42 STANDARDIZATION: A CRITICAL VIEW

Rudi Stouffs, Ramesh Krishnamurti[1]

Faculty of Architecture, Delft University of Technology
[1]Department of Architecture, Carnegie Mellon University

Abstract

Standardization is often touted as the ultimate solution for electronic data exchange. With respect to design, it is arguable that while standardization can improve the design process through effective data exchange, it may also hinder the process by imposing a specific language for designers to express their ideas and concepts. With respect to architectural design, to support the dynamic nature of design, flexibility and extensibility are factors that must be considered in any standardization effort. We contend that a syntactic approach specifying a framework for expressing and comparing various representations holds more promise. In this paper, we describe a framework for representational flexibility, which borrows from various existing approaches and technologies.

Keywords: *standardization, representations, flexibility, extensibility, syntactic*

THE QUESTION OF STANDARDIZATION

Standardization is generally considered to be the ultimate solution to the problem of data exchange among collaborative partners. In non-digital exchanges, standardization has not been much of an issue, except, perhaps, at a national level. Project partners exchange information with little attention to or the need for any global standard, relying instead, either on their own sources of knowledge in order to interpret exchanged data, or querying their colleagues for an explanation. In digital exchanges, there are implicit assumptions made on the capabilities of any computer application. However, one cannot rely on interpretive knowledge being resident in an application. Neither, even assuming that common sense usually eludes software, can one expect nor require the user to assist applications in interpreting this data. As a result, in order to resolve the data exchange problem, a standardization approach has been at the core of most research efforts. The assumption is simple: if everyone adopts the same concepts, vocabulary, and language, any data expressed within this language will be accessible to everyone.

With respect to design, the real question becomes whether standardization will improve the design process through effective data exchange or, instead, would it hinder the process by imposing a specific language for designers to express their ideas and conceptualizations? A standard vocabulary or language facilitates communication between different disciplines in the building industry. It may enable new streams of communication between partners from remote knowledge domains. Representationally, a standard for data exchange reduces the need to concur on a set of software applications for all design partners to adopt. Instead, partners in each discipline may employ software that best suits their needs and processes, that is characteristic of their discipline. Thus, a variety of applications might be employed interchangeably, provided these all adhere to the same standard.

At the same time, a standard vocabulary or data model may act as a straitjacket enabling only certain forms of communication, possibly denying certain solutions, or even impeding creative new approaches to a given problem. One advantage from having a lack of standards is that the range of applications a project partner can choose from is intrinsically indefinite. In the design stage, especially, a creative approach may necessitate the creative selection of design software. This may include using design applications from other design domains and even putting to novel use different types of applications or tools. For example, CAD software such as CATIA, first developed for the aerospace industry, and Maya, developed for the animation and movie industry, have been used for architectural design. If there is, however, an insistence that electronic data adhere to a specific standard, this then may strongly constrain the selection of software applications. It may further require the author of any information to recast it into a different format, resulting in a possibly tedious and uninteresting conversion process.

Two practical considerations are important here. The first is the time it takes to develop and have accepted a standard that covers the disciplines and domains within the AEC industry, and, perhaps, beyond. The second is the rate of development of software solutions that adhere to this evolving standard, that also respond to evolving needs of designers and other users and for tools to support them in their creative as well as minutiae work. With respect to architectural design, given the dynamic nature of design, additionally, flexibility and extensibility must be considered in a standardization approach. For these reasons, a syntactic approach specifying a framework for expressing and comparing various representations holds more promise.

DEVELOPMENT AND ACCEPTANCE OF STANDARDS

Standards are difficult to develop and, more so, to have accepted within the industry. This difficulty is primarily due to the extended range of disciplines and knowledge domains that are involved and the need for a broad consensus among industry members. This is particularly the case in the building industry where such consensus is hard to achieve both within and between disciplines. There are many reasons for this. Most commonly, the fragmented nature of the building industry and the uniqueness of each building project (Buckley *et al.*, 1998) are cited as primary causes for the failure to achieve a standard for data sharing among project partners. On the other hand, partial standards have a much higher chance of success. These may be achieved by focusing on a particular aspect of the building project such as a single phase in the design or construction process, a specific knowledge domain, or a particular set of interactions between project partners. By isolating this aspect from the broader context, standardization processes that have proven to be successful in other fields may apply and yield a usable solution to a limited form of the data exchange problem.

Recent standardization efforts show a renewed interest in such solutions. For example, *Object trees* (van Nederveen, 2000) constitute an approach primarily aimed at the construction planning phase of large-scale construction projects. Object trees serve to improve electronic communication between different disciplines by offering participants a methodology for developing representational object trees that correspond to concept hierarchies of construction aspects, elements and their attributes. The methodology requires all participants to concur on the concepts and attributes involved; in return, it presents them with a unified framework for relating activities and for data exchange among participants. It is specifically suited for the construction and construction planning phases of large-scale projects in which the advantages of the conceptual and representational framework far outweigh the disadvantages of the need for an a-priori consensus. Distinct from many other standardization approaches, the object trees methodology does not impose a concept vocabulary. Instead, it leaves the definition of the vocabulary up to the project participants, typically based upon their prior experience and on the project specifics.

The E-Construct project is another example of a standardization effort, in which a particular need in the building construction industry is recognized and a solution developed: bcXML, an extension of XML, is a Web-based universal format for structured documents and data to support e-commerce in the E.U. building industry, taking into account national languages, classification and code systems (Tolman and Böhms, 2000). It is perhaps characteristic of recent efforts that both examples are concerned with aspects related to the construction phase, in particular, construction planning and tendering, and avoid building design phases. These examples are illustrative of the additional difficulty of developing effective standards to support the design stages.

SUPPORT AND INTEGRATION OF STANDARDS

Development of standards is further complicated as design cannot be completely rationalized nor captured into specific processes. By nature and context, architectural designs are ill defined and poorly constrained; architects find themselves having to conceive and create novel solutions. In this creative environment, design applications serve as tools for exploring new ideas both in form and function. Preferably, such explorations are neither restricted nor hindered by software functionality and their underlying representations. For this reason, designers may adopt applications from outside their design domain, to extend their ability to express their ideas and methods. If designers are bound to a standard representation for data exchange, it may undermine their ability to exploit new tools, in the design process, that do not adhere to this representation.

It is unrealistic to expect the software industry to stay abreast of demands in supporting new design techniques and methodologies. Recent history shows that the AEC CAD software industry, in particular, is far from being a frontrunner when it comes to developing new support for creative form finding. Evolutions in design are led by those who are willing to step outside conventional approaches and explore new design techniques and technologies. When faced with project partners requiring an adherence to specific standards for electronic data exchange, such pioneers may be forced to invest time and effort in developing the necessary translation support. Depending on the current status of the standard, it may not even be possible to develop such support that is fully satisfactory with respect to the translation needs. Consider, as an example, the current "standard" for the exchange of CAD drawings in the AEC industry, DXF. In the case of translation from Maya to DXF, the corresponding facility can only consider that part of the data that is understood by the DXF format and has to omit information about texture and NURBS geometry. Unless the standard is fully compatible to the concepts or techniques underlying the outside tool or application, the designer may be forced to forego its use.

In the extreme, it could be argued, with the danger of antagonizing those who are leading the drive for standardization, that this process of standardization supports the software industry in their struggle for self-preservation. The acceptance of a single standard for the building industry would afford AEC CAD software developers as before to lag behind in their inclusion of new features and techniques. A standard to which almost everyone adheres offers them the necessary time to respond to influences that find their origin in other design domains, without the fear of being left at the wayside of such evolutions. CAD software developers active within other disciplines may not be tempted to invest in a standard that is of little or no importance to their target market. Though this target market may shift or expand as a result of outsiders exploring the effectiveness of the software application for their purposes, such market changes do not occur overnight.

At the same time, software providers argue that it is the industry's responsibility to develop and agree on a standard for data exchange. When applied to a conceptual standard, their argument makes sense. After all, concepts are defined by the industry members, and are merely adopted by the software developers. On the other hand, software providers have better knowledge of the technical possibilities for standardization and have the ability to pull the standardization process by adopting and integrating standardization efforts early on. Currently, the AEC software industry plays an important role within the International Alliance for Interoperability (IAI). Standardization efforts of the IAI have resulted in a specification of Industry Foundation Classes

(IFCs) defining a building object model shared by all IFC-compliant applications (Bazjanac, 1998). The IAI also supports the aecXML Working Group, which is working on an extension to XML in order to facilitate electronic communication primarily in the U.S. architectural, engineering, and construction industries (aecXML, 1999).

FLEXIBILITY AND STANDARDIZATION

Though different areas may require different solutions for data exchange, when it comes to architectural design, one must consider flexibility and extensibility in the solution in order to support the dynamic nature of design (van Leeuwen and de Vries, 2000; Stouffs and Krishnamurti, 2001a, 1996). Effective flexibility and extensibility with respect to a representational format will enable designers and developers to alter this format, to various extents, in order to support new design tools and descriptions. These must also ensure the ability to exchange data in this new format with other applications and participants in the design process. Though a standard vocabulary will enable all participants to effectively communicate and exchange data within the context of this standard, it will not support such flexibility and extensibility. Instead, a representational framework is needed that encourages participants to express their design information in such a way that data exchange is supported to the best extent possible.

Such a framework will most likely be based on a representational language for expressing various information models, together with a collection of tools to compare these models and exchange information between these. Various standardization approaches already encompass a descriptive language for the development of different product models, even if their main emphasis may be on the conception of a semantic model. The EXPRESS modeling language from the ISO STEP development (ISO, 1994) and to some extent the Internet Foundation Classes of the IAI serve as examples. However, the purpose of EXPRESS is to serve as a tool for specifying product models and, as such, does not provide support for exchanging information between these models. While the specification and sharing of object classes, in the IFC, may support data exchange to some extent between various models, the adoption of an object model is in itself insufficient to support effective data exchange.

The situation might be remarkably different if a standard could be designed that would encompass almost all design disciplines. It would enable cross-fertilization between design disciplines in terms of tools and applications as well as techniques and methodologies. However, as it already proves to be so difficult to achieve a standard for the AEC industry, even conceptually, it is indefinitely more difficult to design such a global standard. Even if such a standardization effort bears fruit, it can be expected that due to the immense challenge, the result will be a syntactic language of representational objects and relationships rather than a semantic model of conceptual entities and their relationships.

SYNTACTIC STANDARDIZATION

In order to support representational flexibility and extensibility, a framework must be conceived and developed that provides support for exploring alternative design representations, for comparing design representations with respect to scope and coverage, and for mapping design information between representations, even when scopes are not identical. A representation can be considered as a structure of primitive data entities and compositional relationships (Stouffs *et al.*, 1996). Comparing different representations, therefore, requires a comparison of the primitive components as well as the overall compositional structures. On the other hand, the expressive power of a representational framework is defined by its vocabularies of primitive data types and compositional relationships. By carefully selecting the vocabulary of compositional relationships, designers can be given the necessary freedom and flexibility to develop and adopt representations that serve their intentions and needs. At the same time, these can be formally compared with respect to scope and coverage in order to support information exchange. Such a comparison will not only yield a possible mapping, but also uncover potential data loss when moving data from less restrictive to more restrictive representations. Translation services can be provided based on syntactic similarity, next to semantic identity.

The Lexicon model suggests a syntactic approach. Though as part of a semantic model, it considers a semi-syntactic approach in which concepts are unambiguously defined by their constituent attributes (Woestenenk, 1998). These attributes then comprise the primitive concepts that define the semantic vocabulary of this model. Taking this descriptive approach one step further, the attributes can be described syntactically, leading to a purely syntactic description of the concepts as compositions of primitive data types. Within a formal structure, these syntactic descriptions may be compared independently of their conceptual meanings, thus allowing for synonym concepts.

XML too offers such a formal framework. XML can be considered a meta-language that serves to define markup languages for specific purposes. By specifying a grammatical structure of markup tags and their composition, a markup language is defined that can be shared with others. When project partners agree on tags, they can exchange data described in any markup language based on these tags, even when their own markup language differs in scope or composition. XML has the advantages that it is readable both by humans and by the computer. Markup languages based on XML can easily be adapted or extended to specific purposes or needs. In this way, XML allows for syntactic standardization, providing all participants with the ability to define or adopt their own data model, and considering ways of translating these different models between one another at a later stage, using tools developed for this purpose. Already, XML may be considered to lead the way to such a standard syntax, as a number of standardization and product modeling efforts are “grafting” themselves onto XML (Tolman and Böhms, 2000; aecXML, 1999). It has been argued, however, that the use of XML to create standards misunderstands the real power of XML (O’Brien and Al-Biqami, 2000).

AN ECLECTIC APPROACH

XML is particularly suited to structure otherwise unstructured information, such as textual data, and to organize information available on the Web. However, it does not provide any information on how to manipulate the data and, as such, is ill suited to represent detailed graphical or geometrical data (Tunçer and Stouffs, 2000). Instead, a framework for supporting representational flexibility may be conceived by borrowing from different approaches in order to combine their respective advantages. From XML, it may inherit a foundation consisting of an extensible vocabulary of data components that can be composed hierarchically into a representational language. From the IFC effort, it may borrow the object-oriented approach, defining the data components as objects that encapsulate both the data structure and the operations defined on these structures. The symbiosis of these two approaches requires that the compositional operators be defined so that any compositional structure offers the same functionality as each component object separately. Hereto, a behavior can be defined for every component and structure as a collection of common operations on these structures for creation or deletion, or the merging of structures under some formal operations. Through a careful definition of the compositional operators, structures may derive their behavior from their components in accordance to the compositional relationship.

Similar to the IFC approach, a language specification can be derived on two levels. A first syntactic level specifies the vocabulary of primitive object classes and their respective behaviors. This behavior, in itself, does not provide any meaning to the object class. In fact, the same data structure may define two or more object classes, if as many different behaviors can be said to apply to different purposes. On a second level, a selection of object classes is defined and, individually, named in order to express a semantic concept. These named classes can, subsequently, be composed into a hierarchical structure in order to define an appropriate representational schema. In contrast to the IFC approach, users can specify this semantic concept and the representational structure composed accordingly. Alternative representations can be defined by altering the compositional structure or the selection of component classes. As each representation defines the same common operations, these can be reasonably plugged into an applicative interface for manipulation.

Sorts (Stouffs and Krishnamurti, 2001b, 1997) specify such a framework for representational flexibility. Elementary data types define primitive sorts that combine to composite sorts under formal compositional operations. Examples of such operations are an operation of sum, allowing for disjunctively co-ordinate compositions of sorts, and an attribute relationship, providing for (recursively) subordinate compositions of sorts in both one-to-many and one-to-one instantiations. The result is a constructive, hierarchical description of sorts as compositions of other sorts, where each leaf node specifies a primitive data type and every other node defines a compositional operation on its operand children nodes (figure 1).

The definition of a sort includes a specification of the operational behavior of its members and collections, denoted as forms. The behavioral specification enables a uniform handling of forms of different sorts, on the proviso that the universe of all forms of a sort is closed under the respective operations. Primitive sorts have their behaviors assigned in order to achieve a desired effect, e.g., discrete behaviors for points and labels, an interval behavior for line segments, and an ordinal behavior for weights such as thickness or tones. On the other hand, a composite sort receives its behavior from its component sorts, based on its compositional relationships (Stouffs and Krishnamurti, 1997). The formal relationships between sorts enable the comparison and mapping of sorts as representational structures; the behavioral specification of sorts supports the mapping of information structures onto different sorts, such that the resulting information structures conform to the definition of the respective sorts or representations.

The concept of sorts only specifies a common syntax, allowing for different vocabularies and languages to be created, and providing the means to develop translation facilities between these. For example, a point may be specified with any number of coordinates depending on its dimensionality, its coordinates may constitute integers, reals, or rationals, these may be bounded in space, etc. Sorts can be defined accordingly and, based on their compositional structures, compared and related. For example, the operation of sum specifies a subsumption relationship on sorts, where one sort may match a part of another sort, under sum (Stouffs and Krishnamurti, 1997). Compositional structures under the attribute relationship, if not equal, may be fully (or partially) convertible: the attribute relationship is associative though not commutative. Based on the result of this comparison, translation support can be provided for and data loss monitored. For example, partial conversions always result in data loss; complete conversions may result in data loss depending on the behavioral categories of the constituent sorts.

```
sort conceptrefs : (concepts : [Label]);
sort (hasrefs, isrefs) : [Property] (concepts, conceptrefs);
sort concepttree : concepts ^ concepttree + concepts ^ hasrefs + concepts + conceptrefs ^ isrefs
```

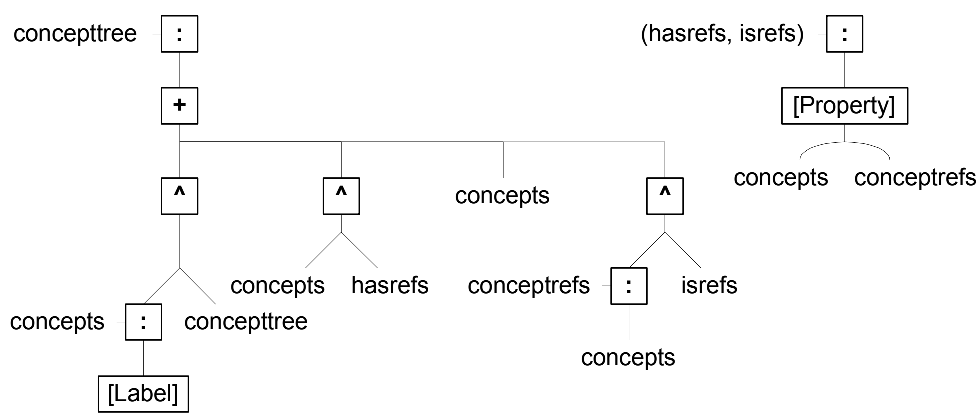

Figure 1. Textual and graphical definition of a recursive concepttree sort. A concepttree may include multiple instances of a single concept, with one instance defined and referenced by all other instances. '+' and '^' denote the operations of sum and attribute, respectively. ':' denotes the naming of a sort. 'Label' and 'Property' are primitive sorts; the latter defines a property relationship sort between two given sorts.

CONCLUSION

Standardization is not the obvious solution to data exchange in the design process, if it attempts to impose a common semantic model for all to adhere to. Instead, design participants should be offered tools to map representations in order to assist electronic data exchange. In this way, each participant can decide for herself whether to rely only on common software applications that are known to fully support this representational standard or to explore new tools and develop translation support for data exchange with others. Such a representational framework must offer a large collection of different representational building blocks and compositional relationships in order to develop a variety of different representations, and tools for comparing and mapping representations and converting data between representations. A range of predefined representations can be used as targets for comparing and mapping representations and as a basis for developing new representations through alterations and extensions. In this way, a large variety of data exchange situations can be resolved and representational freedom may be supported to various extents, depending on the effort one is willing to make.

ACKNOWLEDGMENTS

The research on sorts is partly funded by the Netherlands Organization for Scientific Research (NWO), grant nr. 016.007.007. The authors would like to thank Bige Tunçer for reviewing various drafts of this paper.

REFERENCES

aecXML, 1999, *AecXML: A framework for electronic communications for the AEC industries*, aecXML White Paper, IAI aecXML Domain Committee. http://www.aecxml.org/technical/

Bazjanac, V., 1998, "Industry Foundation Classes: Bringing software interoperability to the building industry", *The Construction Specifier*, 6/98, p. 47-54.

Buckley, E., A. Zarli, C. Reynolds, and O. Richaud, 1998, "Business objects in construct IT", in: R. Amor (ed.), *Product and Process Modelling in the Building Industry*, Building Research Establishment, Watford, England, p. 117-130.

ISO, 1994, *ISO 10303-1, Overview and fundamental principles*, International Standardization Organization, Geneva, Switzerland.

O'Brien, M.J. and N. Al-Biqami, 2000, "XML, flexibility and systems integration", in: G. Gudnason (ed.), *Construction Information Technology 2000*, Vol. 2, Icelandic Building Research Institute, Reykjavik, Iceland, p. 656-661.

Tolman, F.P. and H.M. Böhms, 2000, "Electronic business in the building-construction industry: preparing or the new Internet", in: G. Gudnason (ed.), *Construction Information Technology 2000*, Vol. 2, Icelandic Building Research Institute, Reykjavik, Iceland, p. 928-936.

van Leeuwen, J. and B. de Vries, 2000, “Capturing design knowledge in formal concept definitions”, in: *Design and Decision Support Systems in Architecture, Proceedings of the 5th International Conference*, Eindhoven University of Technology, p. 220-237.

van Nederveen, G.A., 2000, *Object trees: improving electronic communication between participants of different disciplines in large-scale construction projects*, Delft University of Technology, Delft, The Netherlands.

Stouffs, R. and R. Krishnamurti, 2001a, “Flexibility and dynamism in digital design representations”, in: R. Beheshti (ed.), *Advances in Building Informatics*, Europia Productions, Paris, France.

Stouffs, R. and R. Krishnamurti, 2001b, “On the road to standardization”, to appear in: B. de Vries (ed.), *CAAD Futures 2001*, Kluwer Academic, Dordrecht, The Netherlands.

Stouffs, R. and R. Krishnamurti, 1997, “Sorts: a concept for representational flexibility”, in: R. Junge (ed.), *CAAD Futures 1997*, Kluwer Academic, Dordrecht, The Netherlands, p. 553-564.

Stouffs, R. and R. Krishnamurti, 1996, “The extensibility and applicability of geometric representations”, in: *3rd Design and Decision Support Systems in Architecture and Urban Planning Conference, Architecture Proceedings*, Eindhoven University of Technology, Eindhoven, The Netherlands, p. 436-452.

Stouffs, R., R. Krishnamurti, and C.M. Eastman, 1996, “A formal structure for nonequivalent solid representations”, in: S. Finger, M. Mäntylä and T. Tomiyama (eds.), *Proceedings of IFIP WG 5.2 Workshop on Knowledge Intensive CAD II*, International Federation for Information Processing, Working Group 5.2, p. 269-289.

Tunçer, B. and R. Stouffs, 2000, Modeling building project information, in: G. Gudnason (ed.), *Construction Information Technology 2000*, Icelandic Building Research Institute, Reykjavik, Iceland, p. 937-947.

Woestenenk, K., 1998, “A common construction vocabulary”, in: R. Amor (ed.), *Product and Process Modelling in the Building Industry*, Building Research Establishment, Watford, England, p. 561-568.