4º SIGraDi'2000 • Rio de Janeiro
constructing the DIGITAL space

Alternative Computational Design Representations

Rudi Stouffs
r.stouffs@bk.tudelft.nl
Faculty of Architecture, Delft University of Technology, Delft, The Netherlands

Ramesh Krishnamurti
ramesh+@andrew.cmu.edu
Department of Architecture, Carnegie Mellon University, Pittsburgh, Pa., USA

Abstract

Supporting data sharing among different disciplines, applications, and users in the building industry is a complex and difficult task. Standardization efforts and research into product models have since long attempted to facilitate data exchange among building partners, with little result so far. Different technologies have resulted in different approaches, in particular, an object-oriented approach has led to the specification of IFCs as a basis for information sharing, while other initiatives adopt XML as a flexible language for marking up and describing project information. We propose a concept for representational flexibility, named sorts, that combines many of the advantages of both approaches. Based on an extensible vocabulary of representational classes and compositional relationships and grounded in an object-oriented framework that has each of the representational classes specify its own operational behavior, it will enable a designer to define, develop, and adopt alternative design representations that can suit a specific purpose or task at hand.

Resumen

Apoyar el compartir datos entre diferentes disciplinas, aplicaciones, y usuarios es una tarea compleja y difícil en el sector de la construcción. Desde hace años no se escatiman esfuerzos para facilitar el intercambio de datos entre socios constructores por medio de estandarizaciones e investigaciones de modelos de producto, hasta el presente sin embargo, con poco resultado. Tecnologías diferentes han dado lugar a diferentes formas de proceder, en particular, un acercamiento 'orientado hacia objetos' ha llevado a la especificación de IFCs como una base para compartir información, mientras otras iniciativas adoptan XML como un lenguaje flexible para marcar y describir información de proyectos. Proponemos un concepto para la flexibilidad representacional, llamado sorts, combinando muchas de las ventajas de ambos acercamientos. Basado en un vocabulario extensible de clases representacionales y relaciones composicionales y fundado en un marco 'orientado hacia objetos', que deja cada uno de las clases del representacional que especifique su propia conducta operacional, le permitirá a un diseñador que defina, desarrolle, y adopte representaciones de diseño alternativas que pueden servir para un propósito o una tarea específico actual.

Data exchange and standards

Building projects commonly involve a large number of participants from various disciplines. Between and within disciplines, building partners use a variety of applications based on many distinct data formats. This diversity makes supporting data sharing within a building project a difficult task. Various approaches to facilitate data exchange exist, based on different techniques and technologies. The most obvious approach is to develop a specific utility for translating data between two given formats. Despite attempts at developing alternative approaches, this is still the most widely used. The advantages are clear: the single purpose supports a focused development that emphasizes the nature of either or both formats or the specifics of the context in which the utility will perform. Such a utility may be used stand-alone or integrated into an application that uses either data format, e.g., in the form of an import or export functionality, or into a system that offers multiple translational facilities.

Most often, such utilities serve data sharing in conjunction with a standard or pseudo-standard. Consider DXF, a data exchange format developed for AutoCAD™ 's own translational needs between subsequent software versions. This format was adopted by the CAD software industry as a pseudo-standard for data exchange. By integrating import and export functionalities from and to DXF into every application, the format serves as a standard for data sharing among CAD applications. This approach has the advantage that each application needs to support translation only between a single pair of data formats, i.e., the proprietary format and the standard. However, pseudo-standards are ill-suited to this task. As they were never developed for this task, they neither reflect on the nature of the proprietary format nor on the context of the exchange. For example, DXF supports neither NURBS, a popular geometric representation for curved surfaces, nor textures, making it ill-suited for sharing advanced 3D modeling data.

General standards for exchanging building data may overcome these limitations. However, standards are difficult to establish as they require a general consensus among industry members. Particularly in the building industry, such a consensus is hard to achieve. Many reasons can be thought of. Most commonly, the fragmented nature of the industry and the uniqueness of each building project (Buckley et al. 1998) are mentioned as primary reasons for this failure.

However, hope has not yet faded. New approaches based on advances in software technology have resulted in increased efforts and better chances of success. Next to the efforts to reach an ISO STEP standard for the exchange of product model data (ISO 1994), an International Alliance for Interoperability (IAI) has been founded within the building industry that aims to define an object-oriented data model, in the form of Industry Foundation Classes (IFCs), as a basis for project information sharing (Bazjanac 1998). Other efforts are adopting XML for facilitating data exchange (aecXML 1999, Tolman 2000).

Alternative design representations

While these standardization efforts all have more or less the same aim, their strategies are quite different and, with it, the advantages and disadvantages of each in supporting flexibility in data formats. The necessity for data exchange support reflects on a desire to use alternative design representations that enable a particular expression, analysis, or organization. While translational utilities can support data exchange between the standard, on the one hand, and proprietary data formats, on the other hand, there's a limit to what can effectively be catered for in this way. Advances in techniques and technologies repeatedly require new representations of a same building component or aspect. Standards, however, are necessarily based on current knowledge, uses, and needs. The difficulty of establishing a standard and having it adopted by the market almost inhibits any subsequent changes in order to update it to new requirements. Unless, such flexibility is built into the technology.

The IFC effort attempts to overcome this difficulty by adopting an object-oriented approach and envisioning an evolving object model. Objects encompass both the data and the operational access to this data. Applications can use this model to define the underlying representation, or incorporate a translation from and to this model into their functionality. When the model is subsequently extended, a corresponding adaptation of the applications may not be necessary, unless to integrate the additional functionality provided. In this manner, a single model can respond to advances in knowledge and technology. At the same time, however, this model still depends on a consensus and, as such, will not be able to support the entire spectrum of alternative design representations that can suit particular users or specific situations. Furthermore, access to this model is only available to software developers and, as a result, a designer will be restricted to what is available on the market rather than be able to exploit the potential of a truly flexible standard.

XML may offer this flexibility. XML is 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 can agree on the tags, they can exchange data described in any markup language based on these tags, even if 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 to one's own specific needs.

Representational flexibility

Markup languages are particularly suited to structure otherwise unstructured information, such as textual data, and to organize information available over the Web. However, they do not provide any information on how to manipulate the data and, as such, are ill-suited to represent geometrical data. Instead, a framework for supporting representational flexibility may borrow from all these 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 this structure. The symbiosis of these two approaches requires the compositional operators to be defined such 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, a same data structure may define two or more object classes if as many different behaviors apply. 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 representation. Alternative representations can be defined by altering the compositional structure or the selection of components. Translational services can be provided based on semantical identity and syntactical similarity. Each resulting representation defines the same common operations and as such, can be reasonably plugged into an applicative interface for manipulation.

We are developing such a framework for representational flexibility, named **sorts**. Elementary data types define primitive sorts which combine to composite sorts under formal operations (Stouffs and Krishnamurti 1997), e.g., respective operations of sum and attribute allow for co-ordinate and subordinate compositions of sorts, the latter in both one-to-many and one-to-one instantiations. The definition of a sort also includes a specification of the operational behavior of its members and collections thereof for common arithmetic operations. Whereas the formal compositional relationships enable the comparison and mapping of sorts as representations, the behavioral specification supports the mapping of data onto different sorts such that the resulting data is conform the definition of the respective representation.

Alternative design representations can be defined as variations on a given sort, by altering the components or the composition. As an example, consider a representation for a collection of drawings given a sort that defines a single drawing. By specifying an attribute composition with a sort of labels, a named collection of drawings is enabled similar to a set of layers in a CAD application. Alternatively, by specifying an attribute composition with a sort of points or rectangles, a layout can be represented for these drawings. One step further, this sort can be modified to enable drawings to relate to parts within other drawings, allowing for detailing relationships to be specified in this layout.

The concept of **sorts** aims to provide almost continuous support to evolving representations, providing for an environment that supports exploration and trial, even with respect to the representation. By specifying only a common syntax, it allows for different vocabularies and languages to be created, and provides the means to develop translational facilities between these. There is no imposition of concepts beyond the purely syntactical, and the alphabet of building blocks can be readily extended at all times.

References

aecXML. (1999). AecXML: *A framework for electronic communications for the AEC industries*, 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, 47-54.

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

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

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

Tolman, F.P. and H.M. Böhms (2000). Electronic business in the building-construction industry: preparing for the new Internet, *Construction Information Technology 2002*, Vol. 2 (ed. G. Gudnason), Icelandic Building Research Institute, Reykjavik, Iceland, 928-936.