STANDARDS AND REFERENCE ARCHITECTURES : THEIR PURPOSE AND ESTABLISHMENT

Michael Wilson

CLRC, Rutherford Appleton Laboratory, Chilton, DIDCOT, OXON., OX1 0QX

M.D.Wilson@rl.ac.uk

Abstract

The benefits or standardisation for interoperability and a description of the organisational structure and standardisation process of ISO is presented. The role of architecture and API standards at different stages in the technology life cycle are described. Examples of the development of open standards are taken from graphics, and the WWW to show the role of standardising different aspects of applications, as the application technology matures. The content of the architectural reference models adopted for computer graphics and intelligent multimedia presentation systems are outlined to show the components expected in any reference model for natural language processing, and the constraints they place on such a reference model to maintain compatibility .

1. Introduction

The general goal of IT standards is to help developers to create systems which meet the functional and usability requirements of users, with minimal effort and time. To reach this overall goal, there are standards for achieving different aspects of the goal, which are stated at different levels of specificity consistent with the maturity of theory development in the application area for functionality definition, processing architecture, data representation for interchange, functionality and usability evaluation, and quality assurance.

This paper outlines the standardisation process, as it currently exists, and may be followed in the future for Natural Language Processing (NLP).

1.1 What is a standard ?

A standard is not necessarily the best way to do something, rather a standard is the commonly agreed way to do it. A standard cannot be dictated by a single individual or organisation, rather it is a compromise between those making the standard. These assertions often seem hard for theoreticians in an area to conform to. Although they may believe that they know the best way to do something, they are not going to move forward until their discipline collectively agrees on an approach.

The second aspect of standards which is often hard for theoreticians to conform to is that standards are not tutorials rather they are minimal, unambiguous, registered definitions of things which are accessible to all. To help overcome this minimalism constraint, standards usually contain two main classes of statements which are termed:

Normative statements include both mandatory and permissible behaviour. Mandatory must be conformed to, in order to conform to the standard, while the permissible parts allow some latitude in implementation while remaining conforming.

Informative statements are those which indicate a possible way of instantiating aspects of the normative definitions which are left undefined. Conformance to these is not required in order to conform to the standard, although they may define regions of behaviour within which conformance holds, but outside of which it does not.

Many of those who regularly produce standards, make a lucrative living by writing books and articles which act as introductions to standards, or tutorials on them. However, these are additional publications and are not the minimal standards themselves.

1.2 Who standardises ?

There are many forums which declare standards. For standards to be adopted they must carry some weight in the user community. Generally, there are organisational, *de facto*, and open standards.

Organisational standards. Within a single company procedures for doing things, ways to present things, and components to be used are often regularised, and standardised. These regularisations usually help those in the organisation know how things are done, staff have skills which are common throughout the organisation which increases their ease of movement, the organisation presents a common image to the outside world so that it is always easily identified, and the standard use of common components leads to lower inventory volumes, and therefore lower amounts of capital tied up at any time. IT standards are usually promoted because of the interoperability of components they facilitate.

De facto standards are often proprietary to one organisation, who can change the definition when they wish to, but they are also the dominant force in a domain, so they are also used by many other organisations. Therefore *de facto* standards are intermediary between organisational and open standards, providing the benefits of standardisation, but without providing many users any control over them.

Open standards are standards registered by an open, representative organisation to which others can belong and have some control. The quintessential organisation for registering open standards is the International Organisation for Standardisation (ISO). ISO is divided into a series of committees, subcommittees and working groups whose membership represent national standards bodies. ISO's standardisation process is entirely open through this hierarchy to influence from any organisation. Therefore open standards provide the advantages of interoperability which *de facto* standards do, but also provide an open route for all users to have some control over them.

1.3 ISO IT Standardisation.

The world of international standards lives by committee abbreviations and numbers. This section briefly introduces some of those that you may encounter in this paper and elsewhere.

ISO/IEC Joint Technical Committee 1 (JTC1) is the committee for information technology. All IT standards go through JTC1, except for a few Ergonomics Standards (e.g. ISO 9241 Parts 10 - 17 Ergonomic Requirements for Office Work with Visual Display Terminals) which go through Technical Committee (TC) 159, Sub-Committee (SC) 4 on Ergonomics of Human Computer Interaction.

Several bodies other than ISO also produce open standards, through a formalised process (e.g. IETF for Internet protocols, ITU for telecommunications).

W3C is the consortium of organisations who use, build tools for, or provide services over the WWW, and who collectively agree on the languages and protocols which the WWW will use. W3C is not a standards body like ISO so it produces recommendations rather than standards. W3C has its own closed process of establishing working groups from its membership who draft recommendations, and then all member organisations vote on acceptance of these. If adopted they become recommendations of W3C.

The W3C process is not as open as the ISO one and does not include equal representation at a national level linked to national standards organisations, rather member organisations have direct representation on working which develop recommendations. However W3C does have a category 3 liaison agreement with ISO/IEC JTC1 to work together on activities of common interest. The first W3C recommendation to move into the ISO standardisation process through this mechanism is that for the representation of bitmapped graphics images - PNG.

PNG is being lead through the ISO standardisation process by an editor within JTC1/SC 24, the sub-committee on Computer Graphics and Image Processing. Most technical ISO standards work takes place at this sub-committee level. The sub-committee most likely to undertake any formal standardisation for NLP would be JTC1/SC 35 on User Interfaces, which was established in 1988.

1.4 The Standardisation Process

ISO standards are developed according to the following principles:

Consensus The views of all interests are taken into account: manufacturers, vendors and users, consumer groups, testing laboratories, governments, engineering professions and research organisations.

Industry-wide Global solutions to satisfy industries and customers world wide.

Voluntary International standardisation is market-driven and therefore based on voluntary involvement of all interests in the market-place.

The need for a standard is usually expressed by an industry sector, which communicates this need to a national member body. The latter proposes the new work item to ISO as a whole (stage 0). Once the need for an International Standard has been recognised and formally agreed, the first phase involves definition of the technical scope of the future standard (stage 1). This phase is usually carried out in working groups which comprise technical experts from countries interested in the subject matter.

Table 1: The seven stages of the ISO standardisation process (timings from the STEP standard - ISO 10303).

Stage	Description	Time Schedule
0	preliminary work item	
1	new work item	Start
2	working draft	6 months
3	committee draft	12-18 months
4	draft international standard	
5	final draft international standard	36 months
6	published international standard	

Once agreement has been reached on which technical aspects are to be covered in the standard, a second phase is entered during which countries negotiate the detailed specifications within the standard. This is the consensus-building phase (stages 2,3 & 4).

The final phase comprises the formal approval (stage 5) of the resulting draft International Standard (the acceptance criteria stipulate approval by two-thirds of the ISO members that have participated actively in the standards development process, and approval by 75 % of all members that vote), following which the agreed text is published as an ISO International Standard (stage 6).

1.5 What to standardise ?

Although programming languages are the things standardised by ISO (or the national committees: ANSI in the USA; BSI in the UK) which most NLP researchers encounter, there are other aspects of the development process which can be standardised as an application discipline matures :

Best practice guidelines - often used in HCI, and arguably should not be ISO standards, but IFIP best practice recommendations.

Development & management methods - which link to QA

Since these are both informally specified, they can be adopted in different ways once established. For example, ISO 9241 Parts 10 - 17 describe best practice guidelines as requirements for user interfaces. In Germany they have become part of the national law, and must be followed by all software developers; while in the US, developers usually refer to them in contracts with customers as part of a QA process subject to civil litigation. Since in no case are there any tests stated to clearly adjudge conformance, the UK community generally gives them no more status than as guidelines to designers.

Device Interface Standards - interfaces to hardware device drivers, which ensure portability of applications across hardware. These are usually divided into logical input devices, and physical ones to enable applications to use logical values without regard for the actual physical device.

Reference Model/Architecture (RA) - usually a pipeline description of a process which defines the functionality of modules.

Interchange Format (IF) - defining the representation of data to be passed between modules in a RA. Sub-dividable into abstract and concrete syntax, and between archival and compressed transport formats.

Applications Programmer Interface (API) or Object Model (OM) - to define the access calls to a module in an RA which would pass data.

Language Bindings for an API or OM - bindings for specific programming languages of APIs.

The remainder of this paper will consider examples of these last four classes of standard. However, it should not be forgotten that ISO 9241 Part 19 (handled by ISO TC159 SC4 WG5) on Ergonomics Requirements for Office Work with Visual Display Terminals using Natural Language Dialogues has still not been completed although it was started in 1984. To overcome this delay, the US version of this standard ANSI/HFES 200 has incorporated sections on voice and telephony, including speech recognition, speech output, and interactive voice response, including vocabularies, message format, speech characteristics, and dialog techniques.

The ISO IT standards usually considered the most successful in the long term are the IF ones. However, each has a dominant stage in the development of an application technology as described in the next section.

2. Standards in the IT technology lifecycle

Figure 1 shows the technology development cycle from initial basic research, through demonstrators to marketable products. The objective of technology development is to move a technology into a successful product, that is, one with no risk of failure in the marketplace.

During this progression the risk of the product failing gradually reduces, but concomitantly the cost of each step increases. Standards serve a role of reducing risk both for demonstrators of the application of a technology, and for industrial prototypes which are about to be launched into the marketplace. For the former, architecture standards provide a common language to easily compare application demonstrators produced at different research laboratories. This allows the unification of the demonstrator results, providing more data than any one group has to enable them to tune their industrial prototypes. Once industrial prototypes have been developed, standards for API's and IF's maximise their adoption in the marketplace, by maximising their inter-operability with existing applications and technologies.

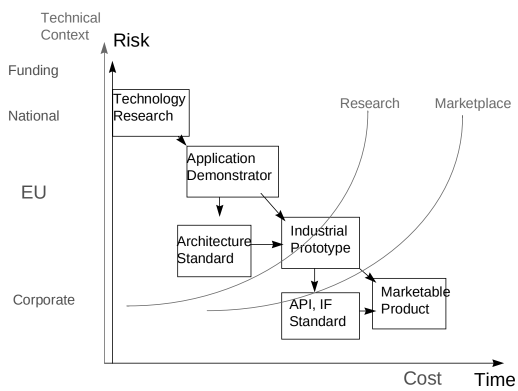

Figure 1: Technology development cycle, showing the reduction in risk of a product failure by use of standards.

Given the current, theoretically diverse, state of the development of NLP technologies, a standard reference model and architecture would provide the notions to be used to compare existing application demonstrators.

3. Standards and Developing Application Technologies

The previous section has asserted how standards can aid technology development and adoption, and how different standards reduce the risk of technology failure at different stages on its life cycle. This section presents two historical accounts of the standardisation of application technologies as they developed. Several lessons are evident from these for any plan to standardise NLP.

The first example chosen is that of computer graphics, since its role and overall architecture is not dissimilar to that of natural language generation, while it is about 25 years ahead in the technology development lifecycle and standardisation process.

The second example is the World Wide Web (WWW) since conformance to its standards appears to be becoming a *de facto* requirement most IT development.

3.1 Example 1 - Computer Graphics

The portability of computer graphics programs became a problem in the early 1960's following the introduction of the first Tektronix storage displays (Arnold and Duce, 1990). Various packages became de facto standards (e.g. Cambridge University's GINO-F) standardisation was not effectively initiated until the creation of the Graphics Standards Planning Committee by ACM SIGGRAPH in 1972. Subsequent considerations of the overall graphics architecture separated four components: the *application programme* which called the *device independent front end*, and the *device specific back end* (or "device driver") on top of the *operating system*. A second distinction made was between transformations used to model or construct the object and those required to view it. Both of these distinctions could usefully be carried through to NLP architecture or API standards.

The first formal meeting of the ISO graphics working group was in 1978 where an API definition called GKS was considered for the first time. GKS separated the API definition from the language bindings which were standardised in parallel. Three further working group meetings considered GKS before it was published as the first graphics standard in 1985 (ISO 7942). As work on GKS was ending, it was agreed in 1983 to standardise three dimensional graphics API's through two proposals - GKS-3D and PHIGS. These were published in 1988 and 1989 respectively.

In parallel with the API standardisation, a proposal for the standardisation of an interchange format was approved early in 1983, which was published as the CGM metafile standard ISO 8632 in 1987. Suppliers appear more neutral about interchange standards than API ones. Adding a CGM interface to an existing product is likely to increase the applicability of a product, whereas to replace an API is to enter a new business. Considering future NLP standardisation in the light of this experience, it may be faster and more productive to develop standards for interchange formats than for API's.

More recently, VRML was published as an ISO/IEC standard in 1998, having originated as a baseline document produced by the VRML Consortium. Similarly, the World Wide Web Consortium (W3C) has proposed a baseline document for the PNG bitmapped graphics image standard which is currently going through the standardisation procedure. These examples may be the start of a new trend for ISO standards to originate from external industry sub-groups, with sub-committee members bringing process and drafting skills to them. For NLP standardisation, it may be faster to establish an industry sub-group to develop standards in this way, passing them over to ISO, rather than try to act from within the existing formal ISO structure.

It was not until 1985 that work started on a suitable reference model for computer graphics, which it was generally thought would have accelerated the process of achieving consensus on the individual standards. It was not until 1992 that the reference model was finally published (ISO, 1992a). When considering the standardisation of NLP, a reference model and architecture would be the best starting point given this experience.

One of the key problems for graphics standardisation in this 25 year period was the changing hardware under the graphics application, e.g. affordable raster graphics arrived in 1983, completely changing the techniques involved. Any standards for NLP would require such future proofing by carefully dividing the device dependent and independent components, as well as predicting the roadmap of technology development.

A second key problem, which has indeed still not been resolved is how graphics standards and window systems co-exist. For the standardisation of natural language generation systems for use at the computer interface, similar problems could arise unless considered early enough.

Thirdly, there was obvious US/European polarisation (e.g. with PHIGS/GKS-3D) and turf-wars between different standards which both slowed their development, and reduced adoption. All potentially powerful actors in the application technology need to be involved in the standardisation process from the outset, both to be involved in the consensus building, and to add weight to, and promote dissemination of, the final publication.

Given the international basis of open standardisation it is not surprising that the various working group meetings were scattered across the world - Bologna, Budapest, Norway, Melbourne, Germany, Canada, Oregon etc. Few funding bodies directly support the creation of standards, and this example itinerary shows that a considerable travel budget is required from somewhere to make them possible.

3.2 Example 2 - The World Wide Web

The WWW provides a recent example of an organisational standard become popular with the global community, thereby becoming a *de facto* standard, but then having to enter into the open standards process to be both rationalised and incorporate new functionality desired by its users.

In 1990 when Tim Berners-Lee started to develop the WWW at CERN as an organisational standard to meet the immediate needs of European physics research, two open standardisation activities also began in hypermedia. The establishment of the Dexter Hypermedia Reference Model (Halasz and Schwartz, 1994) was being organised to create a standard reference model for hypertext systems, and the ISO HyTime standard for hypermedia (ISO, 1992b) was entering the ISO standardisation process.

The WWW became popular, and since the source code was available for a range of platforms, people started to use it outside CERN in 1992. HyTime became an ISO standard in 1992, while the Dexter model was not published until 1994.

HyTime has never been completely implemented, but was nevertheless revised into version 2 in 1997 after the usual five year review that ISO standards undergo, since it serves as a reference model for the functionality of hypermedia systems. Functionality which was only minimally available in the initial versions of the WWW protocols of HTML to describe pages, and HTTP as the transfer protocol.

By 1994 the WWW had become *the de facto* standard for distributed hypermedia, but there was a desire among users to increase the functionality in the WWW while maintaining a single interoperable system. Consequently, the World Wide Web Consortium (W3C) was founded in October 1994 to lead the WWW to its full potential by developing open standards that would promote its evolution and ensure its interoperability.

This development included establishing W3C recommendations to replace HTML by 1999 for three of the four aspects of the development process under consideration (see Figure 2).

A Document Object Model (DOM; W3C, 1998a) was created to define the API available to programme browsers - either directly from scripts or through declarative languages. Unfortunately, no reference architecture of the browser has been developed to support this DOM which leads to problems in developing such declarative data interchange languages to use it, since the stages of rendition are not stated.

A lower level interchange format than HTML was defined as XML (W3C, 1998b) which was a subset of the ISO SGML standard for document description that HyTime had been built upon. This allowed the separation of content from presentation style, facilitated the definition of any class of data, and provided a language on which the additional desired functionality could be built.

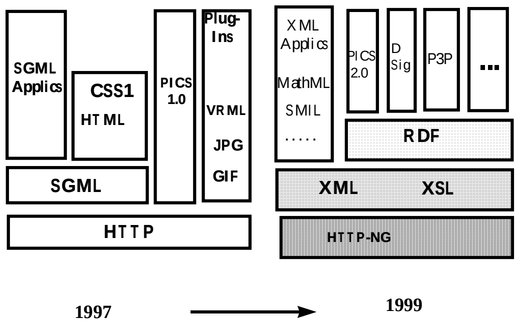

Figure 2: Changing Web technologies to improve performance and facilitate new applications.

On top of XML, a linking language (XLL) and style specification language (XSL) were developed to undertake these functions previously incorporated within HTML, but also to include additional functionality from HyTime. SMIL (W3C, 1998c) was also created to introduce the temporal synchronisation functions from HyTime, while MathML incorporated mathematical notations.

Other applications will progressively be built on top of the DOM to define data models (i.e. XML DTD's) for domain specific representations in a similar way (e.g. CML for chemical formulae) although as these become increasingly domain specific, they will be standardised by international professional bodies for each domain rather than through W3C or ISO. A similar subsidiarity mechanism is likely to be required for specialised vocabularies in NLP.

As XML describes data at a lower level that HTML, a language to describe conceptual metadata which HTML could not express has also been developed - Resource Description Framework (RDF; W3C, 1999). This is a semantic network like language which could have applications for intelligent agents and NLP.

This proliferation of data interchange and description standards from W3C has lead to overlap and conflicts both within W3C, and with ISO. Those in W3C who developed XML want to develop XML Schema to perform the same role as RDF, while retaining control of the recommendation. Those in ISO who had previously developed SGML and HyTime standards want to transform the W3C XML recommendation into an ISO standard under their control. The ISO JTC1/SC29/WG11 that developed the MPEG video encoding standards is now developing a Description Definition Language with Description Schemas to describe audio-visual content on the web in a search engine readable way as part of the MPEG-7 standard (ISO, 1998b), to perform part of the job that W3C's RDF does. The ISO TC 184 (Industrial Automation and Systems Integration), SC4 (Industrial Data) WG3 (Product Modelling) Technical Group 14 who developed ISO 10303 as a Standard for Exchange of Product Model Data (STEP) want to create an XML form of this under their control since XML on the web subsumes its function and is likely to have much wider adoption.

These overlaps will be harmonised; but they show the conflicts that arise between different groups within the open standardisation world over the control of open standards.

4. Reference Models & Architectures

The preceding sections have outlined the standardisation process, and argued the merits of reference architectures and models. This section describes the content of reference architectures in more detail, using three examples.

Reference models and architectures define the fundamental notions which tie together standards in an area. As in the case of the WWW standards, without a reference model, overlap and conflicts arise between the standards. Within the standardisation process it is hard to define the API and application data standards without a clear view of the processing architecture.

Consequently the reference model and architecture is most usefully produced at a stage in technology development before API's and interchange formats are defined. At this time, technology demonstrators are being produced by various research laboratories, and they need a basis on which to compare their performance and diagnostically relate it to their structure in order to refine the demonstrators towards industrial prototypes.

Two existing reference models are briefly described here as examples of what can be stated in the RM. The computer graphics reference model was chosen for the same reason that computer graphics was chosen in the previous section, that it is likely to be close in structure and purpose to a model for NLG. Intelligent multimedia presentation systems was chosen since it actually subsumes NLG at a high level, and describes an architecture into which any NLG architecture would have to integrate.

4.1 CG RM

The main purpose of the Computer Graphics Reference Model (CGRM; ISO 1992a) is to define concepts that shall be used to develop computer graphics standards. Additional purposes are to explain the relations between ISO graphics standards, and to provide a forum where areas outside computer graphics can identify their relationships to computer graphics. Each of these purposes could apply to a NLP reference model too.

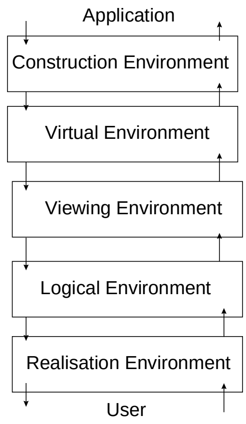

Figure 3: Computer Graphics Environments

The CGRM defines computer graphics in terms of five abstract levels called environments: construction, virtual, logical, and realisation (see figure 3). The internal model of each environment is identical. The symmetry between input and output in the diagram reflects a symmetry of purpose rather than a symmetry of complexity. The CGRM defines operations on data elements in each environment and provides a common internal architecture for each level, although this will not be described here.

External interfaces are defined with the Application and User, but also between the environments and Data Capture Metafiles, and between the application/CGRM interface and Audit Trail Metafiles in order to provide a record or playback of the flow of information across the application interface.

Rather than define the characteristics of each environment, only the top level Construction Environment will be used as an example: "In this environment, the application data to be displayed is prepared as a model from which specific graphic scenes may be produced. The application may only edit the model (composition) and collection store in the construction environment. Naming of parts of the collection is allowed and may be used to aid interaction. Input tokens in the instruction store are constructed in the precise form utilised by the application."

The properties of an output primitive specify its geometry and appearance. The nature of the actions performed in each environment implies some constraints on binding and consumption of properties. Again, for an example, the Construction Environment shall be used: "Since properties controlling the interaction of the application or the operator with the model have to be bound prior to being processed by this environment, such properties have to be specified at the time of creation of the output primitive. Properties of input tokens specifying the actions to be taken on the model in this environment have to be bound to the input tokens prior to their arrival in this environment."

Obviously, the Input Tokens and Output Primitives of the CGRM are defined, as are the Transformations which may change the form of the output primitives or input tokens within any environment (e.g. rotation, scaling, clipping).

Parallels to these environments, properties and property constraints, input tokens and output primitives, and the transformations performed at each level can be found in NLP and would be expected to be defined in any NLP reference architecture comparable in function or role with the CGRM.

4.2 IMMPS RM

The standard Reference Model for Intelligent Multimedia Presentation Systems (IMMPS RM; Bordegoni et al, 1997) was developed as part of the PREMO standardisation activity, although it is not part of the published PREMO standard (ISO, 1998a).

A first draft RM was initially developed by Thomas Rist and myself, while the editing effort throughout the development of the RM was provided by Salvatore Ruggieri. This draft was based on our personal experiences in having developed the IMMPS WIP (Whalster et al, 1993) and MMI[2] (Binot et al, 1990), as well as our awareness of other IMMPS produced. The draft was then firstly edited by an international team experienced in standards production, then presented to the wider community at a series of workshops in Europe, the US and Japan. This enlarged the group of people who felt some sense of ownership of the RM, and was intended to improve not only the RM, but also its chances of adoption. Further editing was required following this exposure as the set of existing IMMPS demonstrators which the RM accounted for grew. Finally, other authors of international standing who had attended the workshops further refined the draft into a final consensual form. The addition of further authors not only improved the RM, but was also intended to add authority to its final publication. The final publication of the RM was accompanied by a series of articles interpreting the architectures of existing IMMPS demonstrators within its terms. This was intended to illustrate the use of the RM to compare system architectures, to show that the RM was at least capable of being used to do so for a majority of existing systems, and to provide the initial core of such descriptions for a hopefully growing body of comparisons. This overall approach to developing a consensual RM, could provide a model for future developments in NLP.

The purpose of the IMMPS RM is to represent an implementation independent view of the processes required for the generation of multimedia presentations, including natural language. The reference architecture is defined in terms of layers, components and knowledge servers. In such generation, goals and commands are processed through the layers together with application data/knowledge, until they are eventually transformed into presentations composed of media objects viewed by a user. The RM focuses on the functions assigned to the layers and components, rather than on API's of IF's that may be employed to realise this functionality.

The core of the reference model is a generic layered architecture for IMMPSs. A layer serves as an abstract location for tasks, processes, or system components. A component is an objectification of a task, function or computing process in a real or abstract system. Components are essentially characterised by their particular input/output behaviour. The internal structure of a component may be given by means of a set of sub-components together with input/output relations between them. A connector between two entities enables (directed) interchange of information between them.

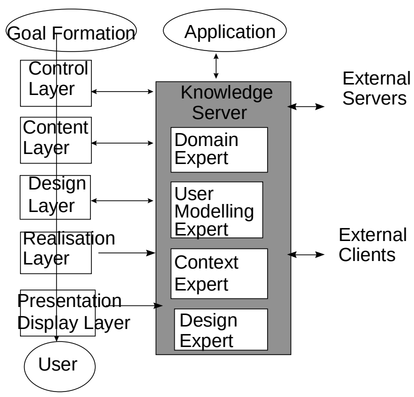

Figure 4: Changing Web technologies to improve performance and facilitate new applications.

Figure 4 outlines the IMMPS reference architecture. Its conceptual design reflects a modularization of the generation process into five layers, which represent particular subtasks: Control Layer, Content Layer, Design Layer, Realisation Layer, and Presentation Display Layer. In addition to their private knowledge, the layers can exploit explicitly encoded knowledge provided by a separate Knowledge Server, which is composed of four shared knowledge sources: Application Expert, Context Expert, User Expert and Design Expert.

The boundaries between the components of the RM and external entities are represented in figure 4 through ellipses labelled Goal Formulation, Application and User.

In the same way that the Presentation Display Layer is compatible with the CGRM and the PREMO presentation framework, it would support the adoption of the IMMPS RM if any reference architectures for natural language generation were also compatible with it. The complete RM will not be described here, but to this end, a brief introduction is given to the Design Layer and the Realisation Layer since they explicitly include design and realisation components for individual media including natural language.

An important task of an IMMPS is the transformation of media communicative acts into specifications of media objects together with a specification of the overall presentation layout. While it might seem straightforward to have two layers, one for the production of media objects, and a subsequent one for their layout, the RM adopt another structure based on the following observations: a) both media-specific production and presentation layout are complex tasks that can each be broken down into (at least) a design task and a realisation task; b) their is no justification for assuming media-specific production must necessarily precede presentation layout.

The Design Layer and Realisation Layer reflect these observations. By design the RM refers to the task of planning how to convey a certain communicative act by using a particular medium/modality and/or layout structure. The design task itself is further broken down into tasks for a Media Design component and a Layout Design component, as shown in Figure 5.

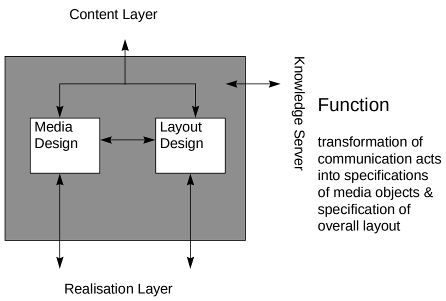

Figure 5:Design Layer.

The Media Design component, shown in more detail in Figure 6, contains a set of Media/Modality Specific Design components, which are dedicated modules for designing 2D and 3D graphics, natural language utterances, animation, video, music etc. A Dispatching Component is included to represent the task of dispatching media communicative acts to the relevant Media/Modality-Specific Design components, and to collect and pass on the results of these components. For the sake of generality, the RM does not provide fine grained descriptions of the Media/Modality-Specific Design components.

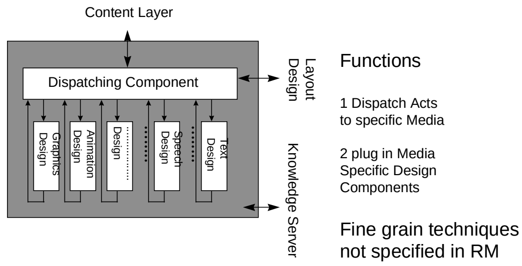

Figure 6: The media design component.

The Layout Design component determines the spatiotemporal arrangement of media objects in the presentation. The Design Layer does not prescribe a particular ordering in which media design and layout design must be carried out.

The results of the Design Layer are realisation plans, which are ordered sets of realisation commands to be executed by dedicated realisation components. We assume that the Media Design and Layout Design components have counterparts in the Realisation Layer's media realisation and layout components. Since these realisation components handle only media realisation commands and layout realisation commands respectively, the exchange of information between the Design Layer and the Realisation Layer is indicatedby two separate arrows in Figures 5 and 7.

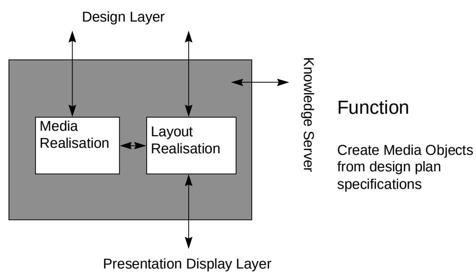

Figure 7: The realisation layer.

By realisation the RM refers to the task of creating from a design plan the specifications for concrete media objects and their layouts. The realisation layer shown in Figure 7, and the media realisation component shown in Figure 8 consisting of Media/Modality-Specific Realisation components mirror the respective structures in the Design Layer.

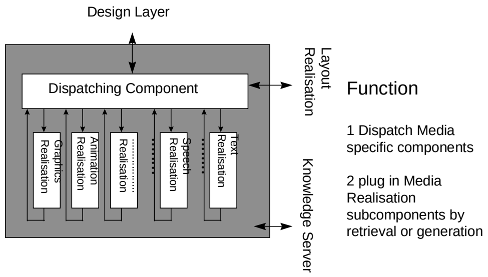

Figure 8: The media realisation component.

The internal structure of the Media/Modality-Specific Realisation components is not considered in the IMMPS RM, although it is noted that they can perform the function through either generation, or retrieval.

The result of the Realisation Layer is a specification of displayable media objects together with layout information. It entails all the information required to run the presentation properly.

This brief description of the overall structure of the IMMPS RM, and how Media/Modality-Specific components are incorporated should provide guidance as to the compatibility of proposed natural language generation architectures. Equally, it raises issues of control, layout and presentation display which may not be considered in such architectures, yet are essential in the complete definition of an application architecture.

5. Conclusion

The development of a reference architecture and reference model for natural language processing, and natural language generation in particular would lead to several benefits. Firstly, it would allow the comparison of the architectures of existing and future demonstrators, to determine the source of their benefits. Secondly, it would lay the basis for the development of future standards for API's and Interchange Formats. Thirdly, it would speed the development of commercial products, and reduce the risks in their development.

The development of such a reference model requires consensus among the community and cannot be thrust upon them, if it is to be adopted. The development of this consensus in turn requires considerable effort, whether through informal organisation, or through formal standards bodies.

In developing a reference model, constraints are placed on such a reference model by the expectation of compatibility with both the structure and content of existing reference models and standards.

Despite the constraints and effort required to reach consensus, the benefits in developing a reference model supported by the community outweigh the problems, if the theoretical diversity in the natural language processing community can be overcome.

Acknowledgements

Thanks to Prof. David Duce whose experience of the standardisation process provided most of the content in this paper.

References

D.B. Arnold and D.A. Duce, *ISO Standards for Computer Graphics: The First Generation*, Butterworths, London, 1990.

J-L Binot, P Falzon, R Perez, B Peroche, N Sheehy, J Rouault, M D Wilson Architecture of a multimodal dialogue interface for knowledge-based systems *Proceedings of Esprit 1990 Conference* p.412-433 Kluwer Academic Publishers, Dordrecht, (1990)

M Bordegoni, G Faconti, S Feiner, MT 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) p.477-496, 1997.

F. Halasz and M. Schwartz. The Dexter Hypertext Reference Model. *Communications of the ACM*, 37(2):30-39, 1994.

ISO. *Computer Graphics Reference Model (CGRM),* International Standards Organisation, ISO/IEC IS 11072, 1992a.

ISO. *Hypermedia/Time-based Structuring Language (HyTime)*, International Standards Organisation, ISO/IEC IS 10744, 1992b.

ISO. *Presentation Environments for Multimedia Objects (PREMO)* International Standards Organisation, ISO/IEC IS 14478, 1998a.

ISO. *MPEG-7 Context and Objectives*, International Standards Organisation, ISO/IEC JTC1/sc29/WG11 N2460, (http://drogo.cselt. stet.it/mpeg/standards/ mpeg-7/mpeg-7.htm), 1998b.

W3C *Document Object Model (DOM) Level 1 Specification Version 1.0* http://www.w3.org/TR/REC-DOM-Level-1/, 1998a.

W3C *Extensible Markup Language (XML) 1.0* http://www.w3.org/TR/1998/REC-xml-19980210, 1998b.

W3C *Synchronized Multimedia Integration Language (SMIL) 1.0 Specification*, http://www.w3.org/TR/REC-smil/, 1998c.

W3C, *Resource Description Framework (RDF) Model and Syntax Specification*, http://www.w3.org/TR/REC-rdf-syntax/, 1999.

W. Whalster, E. Andre, W. Finkler, H-J. Profitlich, T. Rist, Plan-based integration of natural language and graphics generation, Artificial Intelligence, 63: 387-427, 1993.