. Thus, having a common classification is an important step toward the sharing, exchange, and reuse of design knowledge. The construction of such classifications has always proved to be difficult since there are many often incompatible factors to consider. The SRM itself does not attempt to present such a classification, nor does it offer guidance on the structuring of design knowledge.

The final layer in the SRM pipeline is a Presentation Display Layer which serves as a presentation runtime environment for the constructed presentation. In this layer, the issue of interfaces for the exchange of information on media objects must be addressed. However, the SRM does not instantiate this layer in detail. For an appropriate candidate to instantiate this layer, the SRM refers to the Presentation Environment for Multimedia Objects (PREMO) ISO/IEC standard which is being developed for such environments. This instantiation is not devel-

oped in this paper, but interested readers should consider the overall PREMO standard more fully [4].

The PREMO Standard contains a hierarchy of primitive, representing a system-oriented approach to media classification. This is in contrast to the modality classification of Bernsen [5], which is based on a human-oriented assessment of representational modalities, rather than system considerations. If PREMO were used to instantiate the Presentation Display Layer of the SRM, then it would obviously be consistent to use the PREMO classification in order to simplify the interface to that layer.

In Section 2, we list some of the issues that must be addressed when defining a set of primitive media types. Then we give a brief description of the PREMO approach in Section 3 and, by giving a short overview of the current primitive hierarchy in Section 4, we will show how the designers of PREMO addressed these issues. Finally, in Section 5, the benefits and costs of using the PREMO primitive hierarchy for the SRM are discussed.

2. Defining primitive media types: some issues

Many attempts at the classification of media have been made, spanning many disciplines that include semiotics, psychology, information science, telecommunications, and informatics. Whether they are useful or not in a certain context depends on many issues. The five issues presented in this section are of particular interest with regards to the context of the current paper as they have been addressed by the designers of the PREMO Primitive Hierarchy which is presented later.

2.1. Perspective: human-centered vs. system-centered

Frohlich [6] and Bernsen [5] present classifications based on the human sensory/representational modalities of the user. In contrast, Nigay and Coutaz [7] present an analysis of the interface design space which is centered on the system in order to aid designers to identify the software implications and constraints for developing their systems. Nigay and Coutaz [7] avoid presenting a classification of media types, but like them, the PREMO approach is system-centered and is not structured around the human senses of the user.

The choice of a certain perspective is often a choice between cognitive adequateness on the one hand, and engineering pragmatism on the other hand. Consider, e.g., the choice of a color model in a media hierarchy. The well-known RGB model is based on hardware design, as well as human biological considerations (the human visual system is also based on mixing various primary colors, just as the RGB monitors do). In other words, the RGB model is very well adapted to (computing) systems. On the other hand, the same RGB system is very counterintuitive; anybody, who has ever tried to set a specific color on a display, can witness this. The 'perceptual' counterpart is the HSV system, which is user-oriented, being based on the intuitive appeal of the artist's tint, shade, and tone. The price to pay when using this model is that an extra transformation from the HSV values to the RGB system becomes necessary (to control the underlying hardware), and that the classic color-based algorithms of synthetic graphics (shading, etc.) become more complicated (see Ref. [8]).

With regard to the formulation of design knowledge, a media classification certainly must not ignore the human-centered perspective. To avoid misuse and underuse of media the yardstick of any assessment must involve the human user (see also Ref. [9]). On the other hand, engineering pragmatism has taken into account when striving for a specification of interfaces between layers and components of the SRM.

2.2. Conceptualization: objects vs. attributes and methods

When building a model of a domain, it is not always apparent what should become an object, and what should be considered as an attribute to an object or a relation between objects. In general, one may say, the more abstract the domain, the more difficult the decision. Graph theory even shows that in some cases a structurally equivalent model can be obtained simply through dualization (i.e., objects become relations and vice versa). The conceptualization issue is especially relevant, if one strives for an object-oriented classification such as the PREMO Primitive Hierarchy.

The practice and pragmatics of OO programming can lead to different classifications, too. A classic example is as follows. A fundamental question that must be addressed within any object-oriented graphics or multimedia system concerns the allocation of fundamental behavior, such as transformations and rendering, to object types within an API. Two quite distinct approaches emerge. The first is to attach behavior to the object types that are affected by that behavior. For example, geometric objects and other kinds of presentable media data can be defined with a 'render' method, with the interpretation that such an object can be requested to produce a rendering of itself. Such an approach can be extended to collections of presentable objects, and fits well with the concept of an object as a container for data along with the operations that manipulate that data. The second approach is to define objects whose principle purpose is to act as information processors, and which receive the data that they operate on as parameters to operation requests or through some other communication mechanism. In this case, a 'renderer' object would receive presentable objects as input through some interface, and produce a rendering of those objects via some output mechanism.

With regard to the issue of conceptualization, one may conclude that any classification of media types must leave sufficient freedom to allow designers the choice of their design philosophy.

2.3. Structure of classification: flat list vs. minimal hierarchy

A good classification should be organized in a way that reflects the relevant differences between the essential properties of the classified items, and at the same time, removes redundancies. Choosing the optimal set of primitives for media objects, however, remains a difficult endeavor as long as new multimedia input and output devices are appearing from one day to the next. That is, the development of a classification that is 'complete' in any useful sense is highly problematic. Therefore, a useful media classification must also allow the systematic integration of new media, preferably without resort to the most naive approach to classification, which is simply an open-ended flat list of media types.

2.4. Description of data and processes: mathematical elegance vs. algorithmic efficiency

The mathematical algorithms for generating and manipulating specific primitives in multimedia systems are diverse, and too often result from differences in hardware or software environments. Although undesirable, the 'visible' level of a multimedia system (i.e., its API) may reflect such underlying constraints. Consequently, it is difficult to produce a conceptual model that is equally adaptable to the range of environments found in practice. Also, very often, the mathematical (or 'algorithmic') literature does not always indicate clearly which is the best way of handling various primitives. Often, systems offer only a subset of the algorithms or control over primitives, when compared with the levels that are theoretically possible, again based on, e.g., algorithmic decisions. A typical example is how B-spline surfaces are definable: which degree, how many control points, exactly which control points to specify, etc. Another example is the specification of various transformations: does the system allow the specification of various projective transformations in their full generality, or are only a limited subset of projections allowed?

2.5. Impact: timeliness, coverage, compatibility

No new classification, irrespective of its quality, is likely to have much impact if it ignores current facts and trends in computing practice. It could be argued, e.g., that 'first generation' graphics standards such as PHIGS are now outdated, and consequently should have little impact on the design of any new classification. On the other hand, compatibility with widespread hardware architectures, windowing systems, and media display devices must be considered. And this is a serious problem in practice. The X-consortium, and vendors of graphics systems such as Silicon Graphics or Microsoft, X11, take different approaches to the definition of media, which are often mutually incompatible and present problems for any classification which attempts to encompass all approaches. Examples include the way graphic object hierarchies are defined and maintained, the choice of the coordinate systems (e.g., Are they right-handed or left-handed?), the representation of points, color values, etc. These differences may reflect personal taste, internal optimization algorithms, or cultural differences (e.g., referring to the example above, American education favors left-handed coordinate systems, whereas European usually work with right-handed ones).

3. A short overview of PREMO

This presents a short overview of PREMO; for a more detailed presentation the interested reader should consult Ref. [4]. [4]

The ISO/IEC JTC1 SC24 subcommittee, responsible for standards on computer graphics and image processing, recognized the need to develop a new line of standards for presentation environments, which should include more general multimedia effects to encompass the needs of various emerging application areas. A standard in this area should primarily focus on the presentation aspects of multimedia, and less on the coding, transfer, or hypermedia document aspects, which are covered by a number of other ISO/IEC or de-facto standards (e.g., MHEG, HyTime, or MPEG; see, e.g., Ref. [10], for more details on these standards). It should also concentrate on the programming tool side, and less on the multimedia document format side. To this end, a project was started in SC24 for a new standard called PREMO which is currently a Draft International Standard and is now one of the major ongoing activities of ISO/IEC JTC1 SC24 WG6. Some of the major features of PREMO can be summarized as follows.

3.1. PREMO is a presentation environment

PREMO aims to describe a standard programming environment in a very general sense. The aim is to offer a conceptually portable and standardized development environment that helps to promote portable multimedia applications. PREMO concentrates on the application programme interface to presentation techniques; this is what primarily differentiates it from other multimedia standardization projects.

3.2. PREMO is aimed at a multimedia presentation

Whereas earlier SC24 standards concentrated on either synthetic graphics or image processing systems. Multimedia is considered here in a very general sense; high-level virtual reality environments, which mix real-time 3D rendering techniques with sound, video, or even tactile feedback, and their effects, are, e.g., within the scope of PREMO.

3.3. PREMO is object-oriented

This means that, through standard object-oriented techniques, a PREMO implementation becomes extensible and configurable. Object-oriented technology also provides a framework to describe distribution in a consistent manner.

3.4. PREMO is a framework

This means that the PREMO specification does not provide all the possible object types for making graphics or multimedia. Instead, PREMO provides a general programming framework, a sort of middleware, where various organizations or applications may plug in their own specialized objects with specific behavior. The goal is to define those object types which are at the basis of any multimedia development environment, thereby ensuring interoperability.

At the time of writing (Summer 1997), PREMO is in DIS stage; this means that, on the one hand, its technical content is now more or less final and, on the other hand, that it will become an official ISO Standard in 1998.

A precise object model constitutes a major part of PREMO. The object model is fairly traditional, and is based on the concepts of sub-typing and inheritance. It is also very pragmatic in the sense that it includes, for efficiency reasons, the notion of non-object (data) types, as is the case with a number of object-oriented languages, such as C + + or Java, and in contrast to 'pure' object-oriented models such as SmallTalk. The PREMO object model originates from the object model developed by the OMG consortium for distributed objects, but some aspects of the OMG model have been adapted to the needs of PREMO. The model has also undergone a thorough formal specification process (see Ref. [11]). Note

that here is a strong emphasis in PREMO to make it well adapted to distributed environments; this emphasis also directed some of the design decisions reflected below.

4. The PREMO primitive hierarchy

PREMO is concerned with the presentation of multimedia information, and in allowing different renderers to inter-operate within a potentially distributed system. Also, it was an important design requirement of PREMO to allow for extensibility, i.e., that either applications or other, standardized, components would add their own set of primitives to the PREMO framework. For these reasons, the PREMO standard does not attempt to define the structure of primitives to the same level of detail as found, e.g., in graphics standards such as GKS and PHIGS (e.g., see Ref. [12]). Instead, the approach in PREMO is to provide a general, extensible framework that provides a uniform basis for deriving primitive sets appropriate to specific application or renderer technologies. In general, modellers or renderers may use specific techniques, such as Constructive Solid Geometry for a particular range of applications. Such techniques may require an enriched set of basic primitives. The aim of the PREMO primitive hierarchy is to provide a minimal, common vocabulary of structures that can be extended as needed, either by applications using PREMO, or by other standard components.

Referring to one of the issues cited in Section 3, PREMO has deliberately avoided adding explicit procedural 'behavior' to the primitive objects, explicitly separating media processors such as renderer objects from the primitives. One of the main reasons is the fact that PREMO should operate in a distributed model, where one model or data set may be rendered by several processes working in parallel at various locations. It is difficult to see how this can be realized efficiently in an architecture in which each model object renders itself.

Fig. 1 shows the subtype hierarchy of the PREMO primitives. In PREMO, the concept of primitive encompasses the description of both structure and appearance. At the top level, PREMO distinguishes between seven kinds of primitives, which will be described in somewhat more detail in Fig. 1.
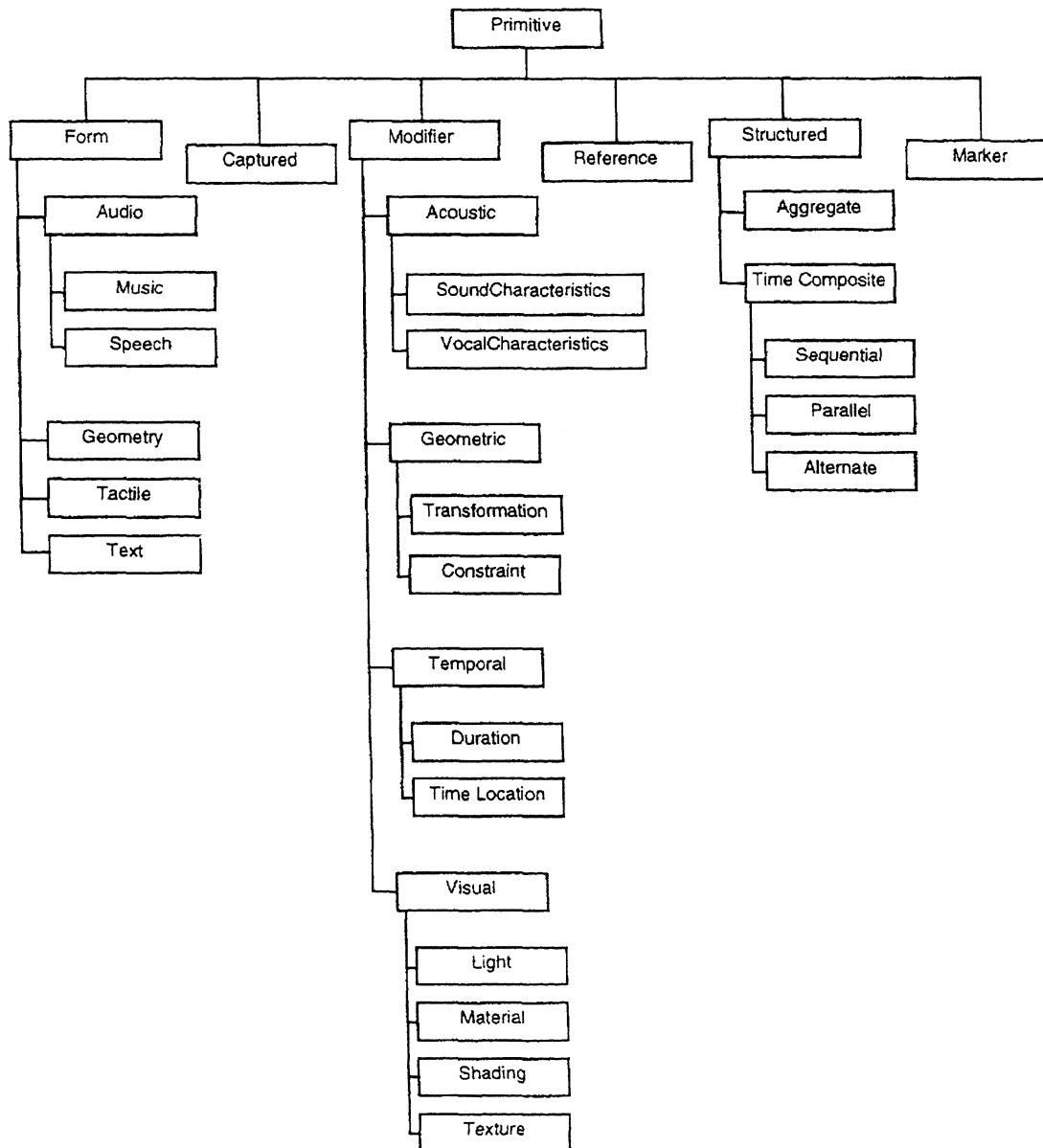
Fig. 1. The PREMO primitive hierarchy.

4.1. Form primitives

A form is a primitive representation has to be generated in some way, e.g., lines, curves, sounds, or textual information. This is in contrast with captured primitives, see below. Note that PREMO does not decide on the various geometric description issues alluded to in Section 3. Instead, by adding some general supertypes for various media descriptions, it allows for applications to 'plug in' their own vari-

ants of media (e.g., their own 'view' of surfaces, point data set, haptic properties, text descriptions, etc.). Additional kinds of form primitives may be added in future to include other categories such as olfactory and taste.

4.2. Captured primitives

A captured primitive contains a reference to a source of raw data encoded in some standard format such as JPEG, MPEG, MIDI, or VRML. This data may happen to be recorded, or live. The detailed specification of this primitive refers to another part of PREMO, called the Multimedia Systems Services, which provides an abstraction for the various (multimedia) virtual devices which may produce such raw data [4].

4.3. Modifier primitives

The primitives in this category have no perceivable representation by themselves. Instead, they carry information that affects the presentation of other primitives. Examples include visual effects (color and texture), geometric transformations, or audio effects. The modifiers have been grouped to reflect the kind of effect that they produce, and the kind of primitives to which they can be applied. PREMO does not describe the order in which modifiers are applied, and whether or not they are accumulative or override previous modifications. The reason for this non-commitment is that applications may realize graphical rendering through existing systems and standards, within which the order and scope of modifications within the rendering pipeline or scene structures varies widely.

4.4. Reference primitives

A reference primitive introduces a link to a structured primitive defined in some other part of the hierarchy. It contains a single name-valued attribute, label, that is intended to be matched against a similar name in a primitive structure.

4.5. Structured primitives

Form, captured, and modifier primitives can be viewed as atomic units of information that determine

or affect the presentation of a multimedia system. Such systems, however, need to define and manipulate collections of primitives, both to represent large-scale or application-specific structures, and to coordinate the presentation of primitives over time. These two roles are somewhat different, and are reflected in PREMO by two object types that encapsulate a collection of primitives. This collection may itself include structured primitives, allowing the construction of hierarchical structures.

Aggregates allow a number of primitives to be combined into a structure without imposing any interpretation on the meaning of such a collection. They provide a facility for building larger-scale primitives and also allow an application to group semantically related primitives into single units that can be named. Application or other standard components may impose a particular view of structuring (e.g., Directed Acyclic Graphs). Aggregates also have a naming mechanism, whereby primitives can be labelled by a name built from a sequence of strings. This name can be used by reference primitives, and/or various selection mechanisms.

Time and temporal extent are fundamental to multimedia presentation and in general a multimedia system will contain a number of primitives which need to be synchronized in time. Although time could arguably be treated in a way similar to that used for spatial coordinates, most multimedia systems will typically treat time in a specialized way, to support the realization of various time-related constraints, synchronization, etc. The *Time Composite* object of PREMO has been introduced to structure primitives in the time domain. It contains a sequence of component primitives, inherited from the *Structured* primitive object type, that defines the content of the composite. The object also contains various attributes which make it possible to monitor and control the timing of the composite as a whole. These include duration, start and end time 'buffers' that provide flexibility in coordinating the presentation of multiple *Time Composite* objects, and an event monitor through which external objects can be informed of, e.g., the progress of a renderer in processing the object. PREMO defines three specific subtypes of *Time Composite*:

- Sequential time composite, in which the component primitives are presented in sequential order;

- Parallel time composite, in which the components are presented and synchronized concurrently;
- Alternate time composite, which also includes a finite state machine, whose current state will direct which component primitive should be presented.

4.6. Marker primitives

Marker primitives are used to provide one level of coordination between media devices by linking the passage of data through media streams to events. A marker primitive contains a reference to an event, and whenever a marker primitive is encountered at a port of a media device, the event carried by the primitive is dispatched to the event handler associated with the port.

5. Assessment and the limitations of the PREMO primitive hierarchy

We assess and discuss some limitations of the PREMO Primitive Hierarchy, firstly, in general (Section 5.1), and secondly in view of the SRM (Section 5.2).

5.1. General comments

The esoteric variety of the various design issues discussed in Section 2, and the often conflicting specification of existing multimedia systems made it very clear to the designers of the PREMO that the only way forward was to be as noncommittal as possible. It was recognized in a very early stage that what PREMO should provide is more of a reference model, a kind of common language, rather than a detailed specification. This approach dominates the various design considerations adopted in the Standard.

The natural question which arises when keeping to such a generality is how useful a minimal common language can be? What are its potential roles? The answer lies in the concept of interoperability of various systems. Experience has shown that no single system or specification is likely to dominate the whole field of multimedia (unless all systems stem from the very same provider) but, instead, different

systems will coexist and, in our inter-networked world, should cooperate. Such cooperation relies on a common vocabulary, a common language, and it can be reasonably required that these systems would at least understand such a common, general vocabulary. Establishing the basis for such a vocabulary is the primary objective of the PREMO hierarchy.

A limitation of this approach is that mechanisms should exist for various systems to cooperate, and this does not only require a common vocabulary, but a way this vocabulary could be put in practice, i.e., systems should be able to disclose their capabilities, their properties, and other systems should be able to understand these properties and coordinate the cooperation of various entities. Another large portion of PREMO is devoted to define such a network of cooperation, but a detailed description of this mechanism would go beyond the scope of the paper. The interested reader should consult either Ref. [4] or the standard document itself.

5.2. Assessment from the perspective of the SRM for IMMPS's

From the first, it is clear that there is no unequivocal answer to the initial question of whether or not the PREMO Primitive Hierarchy satisfies the requirements which the SRM would place on a 'perfect' media classification.

The answer might be affirmative for the role of the media classification in the specification of interfaces between layers and components. The PREMO primitive hierarchy is the best system-centered definition of media types we currently have. Its strength is that it actually captures the diverse spectrum of media which are currently in use. Rather than claiming completeness, the hierarchy aims at providing the umbrella which leaves sufficient room for both different variants of media descriptions and also the incorporation of new media types. The SRM aims at a similar goal. Rather than prescribing a particular set of media-specific design and realization components, the SRM just shows where such components should be plugged in (see Section 4.3 and Section 4.4 of the SRM paper).

Though, in principle, it would be possible to formulate design knowledge upon the PREMO primitives, also, the answer to the initial question might

be negative in this case. This is not surprising since the PREMO Primitive Hierarchy is a purely system-centered classification and does not reflect a human-centered perspective at all. To be more precise, some of the criteria used by the PREMO designers for distinguishing their primitives are not relevant for the formulation of design knowledge. Vice-versa, there are criteria (e.g., the user's cognitive effort for processing media objects of a certain type, applicability constraints, etc.) which are not considered in PREMO but which cannot be neglected when defining a suitable classification of media types that would allow to formulate design knowledge in an intuitive manner. Consider, e.g., the task of media allocation, which is to select from available presentation media the one which can most effectively convey a given information. Whether or not a certain medium fulfills the requirement of being effective can only be answered by relating the properties of media to the capacities and peculiarities of human perceptual processes. The primitives in the PREMO hierarchy do not establish such a relation. The distinction drawn by the PREMO primitives can even lead to more complicated design rules. For example, if the PREMO primitives were used to formulate a naive rule such as 'use graphics for localization tasks', then it would be necessary to replace in the rule the term 'use graphics' by the less intuitive expression 'use either Geometric, or Captured, or Structured'. However, for the consumer of the presentation it doesn't make any difference whether the graphics has been generated by the system, whether the system presents a 'captured' graphics, or whether the presented graphics has been composed of some generated and some captured parts.

6. Conclusions

In this paper, we have discussed some of the typical problems with trying to establish a classification for media types. We have presented the Primitive Hierarchy of the PREMO standard for multimedia runtime environments. Being thoroughly designed to capture the broad array of available media, this hierarchy is the best system-centered definition of media types we currently have. The question was

raised, whether the PREMO primitive hierarchy would be a suitable adjunct to the SRM for IMMPS [1]. One issue in the context of the SRM is the specification of interfaces between layers and components of layers. For this purpose, the PREMO primitive hierarchy has been considered useful. Moreover, adopting this classification would facilitate the instantiation of SRM's Presentation Display Layer with PREMO as it is already suggested by the proposers of the SRM [1].

For the purpose of formulating design knowledge, however, the PREMO Primitive Hierarchy appears to be too much system-centered. This should not be understood as a criticism to the designers of PREMO, since, for them, the formulation of design knowledge was never an issue. One may rather conclude that further research is needed in order to establish an ideal media classification which merges both the human-centered perspective and the system-centered perspective. Since the device-related descriptions of media, and those required to capture rules of thumb, differ so greatly, complexity is required at one stage of the process: either to map from the rules of thumb to a device-based description, which would constrain rule structure but improve runtime efficiency, or from a user-centered description in which rules of thumb are easily expressed to the devices within the SRM at runtime. Similarities were suggested earlier between this attempt to standardize media taxonomies the problem of describing color (this latter has been the subject of research for a long time). The usual solution in the case of color was to use one standard (RGB) which easily mapped to the implementation of systems, and could be used to describe them; a second standard (HSV) was introduced as a user-oriented description, while a third (CIE) was used to map between the two, incorporating both psychological and physical aspects of color. (Indeed, the CIE color model addresses the problem of device independent specification of color by placing the monitor phosphors and white points precisely in color space.) A similar solution may be required in order to formulate design knowledge of media, with one description, such as PREMO, to be used for the generation environment, and another, which is user-centered, to be used to elicit design knowledge and integrate empirical findings with rules of thumb (perhaps based on the proposals of Bernsen [5]), while a

third integrating classification is required to support the device independent description of the rules within an implementation, and to map between the other two. Note that the PREMO approach, which tries to abstract away to the least degree of device independence, shows also the way toward a methodology for the integrating specification scheme.

However, the presented assessment of the PREMO primitive hierarchy makes clear that a common vocabulary would be only one half of the coin. To achieve impact in practice, research must also be directed toward the development of (preferably standardized) mechanisms which allow systems and system components to disclose, exchange, and process information on the use of media, media processing capabilities, media formats, synchronization conditions, applicability constraints, etc.

References

[1] M. Bordegoni, G. Faconti, S. Feiner, M.T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, M. Wilson, A Standard Reference Model for Intelligent Multimedia Presentation Systems, Computer Standards and Interfaces 18 (6,7) (1997) 477–496.

[2] G.R. Bieger, M.D. Glock, Comprehending spatial and contextual information in picture–text instructions, J. Exp. Educ. 45 (4) (1986) 181–188.

[3] P.R. Cohen, The role of natural language in a multimodal interface, in: UIST '92, ACM Press, New York, 1992, pp. 143–149.

[4] I. Herman, G.J. Reynolds, J. van Loo, PREMO: an emerging standard for multimedia presentation, IEEE Multimedia 3 (3) (1996) 83–89.

[5] N.O. Bernsen, Defining a taxonomy of output modalities from an HCI perspective, Comput. Standards and Interfaces 18 (6,7) (1997) 539–551.

[6] D.M. Frohlich, The design space of interfaces: multimedia systems, interaction and applications, Proc. 1st Eurographics Workshop, Stockholm Sweden, Springer, Berlin, 1991, pp. 53–69.

[7] L. Nigay, J. Coutaz, A design space for multimodal systems: concurrent processing and data fusion, Proc. of INTERCHI 1993, ACM Press, New York, pp. 172–178.

[8] J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hugues, Computer Graphics, Principles and Practice, Addison-Wesley, 1990.

[9] E. Hovy, Y. Arens, Virtual devices: an approach to standardizing multimedia system components, in: G.P. Faconti, T. Rist (Eds.), Proc. of the ECAI'96 workshop Toward a Standard Reference Model for Intelligent Presentation Systems, Budapest, 1996.

[10] S.J. Gibbs, D.C. Tsichritzis, Multimedia Programming, ACM Press series, Addison-Wesley, New York, 1995.

[11] D.A. Duce, D.J. Duke, P.J.W. ten Hagen, I. Herman, G.J. Reynolds, Formal methods in the development of PREMO, Comput. Standards Interfaces 17 (1995) 491–509.

[12] D.B. Arnold, D.A. Duce, ISO Standards for Computer Graphics: the First Generation, Butterworth, London, 1990.

David Duke undertook his doctorate in the area of object-oriented formal specification as part of the Object-Z group at Queensland. He joined the University of York in 1992 as a research associate on the ESPRIT Amodeus-2 project, and is now a lecturer in Computer Science. His research interests include computer graphics and multimedia, formal methods and interactive systems, and reasoning about human–computer interaction. Within HCI, his work focuses on multimodal interfaces, using an approach to reasoning about human–computer interaction called 'syndetic modelling' that formally links system components to the cognitive resources of the user. In graphics and multimedia, he has been an active participant in the development of the PREMO Standard within ISO/IEC JTC 1/SC24 since 1993, first through the use of formal description techniques, and since 1995 as an editor of the Standard. He is a member of the Eurographics Association, the IEEE Computer Society, and BCS-FACS.

Dr. Ivan Herman graduated in Budapest, Hungary in 1979 as an applied mathematician. He joined the Computing and Automation Institute of Budapest in 1979. He later left Hungary in 1986 and joined the software company Insotec Consult in Munich, Germany. Finally, in 1988, he joined the Center for Mathematics and Computer Sciences (CWI) in Amsterdam, as a senior researcher, where he is currently project leader at the department of Information Systems. He received his PhD title in 1990, at the University of Leiden (the Netherlands). He was involved in several research and development projects aiming at the realization of graphics and multimedia systems both on pure software or firmware level. These systems also involved various standards like GKS, PHIGS, or techniques like distributed or dataflow programming. He has had numerous results related to these systems work. He won several Eurographics Best Paper Awards and is the author or co-author of more than 40 scientific publications. He is one of the editors of the ISO/IEC JTC1/SC24 PREMO standard. Further, information are also available on the URL

http://www.cwi.nl/ivan.Dr. Thomas Rist is a senior researcher at the Department Intelligent User Interfaces of the German Research Center for Artificial Intelligence (DFKI). Dr. Rist studied computer science and mathematics at the University of the Saarland. He undertook his doctorate in the area of knowledge-based graphics generation and received the degree from University of the Saarland. He was involved in various projects funded by the German Ministry for Education and Research, industries, and the EU. One of these projects, WIP, was a winner of a 1995 Information Technology Award (ITEA). As a Senior Scientist, he was also in charge of management tasks in a number of industrial projects, and is currently coordinator of the European LTR Project Magic Lounge within the i3 initiative on Intelligent Information Interfaces. His areas of interest and experience include: multimedia/multimodal communication, intelligent user interfaces, automated presentation systems, knowledge-based graphics generation, life-like characters, and user modelling. Dr. Rist has served as a programme committee member for national and international conferences and workshops, and was on the organization board of a number of workshops on multimedia systems. In 1995, Dr. Rist became a member of the ERCIM Computer Graphics Network Task 2 Group.

Michael Wilson obtained a bachelors degree in Experimental Psychology from the University of Sussex, and a doctorate in Psycholinguistics from the University of Cambridge. After 1983, he studied how users learn to use windowing systems on a project funded by IBM at the MRC Applied Psychology Unit, Cambridge. Since 1986, he has worked at the Rutherford Appleton Laboratory researching intelligent user interfaces incorporating multiple interaction modes, automatic generation of presentations, ontology-based information retrieval, knowledge acquisition, multimedia information retrieval and presentation, and conversational interaction. He is a member of the BCS and ACM, and has served on the programme committees of many international conferences, and several journal editorial boards on HCI and AI. He has acted as a research programme advisor and reviewer for UK and European research programmes, and is currently coordinator of the UK EPSRC programme on Multimedia Networking Applications.