



Centrum voor Wiskunde en Informatica

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

D.J. Duke, I. Herman, T. Rist and M. Wilson

Information Systems (INS)

INS-R9704 1997

Report INS-R9704
ISSN 1386-3681

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

CWI is the National Research Institute for Mathematics and Computer Science. CWI is part of the Stichting Mathematisch Centrum (SMC), the Dutch foundation for promotion of mathematics and computer science and their applications.

SMC is sponsored by the Netherlands Organization for Scientific Research (NWO). CWI is a member of ERCIM, the European Research Consortium for Informatics and Mathematics.

Copyright © Stichting Mathematisch Centrum
P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

Relating the Primitive Hierarchy of the PREMO Standard to the Standard Reference Model for Intelligent Multimedia Presentation Systems

D.J. Duke

*University of York
Heslington, York Yo1 5DD, UK
Email: duke@minster.york.ac.uk*

I. Herman

*CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Email: Ivan.Herman@cwi.nl*

T. Rist

*German Research Centre for Artificial Intelligence (DFKI)
Saarbrücken, Germany
Email: rist@acm.org*

M. Wilson

*Council for the Central Laboratory to the Research Councils (CCLRC), Rutherford Appleton Laboratory
Chilton, Didcot, Oxon., OX11 0QX, UK
Email: mdw@inf.rl.ac.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 SRM (Standard Reference Model, 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 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.

1991 Computing Reviews Classification System: D.1.5, D.2.0, H.5.0, H.5.1, I.2.4.

Keywords and Phrases: multimedia, multimedia modeling, multimedia presentation systems, knowledge representation, PREMO, geometric primitives, standards.

*Note: This paper is submitted for publication to the journal *Computer Standards & Interfaces*. At CWI, the work was carried out under the project INS3.1: "Information Engineering Framework".*

1. Introduction

Bordegoni et al [4, 5] 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 Allocation component, and the Design and Realisation Layers of this pipeline contain media specific Design and Realisation components, each of which can draw on centralised 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 Realisation components. While early approaches were based on rules or templates, approaches which rely on hierarchical planning, or constraint solving, as well as combinations of them 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. [3]), 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 [6]). 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 towards the sharing, exchange, and reuse of design knowledge. The construction of such classifications has always proved 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 PREMO ISO/IEC standard which is being developed for such environments. This instantiation is not developed in this paper, but interested readers should consider the overall PREMO standard more fully [11].

The PREMO Standard contains a hierarchy of primitive, representing a system-oriented approach to media classification. This is in contrast to Bernsen's [2] modality classification, 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 what follows 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 and, by giving a short overview of the current primitive hierarchy, will show how the designers of PREMO addressed these issues. Finally, 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 a 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 regard to the context of the current paper as they have been addressed by the designers of the PREMO Primitive Hierarchy which is presented later.

Perspective: human centred vs. system centred

Frohlich [9] and Bernsen [2] present classifications based on the human sensory/representational modalities of the user. In contrast Nigay and Coutaz [13] present an analysis of the interface design space which is centred on the system in order to aid designers to identify the software implications and constraints for developing their systems.

[13] avoids presenting a classification of media types, but like them, the PREMO approach is system centred 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 for example the choice of a colour 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 colours, 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 colour 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 colour based algorithms of synthetic graphics (shading, etc.) become more complicated (see [8]).

With regard to the formulation of design knowledge, a media classification certainly must not ignore the human-centred perspective. To avoid misuse and underuse of media the yardstick of any assessment must involve the human user (see also [12]). On the other hand, engineering pragmatism has taken into account when striving for a specification of interfaces between layers and components of the SRM.

Conceptualisation: 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 conceptualisation 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 behaviour, such as transformations and rendering, to object types within an API. Two quite distinct approaches emerge. The first is to attach behaviour to the object types that are affected by that behaviour. 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 conceptualisation, one may conclude that any classification of media types must leave sufficient freedom to allow designers the choice of their design philosophy.

Structure of Classification: flat list vs. minimal hierarchy

A good classification should be organised 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 endeavour 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.

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?

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, for example, 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 Inc. or Microsoft Inc., 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 righted?), the representation of points, colour values, etc. These differences may reflect personal taste, internal optimisation algorithms, or cultural differences (e.g., referring to the example above, American education favours leftordinate systems, whereas European usually work with right

3. A short overview of PREMO

This section presents a short overview of PREMO; for a more detailed presentation the interested reader should consult [11].¹

The ISO/IEC JTC1 SC24 subcommittee, responsible for standards on computer graphics and image processing, recognised 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 (for example MHEG, HyTime, or MPEG; see, e.g., [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 (Presentation Environment for Multimedia Objects) 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 summarised as follows:

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 standardised 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 standardisation projects.

1. The Premo Web home page also contains a number of further references and additional interesting information, see: <http://www.cwi.nl/JTC1SC24/WG6/Premo>.

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, for example, within the scope of PREMO.

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.

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 organisations or applications may plug in their own specialised objects with specific behaviour. 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 [7]). 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, standardised, 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 for example in graphics standards such as GKS and PHIGS (see, e.g., [1]). 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 the previous section, PREMO has deliberately avoided adding explicit procedural 'behaviour' 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 realised efficiently in an architecture in which each model object renders itself.

Figure 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 what follows.

Form Primitives

A form is a primitive representation has to be generated in some way, for example lines, curves, sounds, or textual information. This is in contrast with captured primitives, see below. Note that PREMO does not decide on the var-

Figure 1: The PREMO Primitive Hierarchy

ious geometric description issues alluded to in the previous section. Instead, by adding some general supertypes for various media descriptions, it allows for applications to 'plug in' their own variants 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.

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 [11].

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 (colour 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 realise graphical rendering through existing systems and standards, within which the order and scope of modifications within the rendering pipeline or scene structures varies widely.

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.

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 co-ordinate 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 synchronised 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 specialised way, to support the realisation of various time-related constraints, synchronisation, etc. The *TimeComposite* 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 *TimeComposite* objects, and an event monitor through which external objects can be informed of, for example, the progress of a renderer in processing the object. PREMO defines three specific subtypes of *TimeComposite*:

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

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

Marker Primitive

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

In the following two subsections we assess and discuss some limitations of the PREMO Primitive Hierarchy, firstly in general, and secondly in view of the SRM.

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 non-committal as possible. It was recognised 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 [11] or the standard document itself.

5.2. Assessment from the Perspective of the SRM for IMMPSõs

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 centred 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 realisation components, the SRM just shows where such components should be plugged in (see sections 4.3 and 4.4 of the SRM paper).

Though, in principle, it would be possible to formulate design knowledge upon the PREMO primitives, too, 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-centred classification and does not reflect a human centred 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 for example 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 fulfils 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 localisation 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 centred 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 [4]. 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 [4].

For the purpose of formulating design knowledge, however, the PREMO Primitive Hierarchy appears to be too much system centred. 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 humane 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 userion in which rules of thumb are easily expressed to the devices within the SRM at runtime. Similarities were suggested earlier between this attempt to standardise media taxonomies the problem of describing colour (this latter has been the subject of research for a long time). The usual solution in the case of colour 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 userio, incorporating both psychological and physical aspects of colour. (Indeed, the CIE colour model addresses the problem of device independent specification of colour by placing the monitor phosphors and white points precisely in colour 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 userge and integrate empirical findings with rules of thumb (perhaps based on the proposals of Bernsen [2]), 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 towards the development of (preferably standardised) mechanisms which allow systems and system components to disclose, exchange, and process information on the use of media, media processing capabilities, media formats, synchronisation conditions, applicability constraints etc..

Acknowledgements

The research reported in this paper was partly funded by EC Esprit IV grant 20597 to the Chameleon consortium and partly by EC HCM grant CHRX-CT93-0085 to the ERCIM Computer Graphics Network.

References

1. D.B. Arnold and D.A. Duce, *ISO Standards for Computer Graphics: The First Generation*, (Butterworths, 1990).
2. N.O. Bernsen, A Reference Model for Output Information in Intelligent Information Presentation Systems, To be published in: *Computer Standards & Interfaces* (1998).
3. G.R. Bieger and M.D. Glock, Comprehending Spatial and Contextual Information in Picture-Text Instructions. *The Journal of Experimental Education*, 45(4), (1986) pp. 181-188.
4. M. Bordegoni, G. Faconti, M.T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, M. Wilson, A Standard Reference Model for Intelligent Multimedia Presentation Systems, To be published in: *Computer Standards & Interfaces* (1998).
5. M. Bordegoni, G. Faconti, M.T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, M. Wilson, Intelligent Multimedia Presentation Systems: A Proposal for a Reference Model. In: (J.P. Courtiat, M. Diaz, P. Sénac (eds.), *Proc. of the MMM'96 conference, Multimedia Modeling*, (World Scientific, Singapore – New Jersey – London – Hong Kong), pp. 3–20.
6. P.R. Cohen, *The Role of Natural Language in a Multimodal Interface*, In: *UIST '92*, (ACM Press, New York, 1992), pp. 143-149.
7. D.A. Duce, D.J. Duke, P.J.W. ten Hagen, I. Herman, and G.J. Reynolds, Formal Methods in the Development of PREMO, *Computer Standards & Interfaces*, 17 (1995), pp. 491-509.
8. J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hugues, *Computer Graphics, Principles and Practice*, (Addison Wesley, 1990).
9. D.M. Frohlich, The Design Space of Interfaces: Multimedia Systems, Interaction and Applications. In: *Proceedings 1st Eurographics Workshop, Stockholm Sweden* (Springer Verlag, Berlin
10. S.J. Gibbs and D.C. Tschritzis, *Multimedia Programming*, (Addison-Wesley, ACM Press series, 1995).
11. I. Herman, G.J. Reynolds, and J. Van Loo, PREMO: An Emerging Standard for Multimedia Presentation, *IEEE MultiMedia*, 3(3), (1996) pp 83-89.
12. E. Hovy and Y. Arens, Virtual Devices: An Approach to Standardizing Multimedia System Components. In: G.P. Faconti and T. Rist (eds), *Proc. of the ECAI'96 workshop "Towards a Standard Reference Model for Intelligent Presentation Systems."* (Budapest, Hungary, 1996).
13. L. Nigay and J. Coutaz, A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion. In: *Proceedings of INTERCHI 1993* (ACM Press, New York, 1993), pp. 172-178.