What's in a Representation, Why Do We Care, and What Does It Mean? Examining the Evidence from Psychology

Scott Johnson
University of Michigan
sven@umich.edu

This paper examines psychological evidence on the nature and role of representations in cognition. Both internal (mental) and external (physical or digital) representations are considered. It is discovered that both types of representation are deeply linked to thought processes. They are linked to learning, the ability to use existing knowledge, and problem-solving strategies. The links between representations, thought processes, and behaviors are so deep that even eye movements are partly governed by representations. Choice of representation can affect limited cognitive resources like attention and short-term memory by forcing a person to try to utilize poorly organized information or perform "translation" from one representation to another. The implications of this evidence are discussed. Based on these findings, a set of guidelines is presented for digital representations which optimize displays of cognitive resources. These guidelines describe what sorts of characteristics and behaviors a representation should exhibit, and what sorts of information it should contain in order to accommodate and facilitate change. Current attempts to implement such representations are discussed.
Introduction

All computer software intended for drafting, modeling, or design—indeed, all software intended for any purpose whatsoever—has some sort of inherent way of representing whatever it is that the program is meant to manipulate. In the case of computer-aided design software, representations are often based on graphic mathematical entities meant to facilitate coordinate transformations and other graphics routines, but we often find complaints that such software is unfriendly or difficult to use.

Potentially, such software can be made easier to use if we base the software on representations that facilitate human cognition and design. In order to do this, we must have some understanding of how the human mind works and how it uses representations. What are representations, and how are they used? What are the effects of using alternate representations? What are the implications for architectural design systems? In other words, what's in a representation, why do we care, and what does it mean?

background

the mind and cognition

Cognitive psychologists describe the mind as being comprised of certain resources and mechanisms. These components behave in certain well-documented ways, with certain inherent limitations. Components include short-term (working memory), long-term memory, and attention, as well as various buffers to handle sensory input, and other mechanisms.

short-term memory

Short-term (STM) or working memory holds information which is currently in use. We use it when we repeat a phone number to ourselves in preparation of dialing, when we keep track of open nods and partial results as we perform arithmetic operations, when we listen to a lecture, when we compose a sentence, or when we imagine a scene. Information in short term memory decays in a matter of seconds unless it is constantly rehearsed, in which case it can last minutes. There is a verbal component of short-term memory, which is limited to about seven words, or two seconds of speech. There also seems to be a visual-spatial component which is used to form mental pictures, and an organizational component, which helps us keep our places when performing complex tasks. The various components are largely independent, although when one component approaches capacity, other components are somewhat inhibited (Schacter 1989, 68f-68g).

When short-term memory capacity is exceeded, or when information decays, we find ourselves forgetting information we are using. We forget the phone number before we can dial it, forget how a sentence started, or lose track of what we were doing.

long-term memory

Long-term memory (LTM) is what most people refer to when they talk about memory. It holds information for an hour or a lifetime. It holds facts, experiences, or well-rehearsed processes. It seems to have no "maximum capacity." With disuse, information fades from long-term memory, but information which is occasionally used can persist indefinitely (Hayes 1989, 19, 198).

attention

Attention is another resource which is quite limited in nature. It is not considered to be a storage medium, but rather, a mechanism by which processing is consciously directed. In general, a person can attend to only one thing at a time, although attention can be switched from one task to another very quickly. Furthermore, certain actions can be performed more or less automatically, with little or no attention (Laberge and Samuels 1974, 294-295).

conserving cognitive resources

Because cognitive resources especially short-term memory and attention are limited,
humans have ways of economizing use of resources when frequently-used information and processes are involved. Small pieces of information can be “chunked” together into larger units, and frequently-used processes can become automatized.

chunking

When certain pieces of information often appear together, the mind often defines this group in long-term memory as a single “chunk.” This allows short-term memory to refer to the pieces of information as if they were a single entity, reducing the load on STM. Frequently-cited examples of chunks are certain groups of letters and digits, such as “rma,” “usa,” or “457” (Hayes 1989, 105; Anderson 1990, 113). Other examples may include groups of pen strokes that form certain characters, groups of letters that form frequently-encountered words (Laberge and Samuels 1975, 300–301), or in some people, groupings of chess pieces that form frequently-encountered functional units (Chase and Simon 1973). Akin (1986) has identified certain visual chunks based on patterns of recall exhibited by architects. The significance of these mental representations will be discussed shortly.

automatization

Certain frequently repeated sequences of actions become “automated” or “automatized.” For the remainder of this paper, the term “automatized” will be used, in order to avoid confusion with other possible meanings of the word “automatized.”

When first learning a new physical or mental procedure, the mind often traverses a sort of declarative list of steps. This requires a great deal of attention. However, as the procedure becomes more practiced, it can become automatized to reduce the amount of attention necessary for performance. The procedure may demand attention only occasionally, allowing attention to be focused on concurrent tasks. In the extreme case, the procedure becomes a sort of monolithic entity, which requires only a small amount of attention to set in motion, and once set in motion, generally executes in its entirety. Automatized procedures vary greatly in scope, from recognition of characters, or sequences of motor actions, to complex procedures like driving to work, navigating through a video game, or giving a sequence of commands. Without automatization, even understanding written text would be horrendously difficult, as we would have to consciously study each letter to identify it, and consciously put letters together into words, before being able to consider the meaning of the text.

Automatization has been compared to running a compiled procedure, as opposed to one that is being interpreted (Anderson 1983). Not only does automatization allow attention to be focused on other tasks, it also frees short-term memory, which no longer has to be used to remember as many partial results, to remember steps of interpreted procedures, or to keep track of one’s position within such series of steps. Procedures which are automatized require fewer of our cognitive resources, so for instance, we find it easier to understand the ideas of a technical paper if it is written in our native language as opposed to one with which we know but haven’t practiced extensively.

representations

A representation can be defined as “something that stands for something else—some sort of model of the thing (or things) it represents” (Palmer 1978, 369). Architects use physical and digital representations of proposed designs for the real world. Representations also exist internal to a person’s mind. In order to properly understand the role of representations in design and other mental activities, we need to consider both internal (mental) and external (physical or digital) representations, how they are used, and how they relate to each other.
It is important to note that not all representations are created equal. A poor representation, either internal or external, can actually inhibit mental performance. Inefficient internal representations can waste short-term memory. Inefficient external representations can waste attention and short-term memory by forcing the user to filter out extraneous material, and often lead to inefficient internal representations. When internal and external representations do not correspond well, an additional drain is put on a person’s mental resources. When the representation inherent in a device or tool puts a user in such a situation, the user must perform “translations,” to convert their own representation into the device’s terms. This takes cognitive resources away from other tasks (Day 1988). Psychologists note that while a person can develop expertise at working with a specific bad external representation, performance is still likely to suffer in times of peak cognitive demand (Norman 1988, 29).

use of representations

Part of the reason why experts in a domain perform better than novices is because they develop representations better suited for their tasks. In many cases, their representations combine chunks with automated processes. Studies of experts in various domains (architecture, chess, physics, psychiatry, etc.) have shown that experts in a field develop internal representations based on high-level features and problem-solving strategies (Schön 1983; Chase and Simon 1973; Anderson 1985; Chiu, Felovich, and Glaser 1986). They come to think of situations as “conservation of energy problems,” “cases of measles,” “cases of fibel,” etc. Experts develop mental “libraries” of situations and materials they frequently encounter, and how to deal with them. They develop a “repertoire of expectations, images, and techniques” (Schön 1983, 60).

Studies indicate that these “libraries” are not superficial; they are fundamental to an expert’s ability to function. They are very deeply rooted. Chase and Simon (1973, 35) state that even eye movement—a highly automatic and subconscious activity—is largely guided by these internal representations. A chess expert literally sees the board in terms of combinations of pieces that hold certain functional relationships. Weisberg (1986, 15) estimates that the library of an expert contains 20,000–50,000 such chunks. According to another estimate, it takes 10 or more years of working 70–80 hours a week to develop this library (Hayes 1989, 293–298). Clearly, the internal representations developed by professionals in a field are not to be taken lightly.

Much design studies research converges on a similar idea of libraries of partial solutions. Alexander’s (Alexander, et al., 1977) patterns are an example, as are the elements described by authors like Krier (1988), Thisl-Evensen (1991), Ching (1979), and even Durand (1602) in architectural primers. These authors describe elements like overhead planes, domes, walls, columns, doors, half-inch trim, and so forth. Schön (1988), and Peter Rowe (1987), describe more extensive categories of “enabling prejudices,” “heuristics,” “types,” “typologies,” etc. Akin (1986) identifies similar “visual chunks” based on patterns of recall in architects.

effects of representations

The representations we use allow us to work more efficiently. Limiting drain on cognitive resources like attention and short-term memory is crucial for being able to work on complex problems. Experts in a field are able to do this by using internal representations that utilize chunks and evoke processes for dealing with common situations and materials.

External representations, like drawings, physical models, or computer models can improve a person’s performance by effectively augmenting an architect’s memory and allowing certain ideas to be evaluated more easily. However, in order to be valuable, an external representation must avoid being a drain of resources itself. It must
avoid making users translate concepts from their internal representations into the terms of the external representation.

**Discussion**

**Guidelines for Elements of CAD Representations**

This information about the human mind, representations, and design indicates how representations are used. We know something about the sorts of representations used by architects, in particular. We have considered the question of what's in a representation.

We know something about the effects of representations. We have seen that using the wrong representations can force a person to translate from one representation to another, or keep them from taking full advantage of automated processes. We have addressed the question of why do we care about representations.

So, what does this mean for cad software? Based on the information about the effects of representations, I have compiled the following list of guidelines for cad representations:

**CAD representations should let the architect do much work in a graphical format**

The repertoire of skills architects develop through years or decades of training and practice largely concern working with graphical representations of buildings. We learn how to "read" a plan, how to draft, and so on. We develop "libraries" of graphical chunks that are linked to heuristics for solving design problems. Rather than forcing a user to stop using these automated skills and representations, and try to learn others with nearly the same degree of familiarity, a cad representation should allow architects to use these skills.

This means, for instance, that procedural assemblies, where a user writes a script describing a sequence of operations to execute to create a model, are undesirable except perhaps for a few rare situations. Significant mental translation is required in order for a user to express visual/spatial ideas in terms of a logical series of numerical transformations to be performed on sets of coordinates. The skills and processes involved in writing such a script concern logical conditions, flow of control, ordering of operations, working with numerical representations of form and space, and specification of syntax. The representations and processes used are very different from those generally required to imagine and evaluate spatial or graphical compositions. Software should avoid such situations, where the user is forced to develop new repertoires of skills, perform translations in the midst of complicated work, or try to remember multiple representations simultaneously. So, this end, cad software should let the user work primarily in a graphical format.

**CAD representations should correspond to architectural elements**

While cad representations should be largely graphical, spatial in nature, they should not be based on the sorts of low-level geometric entities found in most commercial cad packages. Describing architectural elements in terms of such entities is often a time- and thought-intensive process of translation.

Instead, for reasons of familiarity and appropriateness for architectural tasks, the elements in a cad system should correspond to architectural elements like those described by Alexander, Ching, Rowe, and others. There is a large body of evidence that architects remember and think about designs in terms of elements like walls, doors, rooms, etc. These embody structural functions, symbolic meaning, and often manufacturing products; consequently, our representations are organized around them. Cad representations should be organized around them, as well.

**CAD representations should have characteristics appropriate for each element**

The correspondence between cad elements and architectural elements should not be mere-
ly cosmetic, can system elements should have characteristics appropriate for the type of architectural element they represent. An ordinary hinged door for instance, should have characteristics describing the size of the door, as well as descriptions of jams, moldings, and placement of hardware.

Representations based on low-level geometric entities are generally incapable of corresponding to architectural elements in an appropriate manner. The characteristics of architectural elements are often hard to express in terms of such elements. Instances of symbols, for instance, may look like architectural elements, but their underlying representations are generally based on geometric entities instead of architectural ones. Consequently, they do not act like architectural entities. A door symbol may look like a door; for instance, but inappropriate performance will result if the user tries to perform a conceptually simple change like modifying the door width (Figure 1).

Some of an architectural element’s characteristics can affect the topology of the element. A colonnade should have characteristics indicating the number and placement of columns. A window needs descriptions of how mullions are arranged. This means that parametric shapes based on simple geometric entities are inadequate for the task. Topological changes, like changing the number of cells in a waffle slab or the number and shape of lights in a window, cannot be described by changing lengths of parts of a prototype shape. An example is shown in Figure 2. The window on the right cannot be described by changing the parameters of the parametric shape on the left. Even if substitution is allowed, so that one parametric shape can be substituted for another, full topological flexibility is not supported. A complete set of parametric shapes, capable of representing every conceivable topology, can not be created. The user needs more than a selection of predefined topologies he needs to have a framework for specifying topologies.

A better solution appears to be to describe an entity with high-level, topology-affecting characteristics appropriate for each particular element. A simple colonnade might have a characteristic to describe the number of columns. A window might have a list of mullions and descriptions of their placement. Appropriate lines, planes, volumes, etc. could then be created for display based on the high-level description.

characteristics should be lasting

An element’s characteristics should not disappear after initial creation of the elements, as is the case with procedural shapes. Many drafting systems use procedural shapes to generate representations of walls, stairs, colonnades, or another architectural entities, asking questions about wall thickness, number of risers, column spacing, or other characteristics of the architectural entity being represented. However, the actual representation created is a set of geometrically-based entities. For instance, a set of planes are created for the staircase. Initial spec-
ification of the representations takes place on
the architect’s terms, but afterwards, the represen-
tations are no easier to modify than lines, 
planes, or other graphic elements created by any
other means. The values the user specified ear-
er can not be modified. The colonnade must
be eliminated and a new one created in its stead.
The user’s thought processes are interrupted
and he is forced to translate a relatively simple
conceptual change into the software’s terms.

representations should be nestable
Certain architectural elements can be thought
of as being components of others. For instance,
arhitectural prisms usually describe doors and
windows as being parts of walls. If a wall is moved
or eliminated, so are any of its openings. Can rep-
resentations should reflect this organization. The
characteristics of a high-level element should
include references to component elements, and
descriptions of how they are incorporated.

representations should be refi nable
It is well established that designs do not
jump fully intact into the architect’s mind. They
are gradually developed and refined, in a more
or less top-down manner. Consequently, a rep-
resentation that corresponds closely to the
architect’s internal representation must be capa-le of undergoing a similar process of refine-
ment. Elements must be able to be added, be
removed, or have values describing their char-
acteristics changed. They should also be able to
develop new characteristics as the designer
makes additional decisions about their form. If
the designer decides that a dome should be
ribbed, or that a wall should have a window, the
representation needs to be able to handle addi-
tional characteristics describing the form and
position of the ribs or window.

This is not to say that design consists entire-
ly of refining earlier elements. Architects sel-
dom start with a single nondescript block and
refine a building out of it. Elements are not only
modified, but also added and removed. The abil-
ity to change or add details to existing design
elements is very important, however.

representations should not require
that the users be pestered for information
One of the guidelines for the selection of
representations states that a representation
should contain exactly the information required
for a task — no more, no less (Norman 1993, 79).
Insufficient information is obviously bad, and
there is no reason to distract attention and fill
memory with extraneous information.

Prompting as user for unnecessary infor-
mation can have additional adverse effects.
Much of the efficiency of the human mind is
due to the automatization of processes. Needless
interruption of such smooth-flowing cognitive
processes is to be avoided. Interruption of
thought processes has been shown to cause
ideas to be lost — in fact, it has been cited as the
primary reason why groups of people generate
fewer ideas than the same number of people
working individually (Deil and Strube 1987).

It can therefore be seen that a computer aid
for design should not interrupt the architect’s
thought processes to demand details the design-
er is not ready to provide. Yet, information suf-
ficient for a designer’s immediate task, like eval-
uating the aesthetic effect of an additional win-
don on the facade of a building, may not be suf-
ficient for other tasks the representation might
be used for, like photorealistic rendering or
structural calculations.

One possible solution to this seeming
incongruity is to use default values or daemons
to generate values, as needed, to supplement
values provided by the architect. Thus, an archi-
tect might only need to indicate the desire to
add a window, and perhaps its dimensions, and
the software might be able to cope with the
missing information by generating its own values
to describe moldings, subdivisions, methods of
operation, and other information missing from
the representation. When exact values for these characteristics become important to the architect, he can replace the automatically-generated values with more accurate ones.

**Development of CAD Software**

Steps in this direction have already been taken. Ironically, some of the best examples can be found in cheap “house modeling” software marketed to help non-architects design their “dream house.” Figure 3 shows a screen from one such program, 3D Home Architect, running on an not personal computer. As can be seen, the representation of a window goes beyond lines, planes, or volumes. It is not merely an instance selected from a pre-defined list of symbols. Rather, it is a higher-level entity with attributes appropriate for a window, and produces an a graphic representation based on this description. It contains descriptions of method of operation; number, placement, and size of Mullions; and size of moldings. It is nested within a wall entity. It contains little lighting information, and limits the user’s options with inherent assumptions about construction methods and materials, but it is a step towards the elements described in this paper.

**Current Work**

A project to implement a more extensive system is currently underway. The goal of this project is to allow the editing of computer models by letting the user manipulate representa-

---

Figure 3. Definition of a window in 3D Home Architect, Edition 2.0 Demo version. Model is a sample provided with the demo version, modified by the author. 3D Home Architect is Copyright 1996 by Brenderhouse Software, Inc., with program copyright by Advanced Visual Technology.
<table>
<thead>
<tr>
<th>description type</th>
<th>description method</th>
<th>example element</th>
<th>example element</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard</td>
<td>specification of parameters</td>
<td>round column</td>
<td>radius, height</td>
</tr>
<tr>
<td>process-oriented</td>
<td>parameters controlling the generation of the element according to a given generation process</td>
<td>solid of revolution</td>
<td>diameter of cross-section to revolve, angle of revolution</td>
</tr>
<tr>
<td>bottom-up</td>
<td>operations performed on non-architectural graphic primitives</td>
<td>unique sculpture</td>
<td>CSG description: set operations performed on primitive volumes</td>
</tr>
</tbody>
</table>

Table 1. Element description types to be used in current work.

...tions of the sort described in this paper. In order to accommodate a full range of elements, three types of element descriptions are being used: standard, process-oriented, and bottom-up.

These are shown in Table 1 and described below.

Standard element descriptions are to be used in situations where an element follows a certain "standard" prototype, like a round column, a Vignola Corinthian column, a standard plumbing fixture or piece of furniture, etc. The element can be described by specifying a few parameters for location, orientation, scale, radius, etc.

Process-oriented descriptions are to be used to describe elements according to a process for generating them. Examples include a column described as a series of cross-sections, a solid of revolution, an element created by extruding a cross section along a specified extrusion path, etc. Changing the characteristics of a process-oriented description can result in a change in the topology of an element.

Bottom-up descriptions are to supplement the other description methods. No matter how desperately a programmer tries to anticipate every possible form an architect might wish to create, some architect somewhere will want to create something so bizarre as to not fit any provided standard or process-oriented descriptions. A bottom-up description method, based on Constructive Solid Geometries will be provided for such contingencies.

These three types of element descriptions are not incompatible; they can work together.

For instance, a standard description of a window might include references to component nullions, defined in a process-oriented manner as a cross-section to be extruded along a path. What starts out as a simple round column in early schematic design may be represented initially with a standard element description, but as the design is refined, it may be changed to a column defined as a series of cross-sections or as a unique form sculpted through CSG operations.

**Conclusion**

The mental representations architects use are vital for being able to function as productive designers. A variety of authors have identified elements like walls, columns, doors, rooms, and roofs as examples of the sorts of design elements around which an architect’s knowledge and skills are based. These representations are not merely familiar; they allow efficient use of limited cognitive resources like attention and short-term memory. They are closely linked to the thought processes an architect uses in the process of design, including solution strategies, and likely even perception.

In general, guidelines for user interfaces and for selection of external representations advocate matching internal representations. This avoids forcing the user to perform translations, and also avoids distracting the user with unwanted information. In keeping with this principle, several guidelines emerge concerning representations in an architectural CAD system:

—Representations should let the user do...
much work in a graphical format

--Representations should correspond to architectural elements
--Representations should have characteristics appropriate for the architectural elements they represent
--Characteristics should be lasting
--Representations should be gradually refinable
--Representations should not require that the users be fastened for information

A project is currently underway to demonstrate the feasibility of such elements. Through a combination of standard and process-oriented description methods, most architectural elements should be able to be represented, and a bottom-up description method can be used to represent any other, individual elements. The detail to which such a system of elements can reasonably be carried, the ability to base various analyses and calculations on such representations, and several other issues still remain to be explored, but there is sound psychological evidence supporting the need for such representations.

references


