

Artificial Intelligence for Engineering Design, Analysis and Manufacturing

<http://journals.cambridge.org/AIE>

Additional services for **Artificial Intelligence for Engineering Design, Analysis and Manufacturing**:

Email alerts: [Click here](#)

Subscriptions: [Click here](#)

Commercial reprints: [Click here](#)

Terms of use : [Click here](#)



Sketch-based interfaces for modeling and users' needs: Redefining connections

Catherine Elsen, Jean-Noël Demaret, Maria C. Yang and Pierre Leclercq

Artificial Intelligence for Engineering Design, Analysis and Manufacturing / Volume 26 / Special Issue 03 / August 2012, pp 281 - 301
DOI: 10.1017/S0890060412000157, Published online: 14 August 2012

Link to this article: http://journals.cambridge.org/abstract_S0890060412000157

How to cite this article:

Catherine Elsen, Jean-Noël Demaret, Maria C. Yang and Pierre Leclercq (2012). Sketch-based interfaces for modeling and users' needs: Redefining connections. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 26, pp 281-301 doi:10.1017/S0890060412000157

Request Permissions : [Click here](#)

Sketch-based interfaces for modeling and users' needs: Redefining connections

CATHERINE ELSÉN,^{1,2} JEAN-NOËL DEMARET,¹ MARIA C. YANG,² AND PIERRE LECLERCQ¹

¹LUCID, University of Liège, Liège, Belgium

²Department of Mechanical Engineering and Engineering Systems Division, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

(RECEIVED June 30, 2011; ACCEPTED February 13, 2012)

Abstract

The goal of this paper is to reexamine assumptions about sketch-based interfaces for modeling in the context of designers' needs and practices. Research questions examine (a) the type of sketch support and (b) the timing of support. Both concepts try to determine when, what, why and how to augment design processes in a way that is useful to designers. Two experiments (one in architecture and one in product design) based on ergonomics theory are conducted and intend to question some of these assumptions. The Port Zeeland experiment examines how 20 novices perceive and copy a blurred architectural sketch, which provides clues for a sketch interpretation system. The "Tragere" experiment studies how 12 professional product designers, some of whom are "idea generators" and others "idea pursuers," perceive, recognize, and handle a design sketch. The results take a designer's point of view in assessing the timing and value of sketch assistance in product design. The quantitative data analysis provides rich clues about when, why and how product sketches should be supported. The paper explores the strategies developed by designers to perceive and recognize graphical content and discusses the generation of three-dimensional volumes, the univocity state between sketches and three-dimensional models, and the treatment of features in freehand sketches. The paper concludes with observations on the timing and value of support, as first integrated in NEMO, a tool for early stage architectural design, and then in PEPS³, an early stage framework for product design.

Keywords: Architecture; Design Processes; Product Design; Sketch-Based Interface for Modeling; Users' Needs

1. INTRODUCTION

Two practice-based observations traditionally structure research on early-stage design tools. The first is that freehand sketches remain the most natural and efficient way to launch new ideas (think of a sketch on the back of a napkin), but are less and less suited for reaching time to market goals that increasingly drive the design and development process. The second observation is that computer-aided design (CAD) tools, as powerful as they are for the later stages of design, are still poorly adapted to preserving the ambiguity inherent in the preliminary phases of the design processes. As summarized in Section 2 of this paper, for the last 30 years researchers in engineering and product design, computer graphics, psychology, and user experience (UX) have generated in-depth theories, prototype tools, and methods to address these issues.

In the domain of sketching design tools, a large community of researchers active in sketch-based interface for modeling

(SBIM), computer graphics, and nonphotorealistic rendering have investigated ways to overcome the limitations of CAD software as a preliminary design support tool by merging computational efficiency with freehand sketching capabilities. In doing so, tool developers have made assumptions about sketching behavior, such as the timing of strokes' beautification, or the value of automatic generation of three-dimensional (3-D) models' generation. These assumptions, even if they are often intuitively accurate, are not always grounded by analysis of designers' observed processes and needs.

In contrast, communities of psychologists, ergonomists and UX theorists have proposed models, design methods and guidelines that are based on observations of the real behaviors of designers, but these have only slowly gained adoption in everyday work practice, in part because such models may sometimes be too general or too difficult to realize from the point of view of software development.

If communities of psychologists, design theorists and software engineers individually face challenges gaining adoption of their respective approaches, why not consider a strategy that integrates these multiple points of view? This paper brings to-

Reprint requests to: Elsen Catherine, LUCID, University of Liège, Chemin des Chevreuils 1, Bat. B52/3, Liège 4000, Belgium. E-mail: catherine.elsen@ulg.ac.be

gether civil and architectural engineers, software engineers, mechanical engineers, and cognitive ergonomists to formulate an approach that considers the following:

- methods and models drawn from cognitive psychology to address user needs specifically in early stage design;
- computational approaches to augment early stage tools for design;
- different modes of graphical man–machine interactions as an alternative to traditional input devices;

This work addresses specific research questions (below) concerning strategies designers adopt to capture and create representations, the features that tools should include to support the interpretation of these representations, and the ways that the interpretation of a representation can be adapted to specific fields of design.

The goal is not to suggest a universal model or method that connects computer graphics, design engineering or psychology researchers. In fact, interactions between these areas are complex and context driven and attempting to solve them globally would lead to an abstract and unproductive meta-model. Instead, the aim is to understand designers' practices and how to better formulate SBIM tools with clear and specific recommendations for architecture and industrial design.

This paper centers around two key aspects of the use of design tools: (a) the *type* of computational assistance that is provided to designers as they engage in design activity and (b) the *timing* of that assistance. These notions are phrased as research questions:

Are certain “types” of interpretation better adapted to the design fields we are examining? How should interpretation be adapted to different design fields?

Sketches may be interpreted in a myriad of ways by software. Architectural design, as we will see in next section, typically uses two-dimensional (2-D) and symbolic representations, and are generally handled using a semantic approach for interpretation. Should interpretation systems consider other strategies, such as exploiting the timing of strokes (chronological approach) or the areas of sketches (zoning approach)? And are such approaches appropriate for fields such as industrial design, where fewer prototypes tools have been developed?

What elements of a representation should be considered effective as input data for SBIM tools for preliminary design?

Design representations can be highly ambiguous and difficult to interpret. To limit the combinatorial explosion of possible interpretations, software engineers have developed systems that quickly focus on specific types of input data, such as beautified strokes. Are the types of input data used in current systems in fact the best ones to focus on? What are some of the strategies adopted by designers during the act of perception and recognition? How can these strategies (and their linked input data) be effectively used by software engineers?

What is the appropriate timing of sketch computational assistance in design tools?

Developers make assumptions about the timing of strokes' treatments, such as beautification, the real-time and automatic generation of 3-D models, or about the general univocity existing between sketches and 3-D models. Are these assumptions correct and do they reflect realistic designer behavior?

This paper presents two different experiments to address the research questions, one focused on architecture and the other on product design. The first experiment explores various sketching layout strategies that designers use. This involves an experiment in which 20 subjects reconstruct a 2-D architectural drawing. The analysis of human perception and interpretation processes reveals clues for further computational interpretation. The results are suggestions for how a sketch interpretation system can seamlessly capture the information necessary to provide appropriate, perfectly timed assistance for preliminary architectural design.

The second experiment involves observations of how professional industrial designers generate and perceive freehand sketches. Results illustrate the predominance of perspectives and the importance of shifts from 2-D to 3-D representations. Learning about how these shifts concur to the concept's evolution help us assess the timing and value of assistance in preliminary product design. Appropriation and perception mechanisms in between designers enable us to understand which key features constitute the graphic essence of the representation. These quantitative results provide good clues about when, why, and how design should be supported.

2. RELATED WORK

This paper is built on two assumptions about the relationship between sketching and 3-D modeling in early stage design. These have been empirically established and extensively discussed in Elsen et al. (2010):

- **Reduced emphasis on sketching:** For designers, freehand sketching remains a crucial tool for preliminary design (Garner, 2000; Tversky, 2002; Basa & Senyapili, 2005; Jonson, 2005) but the time allocated to it during the design and development cycle constantly decreases (Jonson, 2005);
- **Increased emphasis on CAD:** As designers sketch less, CAD tools are slowly relied upon to support more of preliminary design. Even if these tools are paradoxically supposed to be everything but suited to assist ideation, designers divert some of their functionalities to do so (through the use of what we called “rough 3-D models”).

The recurrent dichotomies that appear in the literature between sketching and CAD (including tools, processes or other support for individual or collaborative ideation) as well as between “designers that sketch” and “designers that CAD” therefore become more and more outdated. In practice, designers

exploit both tools as needed, and are less concerned with when the “right” phase in the design process to use them. The next sections will show how these dichotomies still appear in the SBIM literature, including the approaches that researchers and engineers have taken and how they impact the formulation and development of tools.

2.1. Sketching and CAD in architecture and product design

Design tools can be considered on several levels of abstraction. The term *sketch* can refer to the physical tool (including its components the paper and the pen) but it can also refer to a process, an intermediary design goal (the designer ideates through the process of sketching) or to an externalized image, documenting the product evolution (the sketch understood as a drawing). Identical polysemy occurs concerning “CAD” artifacts and can be explained, according to Darses (2004), by the coexistence of various abstraction levels among the subject's understanding process.

Researchers have focused on cognitive aspects of using design tools and usually contrast traditional tools (i.e., sketching, physical modeling) with new-generation tools (i.e., CAD tools, rapid prototyping) at the earliest, conceptual phases of the design process (Yang, 2009). Sketching is known as a fast, intuitive technique to represent the opportunistic flow of ideas (Visser, 2006). Sketches reduce cognitive load and provide mnemonic help (Suwa et al., 1998; Bilda & Gero, 2005); they enable an efficient and broad problem/solution exploration with minimal content (Ullman et al., 1989; Cross, 2000) and spur unexpected discoveries by keeping the exploration dynamic (see–transform–see process; Schon & Wiggins, 1992). They also enable ambiguous, highly personal content (Leclercq, 2005) that impact their

adaptability to serve all kinds of communicative purposes (McGown et al., 1998; Détienne et al., 2004). The content of sketches can be implicit, have limited structure (making them difficult to interpret), and their rigid and static aspects make them “old-fashioned” compared to more reactive representations (Leclercq, 2005).

Sketches can also be analyzed in regard to their applications or contents. Several “types” of drawings are recognized: thinking sketch (Tovey & Richards, 2004), communicative or talking sketch (Ferguson, 1992), and reminder sketch (Schenk, 1991). Do and Gross (1997) and Lim (2003) define various taxonomies for sketches, whereas Do (1995) and Dessy (2002) try to determine underlying principles for sketching. At a more detailed level McGown et al. (1998) and Rodgers et al. (2000) are interested in the graphical complexity of traces.

Researchers also point out the specificities of certain representations, like architectural sketches or diagrams for instance. These, mainly 2-D symbolic sketches, enable a semantic computational interpretation (Fig. 1). Leclercq, analyzing several architectural representations in the context of their implementation, proved that more than 80% of the sketches really useful for ideation are 2-D (1994). In contrast, perspectives are used during later stages (once the idea has been developed), mainly for communication and negotiation purposes. In product design, by contrast, more importance is assigned to 3-D representations whereas too little empirical data has been gathered to evaluate the significance of symbolic codes.

In contrast, CAD tools are highly valued for their computational optimization and simulation abilities; they enable relatively quick access to 3-D visualization and ease modifications through parameterization, nurturing a certain type of “heuristic fecundity” (Lebahar, 2007); they ease technical

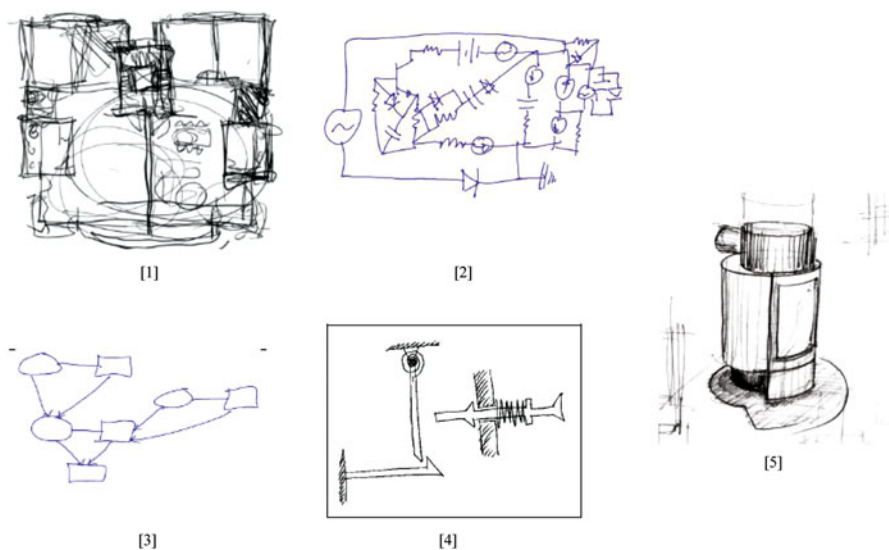


Fig. 1. Symbolic contents in architectural sketches [1] and electric diagrams [2] (Alvarado, 2004); graphical codes in diagrams [3] and in cutout scheme [4] (Davis, 2002), in regard to an axonometric representation in product design [5]. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

data exchange through the unification of formats and sometimes CAD modeling leads to positive premature fixation (Robertson & Radcliffe, 2009). This latter is considered a negative when a “depth” strategy of ideation contributes to the production of fewer alternatives (Ullman et al., 1989). From a user point of view, traditional windows, icons, menus, and pointing device (WIMP) interfaces introduce a level of cognitive overhead and can substantially divert users from their essential designing activities.

These views of the advantages and limitations of sketches and CAD tools in supporting ideation generally force a stand in favor of one or the other design tool. Previous research recommends another approach: to analyze design activity as a whole process that leverages both tools’ complementary features (Elsen et al., 2010).

2.2. SBIM

In the SBIM literature, two prominent research approaches are featured:

- some SBIM prototypes explore new types of interactions for the modeling of 3-D objects inside a 3-D world, and thus serve designers who will make more extensive use of these ways of expression;
- in contrast, other types of SBIM prototypes suggest new modes of freehand drawing with different levels of interactions: simple trace capture (with graphic treatments like beautification); reconstruction of geometries based on various rules or reconstruction of objects based on (sometimes semantic) interpretation of traces. These prototypes address the needs of designers who are supposed to prefer “pen and paper” style interaction.

The next two sections will examine these two approaches and will underline some of their assumptions.

2.2.1. Interactions for 3-D modeling

Whatever the chosen input device (mouse, pen, or haptic; for the latter see Kanai, 2005), the prototype software described here all aim to ease the creation and manipulation of 3-D primitives in order to achieve more complex geometries.

Danesi et al. (1999) suggests three subclassifications for SBIM prototype software:

- software that employs a WIMP interaction (mainly menus and mouse);
- software that recognizes a limited range of gestures for forms selection, generation, and modification (see IDEs, Branco et al., 1994; Sketch, Zeleznik et al., 1996; or 3DSketch, Han & Medioni, 1997; all referenced in Danesi et al., 1999)
- software that exploits surfaces and deformations (like nurbs, volumes of revolution, extrusions). Ides proposed several modes of interaction that can be classified here, as well as 3D Palette (Billinghurst et al., 1997), 3D

Shape Deformation (Murakami & Nakajima, 1994), Virtual Clay (Kameyama, 1997), or 3-Draw (Sachs et al., 1991; all referenced in Danesi et al., 1999).

Interfaces for Solid Sketch and Digital sculpting can also be listed here: they usually enable users to project some virtual material perpendicularly to a reference plane, creating rough volumes that can be reshaped and modified in a second phase (e.g., Z-brush®). We also include approaches that automatically generate complex forms (parametric, genetic, or evolutionary, see Kolarevic, 2000), even if these rely on computational approaches rather than designer intervention during design iteration.

The DDDOOLZ sketching system (through mouse interaction in an immersive 3-D environment called “virtual reality,” Achten et al., 2000; and Quicksketch, which cleans the 2-D traces and builds mainly extruded 3-D models in constant interaction with the user, Egli et al., 1995) finally constitute the transition to SBIM prototypes that focus principally on the “paper and pen” metaphor. If they operate “the line” (through mouse or pen) as input information for sequential and interactive building of the 3-D models, they do not involve the use of geometric reconstruction, let alone some interpretation mechanisms presented in the next section.

2.2.2. Paper–pen metaphors

The development of pen-based interfaces has been closely linked with the development of SBIM prototypes supporting preliminary design processes through a paper–pen metaphor, starting with the seminal work of Sutherland on SketchPad (Sutherland, 1963).

In a survey paper, Olsen et al. (2009) compare over 150 interfaces of such type and summarize the three main steps in creating a SBIM prototype. The first and most crucial step is the generation of a digital model from sketch lines. This can be done in various ways, requiring more or less intense interaction with the user, or by performing a more or less autonomous interpretation of traces. This stage generally includes a phase of filtering the graphic information (through fitting or intentional oversketching), called “beautification.” This beautification step enables the transformation of multiple, redundant, multitraced sketch lines into a unique and accurate trace. In the widespread case of automatic fitting, this usually appears at the same time as the apparition of the trace, so that the user sees his/her strokes beautified as soon as he/she has drawn them. After beautification, reconstruction or interpretation approaches are used to generate a 3-D representation of the project.

The second step consists in deforming the basic model in order to reach, in the most “faithful” possible way, the desired geometry. Once the model is generated (with parametric or meshed surfaces), the user can apply a set of operations (cut, fold, hole, freely deform, Booleans operations, and so on) that are relatively easily supported by the computer, the preexisting 3-D model anchoring the changes. Two difficulties nevertheless remain. The first is the pen. Pens are particu-

larly well suited to the input of the trace, but are not optimal for the modification stage. It is sometimes complex to move in a 3-D virtual space with a pen, and pens do not provide the control necessary to deform accurately. The second is linked to the general univocity of the metamodel linking the sketch and model: once the 3-D model is generated, the modifications imposed on the form will not be translated any longer to the sketch. One might question if this technological break between the conceptual sketch and the editable 3-D model really fits the cognitive and internal processes of the user.

The third and last step enables users to add details to the volumes, like annotations, surface features, and profile features (Aoyama et al., 2007).

This paper mainly concentrates on the first step, that is the creation of the 3-D model based on sketch lines, and its three potential stages: (a) the capture, filtering and spatial positioning of traces, (b) the geometric reconstruction of volumes, and/or (c) the (semantic) interpretation of a sketch's contents.

The capture, treatment, and spatial positioning of traces are supported by several techniques that are summed up in Juchmes (2005). These techniques, including the data filtering and beautification, are the first and almost systematic step of any SBIM. Some software equip the user with "simple" support in the process of drawing. This can be done in various ways: by using tracing guides (that can be volumetric, see, for instance, SketchCad from Kara et al., 2007), through instant corrections, or automatic fitting to basic geometric primitives. A good example of such a system is "I Love Sketch" (Bae et al., 2008), which involves gesture recognition and drawing in a 3-D dynamic world (a technique also called "3-D sketch"), exploiting the epipolar method when more complex curves have to be created. This epipolar method has proven to be cognitively challenging for designers. Another limitation of this prototype stands in the type of input: the 3-D model is non-volumetric in essence (because of its wired structure) and the graphical input in a 3-D world requires strong drawing and 3-D visualization expertise. Using volume perception, further modifications or implementations are difficult, even sometimes impossible.

A question arises here concerning the timing of this first step of assistance: it has always been assumed that the capturing, filtering, and spatial repositioning of strokes should be made immediately, in real time. Could this as-available assistance negatively impact the overall design process? What are the real needs of professional designers regarding this question?

The second stage, that is, the geometric reconstruction of the model, goes a step further in 3-D generation by associating graphical units with some "basic" geometric and spatial information. The computer, for instance, can automatically extract "regions" from the drawing (closed geometrical shapes or blobs; Saund & Moran, 1994; Saund, 2003) by using predefined rules, topological relationships or Gestalt perceptive standards in order to spatially position traces in the 3-D world (Wuersch & Egenhofer, 2008). All these topological, geometrical, and spatial links correspond to complex algorithms,

which are summarized in Company et al. (2004). These so-called "constructive" methods can be semisynchronous and exploit image recognition techniques (like Sketch-VRML; Jozzen et al., 1999), or require the user to draw following the epipolar method (Karpenko et al., 2004; Tian et al., 2009).

Another complementary approach is called "free-form." Features are here captured and recognized as closed contours and are transformed into blobs by software. The best-known example is Teddy (Igarashi et al., 2007): for each recognized contour, this program provides a rough "2-D skeleton" (a sort of neutral axis network) that becomes the structure for the revolution volume. Other prototype tools assume the same principle and add the ability to constrain the volume by hidden edges (reconstruction by T-junctions, PerSketch; Saund & Moran, 1994).

Finally, another group of constructive systems exploits parallel projections or perspective rules to manage the 3-D reconstruction (Lipson & Shpitalni, 1996; Huot, 2005; Lipson & Shpitalni, 2007). Relatively robust for mechanical or architectural parallelepiped objects, these systems first identify the geometric patterns (parallelism, symmetry, angles, isometrics, ...) and associate a "geometrical meaning" with the lines (a line being an edge, apparent or hidden, a contour, and so on). These systems can sometimes be limiting to use: they require that designers express their ideas in correct projection and with a point of view such that no edge is hidden by another. Their main advantage is the ability to quickly infer a coherent 3-D volume, since Lipson and Shpitalni (2007) work on closing "skins" over their wired structure.

Capture, recognition and reconstruction can eventually go a step further with the association of pre-defined meaning to specific content, named the "semantic approach." Dessy (2002) defines three essential key factors for such an interpretation: an intense presence of geometric primitives, the constant repetition of these primitives' properties and some constancy in the repetition of their relationships (juxtaposition, contact, inclusion, interpenetration, etc.). The recognition of these basic geometric forms triggers a process of identification governed by rules that guarantee the uniqueness of the symbol and ignore unnecessary forms. Once the symbol is recognized, the next step is to associate some common sense to the unit and then, if necessary, a set of properties.

Few design domains present these features and symbols in sufficient quantity to allow the development of such prototypes. Many developed tools focus on simple hand-drawn diagrams. Some research has been done on electrical diagrams (Gennari et al., 2004), UML diagrams (Casella, Deufemia, Mascardi, Costagliola, et al., 2008) and sketched user interfaces (Plimmer & Freeman, 2007). In mechanical engineering, one of the most robust system is ASSIST (Alvarado & Davis, 2001), referenced in Davis (2002), that provides real time simulation of objects' kinematics. Another prototype tool, called EsQUISE, interprets architectural sketches in real time (Leclercq, 1994). By capturing and recognizing geometries (see Fig. 2), types of lines (walls or windows), universal architectural symbols and annotations, the system

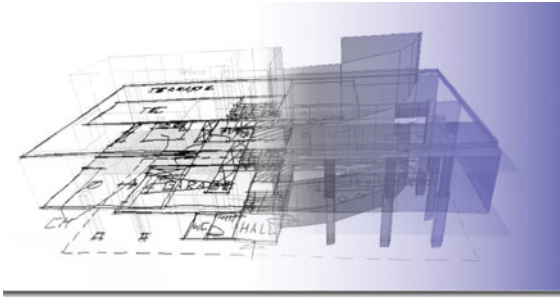


Fig. 2. Screenshots of EsQUISE interpreting architectural sketches into a three-dimensional volume. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

offers designers not only a self-generated 3-D model of the building being designed (through extrusion), but also evaluators (thermal, topological). Another example is VR Sketchpad (Do, 2001) and more recently the work of (Casella, Deufemia, Mascardi, Martelli, et al., 2008) on architectural diagrams.

This semantic approach still encounters three obstacles, limiting its efficiency:

- First, it is still difficult to model more complex 3-D shapes.
- Second, constraints must be applied to the input sketch in order limit the combinatorial explosion of possible interpretations. For instance, Macé and Anquetil (2009) force the user to finish the drawing of one symbol before drawing another one. This restricts the designer's freedom.
- Third, these prototype tools can only work with target domains presenting high symbolic and semantic content.

This related work shows us how varied approaches for reconstruction and interpretation can be. Each software prototype opts for a different strategy to generate the 3-D model. Computational efficiency is usually the main argument for choosing one instead of the other, but we wonder if each strategy is equally respectful of designers' needs and practices.

All of these systems assume that the 3-D model is needed as soon as possible, and as automatically as possible. Again, we want to explore professional designers' expectations considering this assumption.

2.3. Recommendations from psychology and design ergonomics

In parallel, psychologists, ergonomists and UX theorists suggest models, methods and guidelines to optimize various aspects of design ideation. These suggestions can address team performance and organization, task management and sharing or use of tools. Thanks to dedicated methodologies, these researchers provide in-depth analysis of subjects' needs, beliefs and expectations and reveal the "silent realities" or unspoken aspects of their tasks (Nijs et al., 2010).

In the domain of preliminary design, this research covers a wide range of topics, from end-users' needs to the processes that designers use to recommendations for software engineers who develop the design interface. Many suggestions concerning SBIM (or more widely man-machine interactions) can be found in literature (Bastien & Scapin, 1995) and we selectively list some of the guidelines for sketching interfaces. These should

- be transparent, adaptable, and intuitive (Safin et al., 2005); interoperable, "plastic" (Thévenin, 1999, quoted by Demeure, 2007), and perfectly suited to the target end-users (in this case, designers);
- be able to support imprecise information (Darses et al., 2001);
- allow flow between various representations, content, and levels of abstraction (Darses et al., 2001);
- provide upstream feedback, error detection, and evaluation; and
- enable (or even support) discovery, comparison of variants and reinterpretation.

These specifications, drawn from in-depth understanding of complex mechanisms and dynamics, bridge the distance between a basic description of the task and prescription (Dorst, 2008). They equip design engineering with a "bottom-up" approach that should nurture the process of designing of new interfaces and tools to support to ideation.

There remains a gap between these specifications and the prototypes that are created by SBIM software engineers. This could be linked to the very broad nature of these recommendations, while computer engineers must think about very specific questions in software development. This leads to misunderstandings and sometimes hazardous interpretations. Our hope is that psychology and UX researchers will be able to see their recommendations into development of usable software, perhaps through collaboration with software and SBIM researchers.

3. METHODS

The previous research questions are considered through two different experiments: the first one examines freehand sketches in architecture, the other product design sketches. Considering both architectural and product design domains together enables us to highlight the differences between design processes and tool usage and, more importantly, to underline how important it is to define context-specific recommendations for dedicated design support tools.

For both experiments, two assumptions are made (already established in architecture by Leclercq, 1994):

- all of the information needed to enable adaptive assistance of sketching (adapted in content, in intent, and in timing) are already present in designers' sketches and work practices and



Fig. 3. Sketches' perception and retranscription. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

- analysis of human (experts or novices) perception and interpretation of blurred sketches can reveal clues for further computational interpretation.

The first exploratory experiment, named the “Port Zeeland experiment,” is largely built upon this latter assumption. The goal is to observe the elements that designers focus on when formulating sketches. Twenty novices (5 students in architectural design, 12 mechanical engineering students, 2 software engineers, and 1 cognitive psychologist) are shown a blurred, incomplete and preliminary architectural sketch and are asked to copy it, verbalizing their thoughts following the “think aloud” protocol (Fig. 3). A neutral, exterior observer restarts the think aloud process when necessary and takes active notes about how the subject reconstructs the sketch. The whole process is video recorded for further analysis.

Each task is completed in about 20 min and is followed by a short debriefing, built upon a semidirective interview technique. The tapes are then iteratively and qualitatively analyzed

and segmented in successive clips corresponding to distinct phases that describe how the participant handles the questioning of graphical units, understanding graphical units, or recopying them. This segmentation is defined with the help of an expert, familiar with architectural representations and able to track shifts between units presenting different architectural, conceptual or functional meanings.

The analysis of those segments enables us to understand which clues the subjects use to capture the sketch and what kind of strategy is used to recopy it. If semantic interpretation has proved itself an adaptive strategy for highly symbolic content such as in architectural representations, we are interested complementary strategies to reduce the obstacles to computational efficiency. By showing participants a static rough sketch, we can evaluate how difficult it is for people with limited architectural knowledge to capture and understand an architectural representation. Are they distracted by the “off-line” character of the representation? Moreover, we are able to assess if architectural symbols, core to semantic interpretation systems, can be easily understood when blurred and roughly drawn.

The second experiment, named “Tragere,” pursues comparable goals but with a different methodology. It again explores how designs are reconstituted, but in this case examines how they can be incrementally modified, rather than duplicated. This time, we form two groups of professional product designers, experts in consumer design, furniture design, or naval design. Each designer from the first group is asked to tackle a short design problem and to sketch on a Wacom Cintiq® Graphic tablet running a dedicated sketching application (Tragere prototype, see Jeunejean, 2004; Fig. 4). Each of the

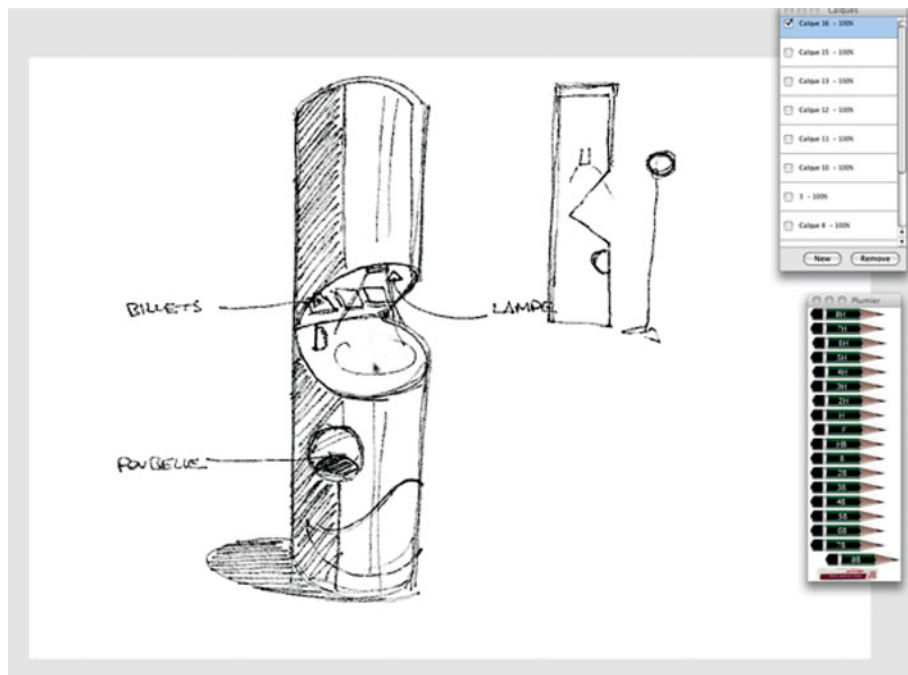


Fig. 4. The Tragere interface and its “paper-pen” rendering. Here, a piece of public furniture design (designer 7). [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

12 participants is presented with one of three design prompts close to the subjects' respective fields of expertise: one prompt relates to the design of a cafeteria tray for children, the second one to a piece of public furniture, the last to a yacht. The sketching interface enables the creation of several transparent layers that can be superimposed.

Once all "group 1" designers have achieved their design task (in about 45 min each), three of the most clear and complete projects are selected to serve as the prompt for the second group of designers. We show each designer in this second group one of the three previously (anonymously) sketched projects, according to his/her respective domain of expertise (product, furniture or naval equipment, Table 1). Each receives a similar design prompt to the one shown to the group 1 designers, except that this time designers are asked to take over the launched project (using the same tablet) as if the first colleague was suddenly no longer on the project, leaving no information other than the sketch. We also ask them to "think aloud" during their "capture–interpretation–appropriation" process, in order to gain data about how they perceive the sketch, which key features help them to understand the group 1 designer's intention, and how they intend to keep the project going. Some semidirective questions are asked as a debrief of the task.

Seven designers thus assumed the role of "idea generator" and five others the role of "idea pursuer," all 12 suggesting preliminary design solutions. Each generative task was preceded by a short exercise in order to help the designers familiarize themselves with the intuitive and easy to use Tragere interface. Each session was video recorded, and dynamic screenshot capture enabled further trace-to-trace qualitative analysis. The data collected is then again segmented in short clips and coded, with more detail than for the "Port-Zeeland" experiment (see the 12 variables and their values in Table 2).

Table 1. Description of the experimental plan

Experiment No./ Designer No.	Design Prompt	Task
1	Tray	Generator, group 1
2	Tray	Generator, group 1
3	Tray	Generator, group 1
4	Tray	Group 2, builds on the work of designer 3
5	Tray	Group 2, builds on the work of designer 3
6	Public furniture	Generator, group 1
7	Public furniture	Generator, group 1
8	Public furniture	Group 2, builds on the work of designer 6
9	Public furniture	Group 2, builds on the work of designer 6
10	Yacht	Generator, group 1
11	Yacht	Generator, group 1
12	Yacht	Group 2, builds on the work of designer 10

Cross analysis of concurrent occurrences enables a quantitative approach of the data.

The type of externalization simply refers to the type of drawing produced: is it a perspective, or an elevation? Is it only annotation, perhaps added to the previous drawing? The "aim" variables are the main objectives a designer can follow during preliminary design. Defined with the help of a professional designer, the values for this variable range from "design" to "modify" or "ask a question." Then, we observed the various shifts occurring between 2-D representations (i.e., elevations or sections) and 3-D representations (i.e., perspectives) and tried to understand what caused these shifts. After an iterative analysis of the data, we reached seven main causes for these shifts, going from "explain, synthesize or synchronize" to "introduce the preexisting environment." The dimension of the internal thoughts, revealed by visual, gestural and verbalized clues of the mental activity, is then coded.

Going deeper into the fine-grained detail of the strokes' analysis, we code the type of trace and its chronological appearance. Different levels of strokes are marked, some of them appearing in specific cycles over time. Cycles of blurred, crystallized strokes appear, and sometimes repeated strokes are added to generate what we call "blurred–repeated–crystallized" (B-R-C) cycles of strokes.

Goel's lateral and vertical transformations have been coded as well, as a way to track the project's evolution during time (Goel, 1995). Lateral transformations occur when the subject goes from one concept to a different one, whereas vertical transformations delve more deeply on the same concept.

The "type of curve" refers to "principal" and "secondary" curves. Principal curves persist throughout the design process: they can still be seen in the final representation. Secondary curves, in contrast, disappear from the drawings and do not strategically structure them.

The "scope" and "exhaustiveness" variables examine the level of detail and the level of completeness reached by a specific drawing (global or detail? completely drawn or with zones that are unfinished?). The "type of reinterpretation," eventually, considers to which extent the designers of the second group (the "idea pursuers") capture the graphic content of the sketches they receive.

As SBIM do not yet fully support the preliminary phases of product design, our hope is that this mechanism of "generating–capturing–perceiving–interpreting" product design sketches will provide important clues about the type and timing of assistance needed on an everyday basis.

4. RESULTS AND DISCUSSION

4.1. Port Zeeland experiments' results

Qualitative analysis of the videos and debriefs of the Port Zeeland experiments provide interesting results about sketches' perception and key features. These results can help software engineers enhance or adapt their SBIM prototypes for preliminary architectural design. To begin with, we immediately

Table 2. Variables and values for data coding scheme

Type of Externalization	Aim	Shifts	Cause of Shift	Dimension of Thought	Type of Trace
Annotation	Question	2-D > 3-D	Negotiate, insist	Programmatic	Axis
Section	Reminder	3-D > 2-D	Explain, synthesize, synchronize	2-D	Alpha-num
Elevation	Modification		Preexisting environment	3-D	Crystallized
Tech. background	Iteration		Modify		Blurred
Perspective	Crystallization		Simulate, evaluate		Blurred > cryst
Scheme	Communication		Generate		Repeated
Tag	Design		Clarify, detail		B-R-C
					Light
					Shadows, textures
					Any form
					Geom. primitives
					Symbols
Transformation	Type of Curve	Scope	Exhaustiveness	Reinterpretation	
Lateral	Principal	Component	Complete	Total	
Vertical	Secondary	Global	Incomplete	Partial	
				None	
				Evaluation	

Note: 2-D, 3-D, two-dimensional, three-dimensional; B-R-C, blurred–repeated–crystallized.

observed that to manage the blurred architectural representation the subjects adopted three different strategies.

The first strategy, which we called the “structural engineer” strategy, consists of a heliocentric approach: subjects start with a global analysis of the building structure (walls, entrance) and then pursue an analysis of the architectural plan through the division of the whole space into six distinct architectural spaces, which structure the following room-by-room (or zone-by-zone) sequence. The subject then treats each room separately and sequentially, recopying symbol after symbol. (The architectural function of these rooms and symbols are not always recognized and does not seem to be the main concern of these subjects.)

In the second strategy, named the strategy of the “visitor,” subjects also take care of the global nature of the plan first (the main four external walls), but then analyze the building and its content through a virtual walk. Subjects usually start with the main entrance, virtually walking along corridors, mentally opening doors and discovering spaces. In front of a specific room “furnished” with various architectural symbols, subjects make deductions from their personal spatial experience to determine its main function (“*this is a bathroom, I recognize the toilet seat,*” “*these must be some stairs,*” . . .) and then recopy the room and its units. This approach also derives from a zone-by-zone approach but is considered as more “egocentric.”

The third and last strategy, called the strategy of “the IKEA® addict,” is close to the previous one except that subjects do not take a virtual walk into the building but rather immediately focus on equipment and furniture. They usually recopy the main four external walls as well as the six main “boxes” of the architectural plan, as a first geometric structure of the drawing, and then go from room to room, without dis-

tinct order, recopying in priority the architectural symbols they recognize (i.e., furniture or equipment). Verbatim in this case is close to “*ha, this is a chair and its desk . . . and here is another one!*” as they recognize the symbols of the chair and the desk and then recopy them (Fig. 5).

Subjects occasionally changed from one strategy to another, mainly at the end of the process. For instance, when the “IKEA® addicts” had considered all the symbols they were able to recognize, they then generally adopted a more “structural engineer” approach to recopy the symbols that made no particular sense for them. However, overall subjects stucked to relatively constant strategy during the whole process of recopying the sketch.

As Figure 6 shows, 13 subjects out of 20 adopted a “structural engineer” strategy, five adopted an “IKEA® addict” approach whereas just two subjects were observed taking a “visitor” approach. There is no clear link at this point between the strategy adopted and the specific background of each subject.

Taking into account these preliminary results, we observe that different subjects, with various level of knowledge about architectural design (from “none” for software engineers or the cognitive psychologist to “some” for the junior mechanical and architectural designers), share three mechanisms in considering, understanding and recopying the blurred architectural sketch. None of the subjects are professional architects, and therefore their level of knowledge can be compared to an expert knowledge-based interpretation system: these complementary mechanisms therefore constitute interesting clues, with low-level abstract data supports tools can deal with.

In terms of visual interpretation, the zone-by-zone (or room-by-room) approach is the most common, whatever its temporality of appearance inside the process. All participants quickly figured out the symbolic meanings of the main pieces

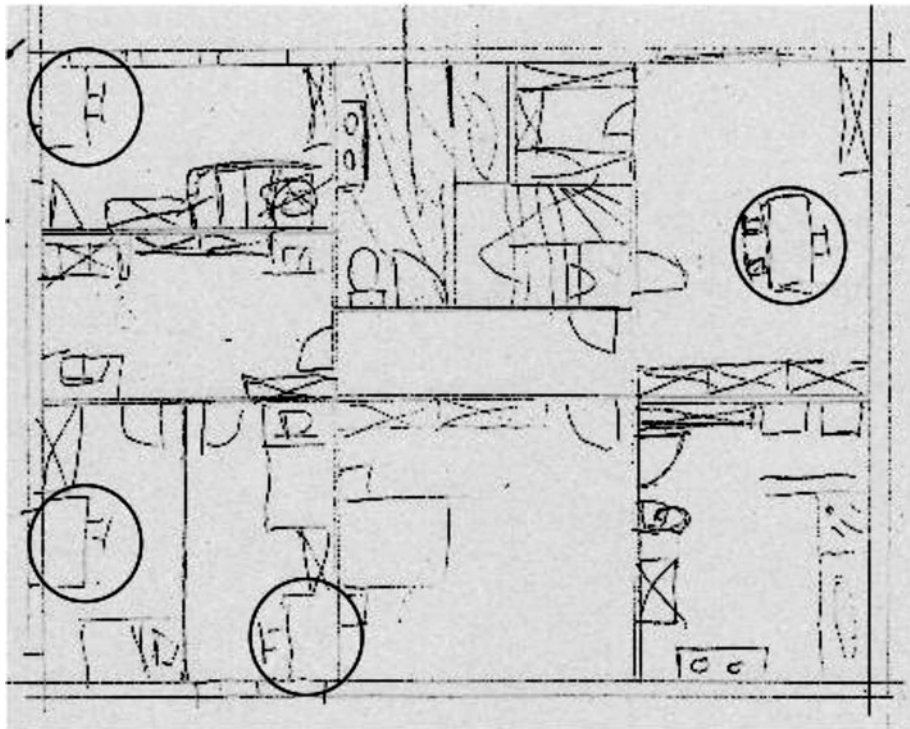


Fig. 5. The blurred architectural sketch to be copied and the various chairs and desks appearing in the plan (circled).

of furniture (the doors, toilet seats, desks, or chairs, for instance, made no difficulties, whereas the beds or the high shelves were sometimes misunderstood). They also instantly recognized the main graphic features of the plan like the main walls, the entrance points and stairs.

In terms of graphical content, subjects quickly understood the main symbols, but it is more important that they were able to manage them even if they were incomplete, ambiguous, or faintly drawn. Subjects did not seem to attach an importance to the thickness of strokes. Moreover, they dealt almost implicitly with nonprimary lines, one of the features that make architectural sketches especially difficult to compute (Fig. 7). A stroke can actually be shared between different symbols (a table drawn against a wall, for instance: both share a common stroke) and this way nurtures different parts of the sketch and different levels of abstraction.

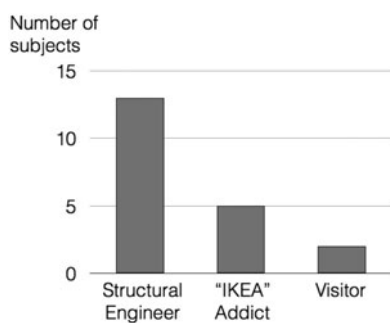


Fig. 6. The distribution of subjects between the three main strategies.

Participants visually understood graphical annotations, like links and arrows even if they cross over other symbols and have no fixed locations. They also easily handled free-form objects like walls (whose shapes cannot be easily described by predefined rules), even if sometimes they did not attach the correct semantic or functional meaning.

A last important observation is that subjects encountered no particular difficulty in copying and understanding an "off-line sketch" (i.e., participants do not know the chronological way it was originally generated). There is consequently no need for copycats to access the synchronous data: an asynchronous approach is sufficient.

4.2. Tragere experiments' results

The Tragere experiments examine how designers generate, then perceive and capture a sketch to obtain clues about

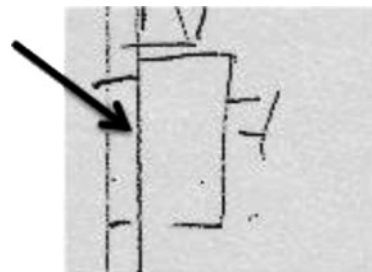


Fig. 7. A nonprimary line.

when, why, and how product design sketches should be supported. In contrast with the previous experiment, participants actually did design, and therefore may attach more importance to how they draw and contribute to the design itself. This aspect of the Tragere experiment has a limited effect on the validity of the results, because it was observed that the group 1 “generators,” knowing that their sketches were going to be later reused, put a bigger emphasis on which graphic clues they wanted to communicate. In contrast, the follow-up designers knew they had to deal with sketches that were not originally theirs, and therefore did mention more clearly which elements they were taking into account (or neglecting) and why. This way, the Tragere experiment provides a wider variety of strokes and representations and is in the meantime closer to actual design processes.

The first result concerns the type of representations usually generated during preliminary product design. Figure 8 and Figure 9 show the value of sections, elevations and perspectives for product design. In contrast to what has been previously demonstrated in architecture, the third dimension developed through perspectives seems to strongly support the ideation phases in product design. Figure 9 also shows how elevations and perspectives are the preferred support for crystallizing ideas and making choices.

This particular characteristic of perspective being central to ideation is also supported by Figure 10. We coded the pro-

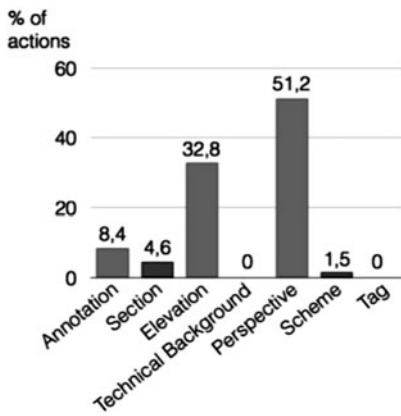


Fig. 8. Distribution (in % of actions) between each type of representation.

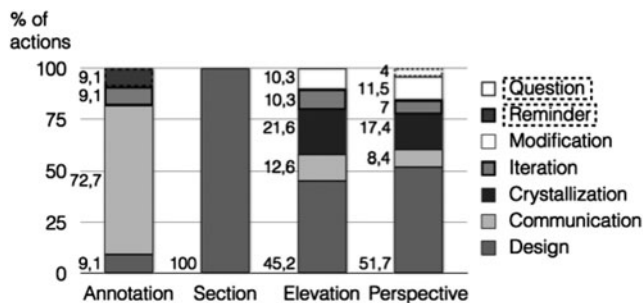


Fig. 9. Distribution (in % of actions) between each type of representations' uses.

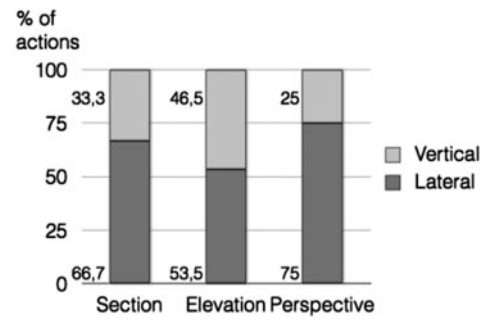


Fig. 10. Vertical and lateral transformations supported by the three main representations.

jects' changes using lateral and vertical transformations (Goel, 1995). This figure shows how these transformations occur in each of the three main representations. Perspectives in particular support the generation of variants, typical of a preliminary design process, whereas elevations (and, to a lesser extent, sections) are more prone to support the deep assessment of a particular solution (i.e., vertical transformations).

Next, the graphic elements of those representations are considered (Fig. 11). Product design sketches do not present the same content as sketches in other design fields. The symbols that structure architectural sketches are almost absent in product design, where only a few geometrical primitives and axes structure the drawing. In the product design sketching, initial strokes are loose and blurry and then crystallize through repetition of strokes and eventual emphasis on a specific one.

Because perspectives are so meaningful for generating a range of ideas during the ideation stages in product design, one might see automatically generated 3-D models as an important way to support preliminary design. If dynamic 3-D representations could bring interesting visual feedback (at least at a later stage, as for architectural design), we nevertheless wanted to evaluate why and how this transition could be of real help to designers. In order to do so, we analyzed the shifts that occurred on paper between 2-D representations (elevations, sections, schemes, . . .) and 3-D representations (perspectives).

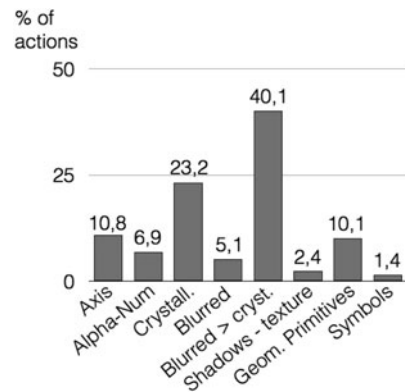


Fig. 11. Types of strokes inside product design sketches.

These shifts were motivated by various reasons, as tracked by the “think aloud” protocol. Three tendencies are underlined in Figure 12:

- shifts from 2-D representations to perspectives are largely caused by a need to generate new ideas (other variants);
- shifts from perspective to 2-D representations respond to a need to simulate and evaluate (mainly dimensions, assembly, conflicts between components, and so on);
- both types of shifts reflect a need to synthesize ideas and to synchronize different elements of the project into a global solution.

The visual, gestural, and verbalized clues of mental activity of the subjects were compared to the visual representations that they created. Figure 13 shows that these clues are quite consistent with the representation used at the same time. Because representations consequently (and quite logically) seem to match the mental state, one could assume that shifts between 2-D and 3-D representations do also match the mental shifts between both dimensional mental states.

These internal and external shifts occur continuously throughout the design process. One might ask if they are simple “rerepresentations” of an idea (e.g., drawing in a different

perspective), that is useful for postideation evaluation of this idea (as in architecture), or if they are integral part of the ideation process itself.

Figure 14 shows that modifications of ideas (evolutions of the project) manifest themselves almost equally as elevations and perspectives. Both of these representations support the evolution of the project; in other words, none is a simple re-representation of the other. This is not the case for sections, as they do not appear to support any modifications.

Shifts from one type of representation to another therefore match mental evolution from one dimension to another, but also a conceptual evolution of the project being designed. Figure 15, Figure 16, and Figure 17 illustrate this concept. Figure 15 represents one state of the project, expressed as an elevation. Figure 16 takes a different point of view but also makes the project evolve in various aspects: another variant is proposed for the foot of the table for instance. Figure 17 is also a 3-D representation of this object but again is not limited to a simple rerepresentation of the previous states: the project has evolved, and the CAD model involves more than its two constituent drawings.

Given the potential of perspectives to support ideation and given how shifts conceptually encourage the evolution of the project, one might ask if the generation of their numerical alter ego, the 3-D models, should be *automatically and simul-*

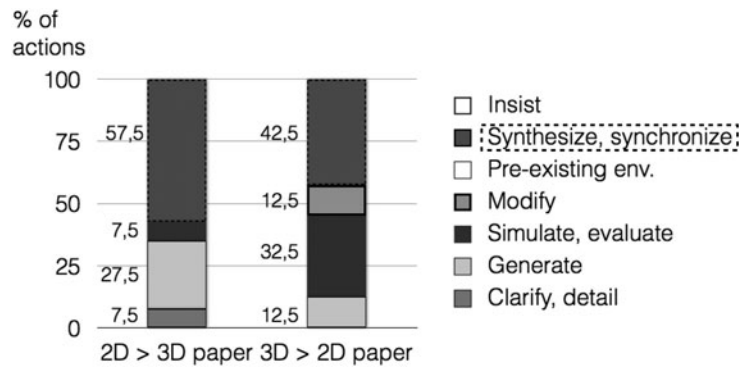


Fig. 12. Two-dimensional > three-dimensional shifts and their causes.

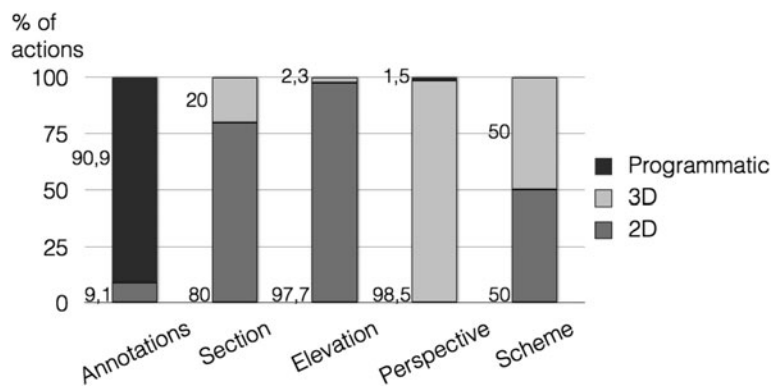


Fig. 13. Use of representations and mode of thought.

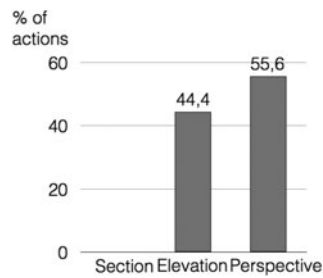


Fig. 14. The percentage of modifications with each type of representations.

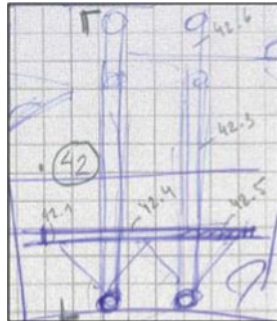


Fig. 15. Evolution of the concept through shifts. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

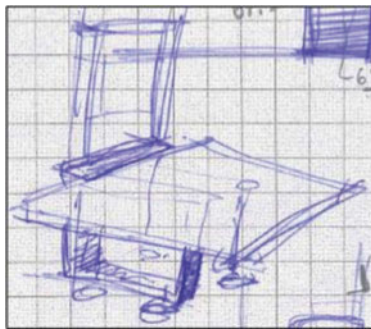


Fig. 16. Evolution of the concept through shifts. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

taneously done. If 2-D to 3-D paper transformations are of such importance for the generation of ideas (and vice versa), would not an automatic transformation from sketch to 3-D models lower (or even degrade) the overall conceptual quality of the process?

Based on the results presented, software engineers would be well advised to respect the slow and iterative building process of the 3-D model instead of imposing a premature 3-D interpretation of the work in progress. If automatic assistance is desired, designers should at least be able to freely switch between 2-D and 3-D representations in order to generate ideas on one medium, simulate these ideas in the second and then synthesize (and add detail) given the feedback this visual conversation would have provided. If needed, these



Fig. 17. Evolution of the concept through shifts. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

concepts' evolution and modifications could be bi-univocally linked on each type of representation, that is designers would have the ability to see modifications they implemented on the 3-D model appear on the 2-D linked representation and vice versa. This bi-univocity should nevertheless stay optional, in order to preserve the natural evolution of concepts from one representation to another, from one mental state to another. The juxtaposition of various types of representations, nurturing a certain level of abstraction and incompleteness, could be crucial for the overall evolution of the project.

If structural symbols (i.e., sketches for stairs, doors, windows), furniture symbols (i.e., sketches for a desk or a couch) as well as a few lines for the main walls constitute the main key features of architectural representations, we observed that the graphic grammar of product design representations is substantially different. Figure 11 shows that these symbols are almost completely absent and that strokes, cycles of strokes, and geometric primitives constitute the only constant features of those product design drawings.

Tracking the presence of “principal” curves (the ones that “propagate” throughout the design process) and “secondary” ones (that disappear or do not strategically structure the drawing), we realize that they are built on some systematic graphical principles that are identical to these main key features (Fig. 18). Principal curves are mainly composed of crystallized and repeated strokes, or by quickly performed B-R-C cycles of strokes. Secondary curves, on the contrary, stay blurred or light, whereas details like shadows or textures might disappear at some stage of the process.

Observing afterward how designers from the second group (the idea pursuers) appropriate the sketches left by the designer-generator, interesting connections between type of curves and type of appropriation could be done. We observed that designers could appropriate the sketches left by the generator following different principles: the appropriation could be total (the “group 2” designer recopying the drawing before making it his/her); partial (only some parts of the drawing being recovered); only visual (the group 2 designer visually evaluating the proposition before starting his/her own) or even totally absent (the pursuer neglecting the work of

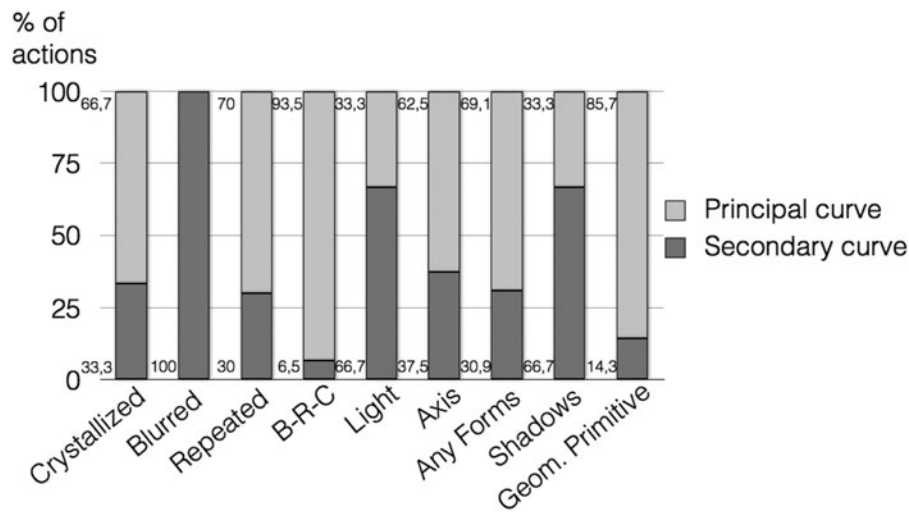


Fig. 18. Graphical content of principal and secondary curves.

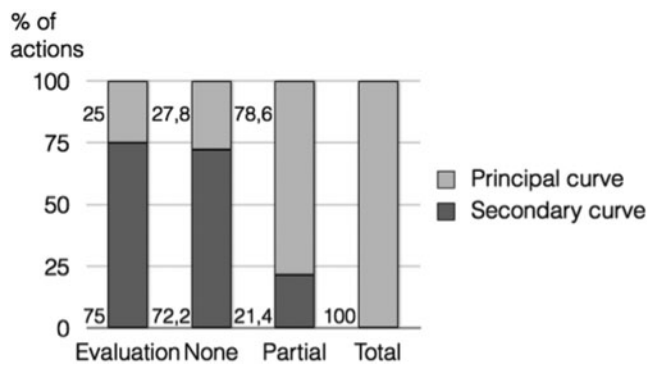


Fig. 19. Types of curves and extent of appropriation.

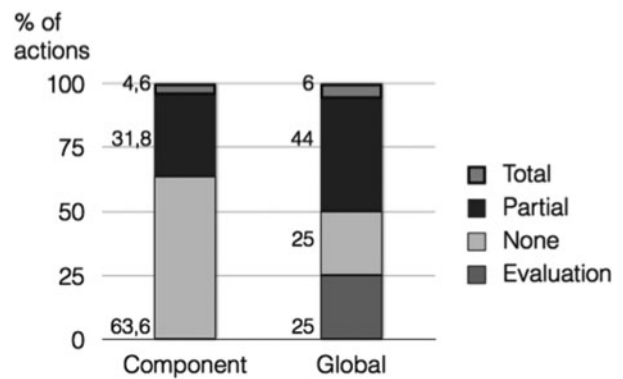


Fig. 20. The extent of appropriation given the global nature of the graphic feature.

his/her virtual colleague and starting from scratch). Figure 19 shows how the principal curves are the ones totally or partially recovered, while secondary curves are mostly only visually evaluated or even neglected.

Meanwhile, Figure 20 illustrates us how global features of sketches (global forms, profiles, . . .) are considered more frequently than components (details, annotations, . . .).

Principal curves, built upon a succession of blurred, repeated, crystallized strokes or geometrical primitives, are therefore the main visual information designers generally exploit in order to capture the visual sense of a representation. These principal curves consequently are the best clues software engineers have at their disposal to capture and to reconstruct product design sketches. Global shapes, in contrast, constitute sufficient support to pursue ideation processes. Designers just seem to need the whole picture to go on with a conceptual idea, leaving the details aside.

Considering that all these clues constitute a grounded basis for 3-D model reconstruction, there are still limitations. The high implicit and blurred content of sketches still make them very difficult to capture, and the absence of symbols

(as shown in Fig. 11) makes a semantic interpretation of product design sketches difficult, even impossible.

The chronological evolution of sketches' states (secondary or principal curves; complete or incomplete in content, see Fig. 21) moreover demonstrates how constantly evolving the contents are, and how incomplete the drawing might stay during preliminary design processes. The connected "complete and principal curves" points on the graph constitute the best chances for the automatic generation of a coherent and useful 3-D volume, which means that given the cyclic construction of those principal curves, this automatic generation should occur once most of the crystallized strokes are done.

In the field of product design, assistance through the generation of 3-D models should carefully consider two points: the necessity of automation (given the importance of shifts for the conceptual evolution of the project) and the temporality of treatment like beautification, given the importance of the cycles of strokes for the global differentiation of principal and secondary curves.

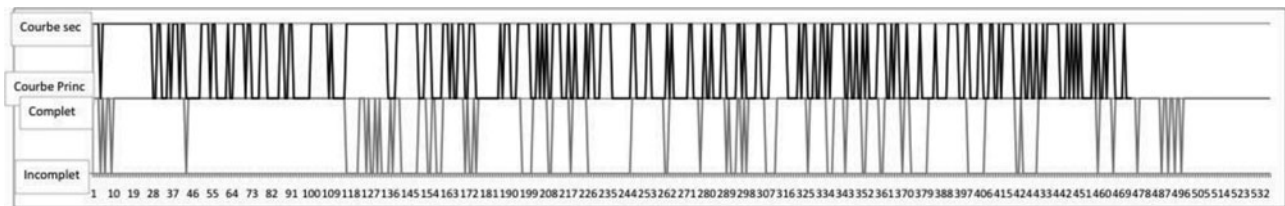


Fig. 21. Connected “complete and principal curves” points for a potential three-dimensional volume generation.

4.3. SBIM for architecture and product design: Discussion

In light of the previous results, this section will provide answers to the research questions presented in Section 3, respectively for architecture and product design.

Are certain “types” of interpretation better adapted to the design fields we are examining? How should interpretation be adapted to different design fields?

The results of this paper suggest that a semantic approach remains an appropriate way to coherently interpret sketches in architectural design. This is because architectural representations are mainly two-dimensional and encode a substantial amount of symbolic content.

Leclercq (1994) points out that architectural perspectives are mainly used for communicative and evaluative purposes, and that architectural sketches usually lay out building and furniture units on a horizontal plane of reference. Because of this common practice, designers can rely on simple and quick extrusions of walls (like those generated by EsQUIsE) during preliminary design.

However, our results show that complementary strategies, such as a zoning approach, are consistent with the way human beings perceive and process architectural sketches and offer valuable clues that can address the computational complexities and inefficiencies like EsQUIsE still encounters. Notably, these additional strategies would ease the computation of nonprimary lines, which are currently not handled by EsQUIsE, and would avoid the need of chronologically consistent symbols, as required by Macé and Anquetil (2009).

As for product design, our results show that the absence of symbols as well as the crucial role of perspectives during ideation make semantic interpretation difficult. New types of interactions for 3-D modeling and/or geometric reconstructions have to be found. Given the continuing importance of 2-D representations (like elevations and sections) for the iterative development of concepts, solid sketch or digital sculpting approaches should not be exclusive of other forms of interaction. Moreover, sketching in a 3-D environment (3-D sketch) should be done in parallel with 2-D inputs to more closely mimic the way designers draw naturally.

The Tragere process of building on each-others' sketches illustrates that the global nature of the project is more impor-

tant than the details, thus supporting a zone-by-zone approach over a chronological approach.

What elements of a representation should be considered effective as input data for SBIM tools for preliminary design?

In order to limit the combinatorial explosion of possible interpretations, software engineers *have* to develop systems that quickly focus on specific types of input data. We defined three different strategies for the perception and interpretation of an architectural sketch, but found that participants understood key symbols in the same way (functional or furniture symbols). Ambiguous, blurred, roughly drawn, and nonprimary graphical content was correctly characterized by participants, even those with no architectural background.

When considered in its immediate context (i.e., main walls of the room and other nearby symbols), each symbol can be understood semantically using elementary space recognition. The main structure of the building, regardless of the walls' thickness, constitutes the geometric basis for the overall layout. This set of graphical units offers the best clues for defining computer interpretation analogous to human perception and recognition.

As for the field of product design, our results showed that sketches are built upon specific cycles of strokes (B-R-C cycles, then crystallized strokes eventually forming principal curves) that constitute the main drawing's key features. We believe that this cycle of strokes is the externalization of the see–transform–see process (Schön & Wiggins, 1992) and impacts sketches' perception and recognition as well. Therefore, sketches should not be beautified and treated as soon as they are drawn. The crystallization process itself is part of the design process, and the materialization of principal curves is a crucial step for the global coherence of the project. There is a need to preserve their ambiguity and allow the designer sufficient time to fully develop them before the computer processes them. This observation is consistent with a zone-by-zone approach to interpretation rather than a chronological approach.

What is the appropriate timing of sketch assistance in design tools?

In addition to the timing of beautification, software systems make other assumptions about the timing of sketch processing.

The literature seems to agree on the need to provide a real-time, automatic generation of the 3-D models, and in the meantime decisions are taken concerning the univocity between the numerical sketches and 3-D models.

Regarding the potential need for real-time 3-D models during the architectural design process, Darses et al. (2008) stressed that 3-D models generated by EsQUIsE were not used as extensively as one might expect (only 10% of the whole sketching experiment). Even if the 3-D models were highly desired by designers and even if researchers captured a great deal of visual and gestural clues to 3-D mental activity, 2-D externalizations seemed a sufficient medium for architectural ideation.

One might conclude that 3-D models in architecture add value to the design process, but should only be created after the concept generation phase, and after floor-by-floor design. This delayed visual feedback can then support a “whole picture” approach instead of a stroke-by-stroke incremental approach and does not require some biunivocity between 2-D sketches and 3-D models. Our results support this point of view: participants of the “Port-Zeeland” experiment did not seem to be bothered by the off-line character of the representation, which leads us to recommend an asynchronous, zone-by-zone interpretation of blurred architectural sketches.

In product design, the analysis of shifts between 2-D and 3-D representations (and their causes) as well as the modalities of modifications suggest that 2-D to 3-D (and vice versa) transformations are key to the design process. They nurture the conceptual and abstract evolution of the object being designed and are a generator of new features instead of being just rerepresentations of the same information (as they can be for architecture). They therefore hold a particularly important place inside the design process.

If software engineers opt for an automatic generation of 3-D models based on 2-D sketches, we suggest that they consider the following:

- realize that automating the 2-D > 3-D transfer might affect the quality of the ideation process, might take away some control from the designer, and thus increase the complexity of the overall design process;
- allow designers to move seamlessly back and forth between 2-D representations and 3-D models in order to keep the ideation process active;
- allow direct modifications to both 2-D representations and 3-D models, and thus preserve the possibility of “paperlike” univocal modifications (with the automatic capture of the different states as a record for efficient design-rationale traceability);
- but, in the meantime, suggest biunivocal modifications (between 2-D and 3-D states of the project) as an “augmented” feature of the 2-D > 3-D transfer, in order to have immediate feedback on the applied modifications.

Generally speaking, studying both architecture and product design sketching in parallel helped us realize how specific

their visual representations were, how different some of their processes were and consequently how important it is to define context- and process-specific dedicated support tools.

We do offer one recommendation for both disciplines: researchers should focus on how designers can benefit from the complementary aspects of tools and representations in each discipline, instead of arguing in favor of one or the other. Current tools and representations may be used all along the design process, and perhaps more closely mimicking designers processes would prove itself the best strategy.

5. A NEW FRAMEWORK

Based on our results and the above discussion, we introduce two strategies to support ideation during the preliminary phases of design. The first is NEMo, a prototype tool to support architectural design, and the other is PEPS³ (for “product design evolution through purposeful sketch support system”), a preliminary framework for product design.

5.1. NEMo: A dedicated design support tool for architectural ideation

NEMo is an experimental prototype that asynchronously interprets architectural floor plan sketches in order to provide rich postideation, visual feedback during the idea evaluation processes (Fig. 22 and Fig. 23). The Port Zeeland experiments provided a number of results that call into question assumptions about how SBIM systems should function. The design of NEMo takes into consideration these Port Zeeland findings and revisits some of the limitations of the current semantic interpretation systems, such as EsQUIsE (NEMo stands for “New EsQUIsE Modeler”).

Most existing sketch recognition systems target diagrammatic sketches such as Unified Modeling Language diagrams or electronic circuit schematics, made of well-defined symbols linked together by connectors (e.g., lines or arrows). These systems make the assumption that symbols and connectors are exclusively composed of distinct strokes and mostly drawn one after the other. On this basis, the stroke is the main entity considered during the recognition process, which consists of finding nonoverlapping clusters of temporally and spatially related strokes that match the symbols. Although it could restrict drawing freedom, this assumption is acceptable for diagrams.



Fig. 22. NEMo in its actual state.

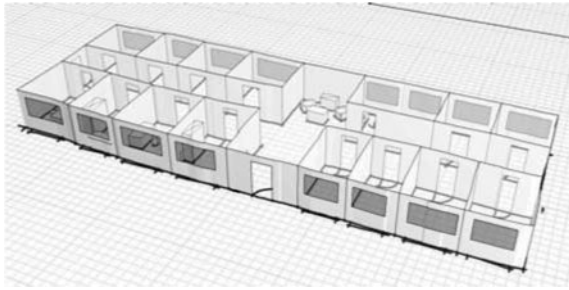


Fig. 23. NEMo in its actual state.

Architectural sketches contain many shared strokes, or nonprimary lines. Stroke clustering is a common way of segmenting drawings (i.e., identifying distinct objects), but it is ill suited to handling shared strokes as it has to face the combinatorial explosion of possible (maybe overlapping) clusters of strokes.

In the Port Zeeland results, we observed that participants focused on subdividing and organizing architectural space, which makes dividing into zones (or regions) an effective strategy for interpretation. Instead of identifying groups of strokes, NEMo identifies perceptual regions in the sketch using perceptual heuristics (Saund, 2003; Wuersch & Egenhofer, 2008). This way, NEMo is able to recognize symbols containing shared strokes and achieve more effective segmentation. It does not require the designer to draw in an unfamiliar way and, therefore, better suits the nature of an architectural sketch.

The Port Zeeland experiments also suggest that several strategies and “spaces of interpretation” could coexist. Consequently, we argue that the ability to use different strategies in parallel is an important feature to increase the robustness of a sketch recognition system. It enables the system to cross-validate interpretation hypotheses generated by different approaches in order to resolve ambiguities. For example, the recognition of walls by one process will facilitate the segmentation task of another process for recognizing furniture.

In this regard, the computer model underlying NEMo is inspired by the Copycat program (Mitchell, 2001) that aims at discovering analogies between letter strings. NEMo exploits the multiagent paradigm, making seamless use of heterogeneous methods for recognizing different types of graphic objects possible (Casella, Deufemia, Mascardi, Costagliola, et al., 2008). Knowledge is distributed between several agents that cooperate and compete to build a global sketch interpretation: some of them might be responsible for sketch segmentation, some for recognizing architectural symbols or textual annotations, and so on. Because of its multiagent architecture, the system is able to use different strategies in parallel to perform the same task and, in doing so, it improves its robustness. For example, segmentation can be performed by using perceptual regions extraction (Saund, 2003; Wuersch & Egenhofer, 2008), by exploiting connected components, or by grouping strokes (Peterson et al., 2010).

All interpretation hypotheses are built in a common global workspace. This shared structure enables indirect communication between agents and between various strategies: hypotheses built by one agent will exploit, reinforce or compete with hypotheses built by other agents. This active structure supports a continuous competition between hypotheses: winning hypotheses gain activation, others lose it; when the activation of a hypothesis falls to zero, it is discarded. This specific method presents two advantages: first it avoids the combinatorial explosion of the number of hypotheses stored in the workspace, and second it allows initially weaker hypotheses to survive for some time, giving them a chance to be consumed by higher-level structures or to be reinforced by contextual relations.

Another important feature of NEMo is its adaptive behavior. Unlike conventional deterministic systems, this behavior is not planned beforehand but depends on a population of processing agents that evolve during the interpretation process. Each agent has a priority value that determines the speed at which the task will be executed. Agents searching for more common or more promising structures will have a higher priority value. This allows more favorable hypotheses to be explored faster. For instance, if a letter hypothesis, which is probably part of a word, is instantiated in the workspace, agents looking for other letters close to it will be added to the system, increasing the probability of other letters to be found in the neighborhood. The evolution of the agent population is driven by a fixed set of knowledge agents that reacts to the instantiation of new hypotheses in the workspace, by adding one or more processing agents in the system. These can be bottom-up agents, which will try to use the previously found hypotheses to build higher-level structures, or top-down ones that will look for contextually related objects. The latter enable to perform deeper exploration in order to find the expected object (using for instance less usual thresholds).

Figure 24 illustrates the overall NEMo model. The system consists of three main components:

- the workspace, the shared structure where interpretation hypotheses are built;
- the dynamic population of processing agents that implements all processing tasks related to sketch analysis;
- the set of knowledge agents which contain high-level knowledge and drive the adaptive behavior of the system.

NEMo, unlike EsQUIsE, is therefore able to use different strategies in a parallel mode to analyze a sketch, thus improving its robustness. It is capable of handling competing interpretation hypotheses and can therefore explore several contradictory solutions and recover from recognition errors. Moreover, it exploits the “island of certainty” formed by existing, strong hypotheses to adapt its behavior and to look for more promising interpretations. In this way, the system is able to explore the huge space of possible interpretations more efficiently and to create a better 3-D interpretation.

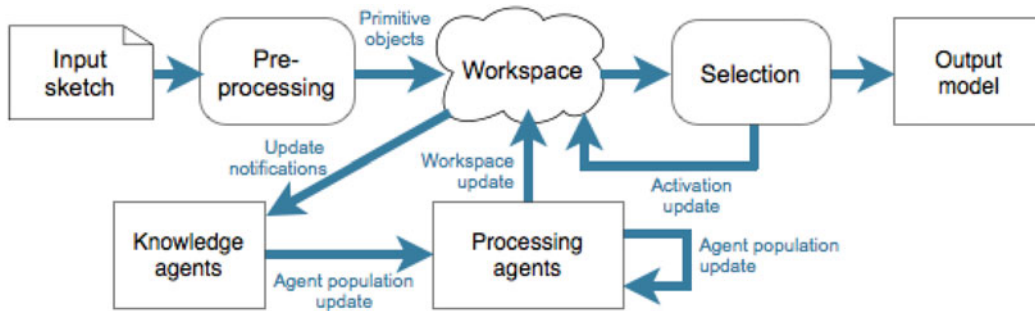


Fig. 24. Overall functioning of the proposed model. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

Finally, NEMo differs from most other state-of-the-art systems because it is asynchronous. It is designed to interpret an already completed architectural sketch, rather than provide continuous interpretation while the sketch is being drawn (like online sketch recognition systems). This key feature is grown from our research, and preferred because the recognition system will not interfere with the designer's creative process. We emphasized that as soon as the 3-D model is not useful during the whole architectural ideation process but only at some intermediate steps (Darses et al., 2008), immediate feedback is not required. Because it is asynchronous, it avoids any chronological constraint (e.g., drawing one symbol after the other) and enables any changes and deletions in previously drawn symbols. It is, as a result, more compatible with a naturalistic, creative design process.

From a computational performance perspective, online sketch recognition may seem attractive as it enables a better use of available computer resources (most are idle during drawing). But again, this type of recognition can only be truly exploited if the sketch is made of distinct objects, recognized one after the other while they are drawn, a feature not shared by architectural sketches.

Moreover, an asynchronous approach allows simpler editing and modification of sketches, such as erasing. Most online systems are complicated by the superfluous, incremental nature of the interpretation and do not permit such operations. In the future, an asynchronous system coupled with a dedicated strokes extraction algorithm (Rajan & Hammond, 2008) might be able to analyze a scanned paper sketch. This can be especially beneficial because nondigital pen and paper still remain the most natural tools to support creative work.

5.2. PEPS³: A dedicated conceptual framework for SBIM in product design

Based on findings from the Tragere experiments, we propose an initial framework for SBIM in product design, named PEPS³.

This framework is built upon understanding of users' needs and practices with ramifications for software engineers. Our results have shown that automatic, real-time generation of a 3-D model can potentially slow down the design process

even if 3-D representations (contrary to architecture) are still crucial during the whole product design process. Instead, our strategy opts for assisted reconstruction of the 3-D model, in a synchronous and interactive way.

The framework for the future system is represented in Figure 25. It consists of two distinct layers:

- the top layer shows the process designers might follow in order to transform preliminary sketches into a responsive, flexible 3-D model; and
- the bottom layer suggests some simple, intuitive tools and functionalities for manipulating the data.

The framework enables the designer to begin by either drawing using predefined plans or immediately start tridimensional modeling.

Whatever the chosen method, the first step enables the designer to introduce background technical or formal plans, or any other kind of existing environment useful for initiating the design process (step 1, layer 1). The designer can then apply geometrical primitives or axes in order to structure the drawing or model (step 1, layer 2). These primitives can immediately be "beautified" so that the designer can take advantage of the geometrical accuracy in order to sketch more easily. If the primitives are 3-D, positioning and managing can be done either through pen or haptic interaction.

Next, the designer builds the blurred sketch using a pen input (step 2, layer 1), without any kind of beautification or in-

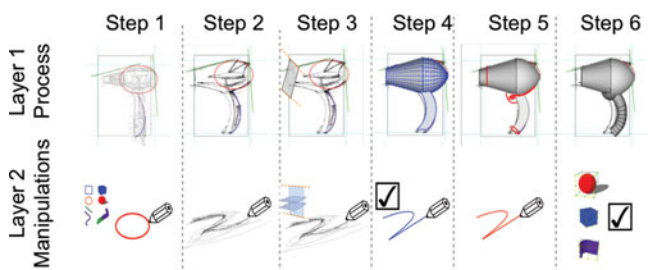


Fig. 25. Conceptual model for a sketch based interface for modeling for product design. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

terpretation until the designers requests it (step 2, layer 2). For representations such as elevations, sections, perspectives, . . . drawn flat as on a sheet of paper, the strokes would be by default attached to a reference plan, perpendicular to the axis of view. If the designer wants to create a drawing that could later become a 3-D model, he/she should then develop the other sides of his object by defining and positioning new drawing planes inside a 3-D world (step 3, layer 1). Structural guides and grids could be used if the designer wants to make sure that perspective, symmetry, or orthogonal rules are respected (step 3, layer 2). The spatial positioning of the reference plans might be difficult to realize through pen interaction, but this will be tested after implementation. These reference planes present the huge advantage of anchoring drawing on a 2-D structure, closer to human visual principles, than sketching directly in a 3-D world, without any kind of control on the "depthness of the drawing move."

Once all the facets of the object are drawn (and after potential modifications are made at this stage), the designer can choose to declare principal curves (edges, profiles, strength lines, and so on, step 4, layer 2) using the blue input pen. These curves will connect several points on various reference planes and will form a wireframe 3-D structure (step 4, layer 1). The system would then, on demand, generate the skins around the wired structure to compose the 3-D volume.

Once the 3-D volume is created, it can anchor modifications: dynamic modification of profiles, deformation of volumes, adding of details, and so on, just as supported by many prototypes tools presented in the state of the art (step 5 layer 1). A specific pen (red, for instance) could be used to specify that modifications are being implemented (step 5, layer 2). Some (gestural) interactions have to be determined in order to handle details like voids or to control the change in volume depth. These modifications could, on demand, be univocal or bi-univocal to allow the designer to freely shift from 2-D to 3-D views.

This 3-D structure, once validated, could then be exported to a CAD tool in order to proceed to production modeling. The format of the export should be as universal as possible and should preserve the 2-D/3-D dynamic structure of the object being designed. The system would finally maintain any variations (several layers organized inside a hierarchical tree for instance), in order to enable the designer to compare several variants or come back to an old state to input other ideas.

6. CONCLUSIONS AND FUTURE WORK

This paper underlines the value of designer's needs, practices and uses of tools in the development of SBIM. Two case studies examine assumptions about designer's sketch behavior in both architecture, with its highly 2-D, symbolic representations, and product design, with its highly 3-D, fluid representations.

Some significant results are presented regarding strategies of perception and recognition, generation of 3-D volumes

(pertinence and temporality of assistance); the 2-D > 3-D shifts (their relations, their reactivity to modification) and treatment of freehand sketch features (pertinence and temporality).

Differences between both fields reveal the complexity of offering universal "augmented" support, so we offer two different responses based on our findings. First is NEMO, a robust, ready for testing multiagent system for architects that interprets asynchronous, blurred architectural free-hand sketches. Second is the PEPS³ framework, an initial model that addresses needs, processes, and methods to support preliminary phase of product design.

The different methodologies used to capture the data and to analyze it, as well as the limited number of participants, points to the need for further work in order to evaluate the representativeness of the results.

Future work regarding NEMO will include evaluation with end users, in order to validate its robustness and to ensure that it supports realistic design scenarios. NEMO builds on EsQUISE and overcomes some of the older system's limitations through different design choices and software architecture. PEPS³, in contrast, has now to be implemented with the help of software engineers. A first rough prototype will then have to be evaluated in real-working environment.

REFERENCES

- Alvarado, C., & Davis, R. (2004). SketchREAD: a multi-domain sketch recognition engine. *Proc. 17th Annual ACM Symp. User Interface Software and Technology (UIST 2004)*, pp. 23–32.
- Achten, H., Dorst, K., Stappers, P.J., & De Vries, B. (2005). A decade of design research in the Netherlands. *Design Research in the Netherlands Symp.*
- Achten, H., Vries, B., & Jessurun, A. (2000). DDDoolz—a virtual reality sketch tool for early design. *CAADRIA 2000: Proc. 5th Conf. Computer Aided Architectural Design Research in Asia, Singapore* (Tan, B.-K., Tan, M., & Wong, Y.-C., Eds.), Centre for Advanced Studies in Architecture, pp. 451–460.
- Alvarado, C. (2004). *Multi-domain sketch understanding*. PhD Thesis. Massachusetts Institute of Technology.
- Alvarado, C., & Davis, R. (2004). SketchREAD: A Multi-Domain Sketch Recognition Engine. *Proc. 17th Annual ACM Symp. User Interface Software and Technology (UIST 2004)*, pp. 23–32.
- Aoyama, H., Nordgren, A., Yamaguchi, H., Komatsu, Y., & Ohno, M. (2007). Digital style design systems from concept to sophisticated shape. *International Journal on Interactive Design and Manufacturing 1*, 55–65.
- Bae, S.-H., Balakrishnan, R., & Singh, K. (2008). ILoveSketch: as-natural-as-possible sketching system for creating 3-D curve models. *Proc. 21st Annual ACM Symp. User Interface Software and Technology*, Monterey, CA.
- Basa, I., & Senyapili, B. (2005). The (in)secure position of the design jury towards computer generated presentations. *Design Studies 26*, 257–270.
- Bastien, J., & Scapin, D. (1995). Evaluating a user interface with ergonomic criteria. *International Journal of Human-Computer Interaction 7*, 105–121.
- Bilda, Z., & Gero, J. (2005). Does sketching off-load visuo-spatial working memory? In *Studying Designers '05* (Gero, J., & Bonnardel, N., Eds.). Sydney: University of Sydney, Key Centre of Design Computing and Cognition.
- Casella, G., Deufemia, V., Mascardi, V., Costagliola, G., & Martelli, M. (2008). An agent-based framework for sketched symbol interpretation. *Journal of Visual Languages and Computing 19*, 225–257.

- Casella, G., Deufemia, V., Mascardi, V., Martelli, M., & Tortora, G. (2008). Reasoning about hand-drawn sketches: an approach based on intelligent software agents. *Proc. 10th Int. Conf. Visual Information Systems (VISUAL 2008)*, pp. 302–314.
- Company, P., Piquer, A., & Contero, M. (2004). On the evolution of geometrical reconstruction as a core technology to sketch-based modeling. *Eurographics Proc. Workshop on Sketch-Based Interfaces and Modeling* (Hughes, J., & Jorge, J., Eds.), pp. 97–106. Eurographics Association.
- Cross, N. (2000). *Strategies for Product Design* (3rd ed.). Milton Keynes: Open University.
- Danesi, F., Gardan, Y., Martin, B., & Pecci, I. (1999). *L'esquisse, définition et utilisation en conception 3D*. Accessed at <http://www.valoriam-univ-ubs.fr>
- Darses, F. (2004). Processus psychologiques de résolution collective des problèmes de conception: contribution de la psychologie ergonomique. *Document de synthèse en vue d'obtenir une Habilitation à Diriger des Recherches*. Paris: Université Paris V–René Descartes.
- Darses, F., Détienné, F., & Visser, W. (2001). Assister la conception: perspectives de reconstruction as a core technology to sketch-based modeling. *Eurographics Proc. Workshop on Sketch-Based Interfaces and Modeling* (Hughes, J., & Jorge, J., Eds.), pp. 97–106. Eurographics Association.
- Darses, F., Mayeur, A., Elsen, C., & Leclercq, P. (2008). Is there anything to expect from 3D views in sketching support tools? *Design Computing and Cognition: Proc. 3rd Int. Conf. Design Computing and Cognition* (Gero, J., & Goel, A., Eds.). Atlanta, GA: Springer.
- Davis, R. (2002). Sketch understanding in design: overview of work at the MIT AI Lab. *AAAI Spring Symp.*, AAAI Technical Support.
- Demeure, A. (2007). *Modèles et outils pour la conception et l'exécution d'Interfaces Homme-Machine Plastiques*. PhD Thesis. Université Joseph Fourier.
- Dessy, J. (2002). *De l'emploi des symboles dans les esquisses architecturales*, Université de Liège.
- Détienné, F., Boujut, J., & Hohmann, B. (2004). Characterization of collaborative design and interaction management activities in a distant engineering design situation. In *Cooperative Systems Design: Scenario-Based Design of Collaborative Systems* (Darses, F., Dieng, R., Simone, C., & Zacklad, M., Eds.). New York: IOS Press.
- Do, E. (1995). What's in a diagram that a computer should understand. *CAAD Futures '05, 1995 Singapore*, pp. 469–482.
- Do, E., & Gross, M. (1997). Inferring design intentions from sketches: an investigation of freehand drawing conventions in design. *CAADRIA 1997*, pp. 217–227, National Chiao Tung University.
- Do, E. Y. (2001). VR sketchpad. *CAAD Futures 2001*, pp. 161–172, Eindhoven.
- Dorst, K. (2008). Design research: a revolution-waiting-to-happen. *Design Studies* 29, pp. 4–11.
- Eggl, L., Brüderlin, B., & Elber, G. (1995). Sketching as a solid modeling tool. *Proc. 3rd ACM Symp. Solid Modeling and Applications*. Salt Lake City, UT: ACM.
- Elsen, C., Darses, F., & Leclercq, P. (2010). An anthropo-based standpoint on mediating objects: evolution and extension on industrial design practices. *Design Computing and Cognition* (Gero, J., Ed.). Stuttgart: Springer.
- Ferguson, E. (1992). *Engineering and the Mind's Eye*. Cambridge, MA: MIT Press.
- Garner, S. (2000). Is sketching still relevant in virtual design studios? *DCNet Conf.*, pp. 1–6.
- Gennari, L., Kara, L., & Stahovich, T. (2004). Combining geometry and domain knowledge to interpret hand-drawn diagrams. *AAAI Fall Symp. Series 2004: Making Pen-Based Interaction Intelligent and Natural*.
- Goel, V. (1995). *Sketches of Thought*. Cambridge, MA: MIT Press.
- Hammond, T., & Davis, R. (2005). LADDER, a sketching language for user interface developers. *Computers & Graphics* 29, 518–532.
- Huot, S. (2005). *Une nouvelle approche pour la conception créative: de l'interprétation du dessin à main levée au prototypage d'interactions non-standard*. PhD Thesis. Ecole Nationale Supérieure des Techniques industrielles et des Mines de Nantes.
- Igarashi, T., Matsuoka, S., & Tanaka, H. (2007). Teddy: a sketching interface for 3D freeform design. *ACM SIGGRAPH 2007 Courses*. San Diego, CA: ACM.
- Jeunejean, A. (2004). *Algorithme de tracé réaliste pour environnement de dessin virtuel*. Master's Thesis. Université de Liège.
- Jonson, B. (2005). Design ideation: the conceptual sketch in the digital age. *Design Studies* 26, 613–624.
- Jozen, T., Wang, L., & Sasada, T. (1999). Sketch VRML-3D modeling of conception. *Architectural Computing: From Turing to 2000*, pp. 557–563.
- Juchmes, R. (2005). *Etude comparative des techniques temp réel d'interprétation de croquis*. Diplôme d'Etudes Approfondies. Université de Liège.
- Juchmes, R., Leclercq, P., & Sleiman, A. (2005). A freehand-sketch environment for architectural design supported by a multi-agent system. *Computers & Graphics* 29(6), 905–915.
- Kanai, S. (2005). Human-computer interactions for digital styling design—a difficult road toward bridging the gap between concepts and 3D models. *Proc. Virtual Concept 2005*, Biarritz.
- Kara, L., Shimada, K., & Marmalefsky, S.D. (2007). An evaluation of user experience with a sketch-based 3D modeling system. *Computer & Graphics* 31, 580–597.
- Karpenko, O., Hughes, J., & Raskar, R. (2004). Epipolar methods for multi-view sketching. *Eurographics Workshop in Sketch-Based Interfaces and Modeling*.
- Kolarevic, B. (2000). Digital architectures. *Proc. ACADIA 2000: Eternity, Infinity and Virtuality*, pp. 251–256 (Clayton, M.-J., & Vasquez De Velasco, G.-P., Eds.). New York: Virtualbookworm.com Publishing.
- Lebahar, J. (2007). *La conception en design industriel et en architecture: Désir, pertinence, coopération et cognition*. Paris: Lavoisier.
- Leclercq, P. (1994). *Environnement de conception architecturale préintégré. Elements d'une plate-forme d'assistance basée sur une représentation sémantique*. PhD Thesis. University of Liège.
- Leclercq, P. (2005). Le concept d'esquisse augmentée. *Proc. SCAN 2005, Séminaire de Conception Architecturale Numérique*.
- Lim, C. (2003). An insight into the freedom of using a pen: pen-based system and pen-and-paper. *22nd ACADIA Conf.: Connecting Crossroads of Digital Discourse*.
- Lipson, H., & Shpitalni, M. (1996). Optimization-based reconstruction of a 3D object from a single freehand line drawing. *Computer-Aided Design* 28, 651–663.
- Lipson, H., & Shpitalni, M. (2007). Correlation-based reconstruction of a 3D object from a single freehand sketch. *SIGGRAPH 2007*.
- Macé, S., & Anquetil, E. (2009). Eager interpretation of on-line hand-drawn structured documents: the DALI methodology. *Pattern Recognition* 42, 3202–3214.
- McGown, A., Green, G., & Rodgers, P.A. (1998). Visible ideas: information patterns of conceptual sketch activity. *Design Studies* 19, pp. 431–453.
- Mitchell, M. (2001). Analogy-making as a complex adaptive system. In *Design Principles for the Immune System and Other Distributed Autonomous Systems* (Segel, L.A., & Cohen, I.R., Eds.), pp. 335–360.
- Nijs, G., Vermeersch, P., Devlieger, P., & Heylighen, A. (2010). Extending the dialogue between design(ers) and disabled use(rs): from conversation to embodied skill. *Int. Design Conf., Design 2010*, pp. 1817–1826, Dubrovnik.
- Olsen, L., Samavati, F.F., Sousa, M.C. & Jorge, J.A. (2009). Sketch-based modeling: a survey. *Computers & Graphics* 33, 85–103.
- Peterson, E.J., Stahovich, T.F., Doi, E., & Alvarado, C. (2010). Grouping strokes into shapes in hand-drawn diagrams. *Twenty-Fourth AAAI Conf. Artificial Intelligence (AAAI-10)*, pp. 974–979.
- Plimmer, B., & Freeman, I. (2007). A Toolkit approach to sketched diagram recognition. *BCS HCI 2007*, pp. 205–213.
- Ramel, J.-Y., Vincent, N., & Emptoz, H. (1998). Interprétation de documents techniques par “cycles perceptifs” à partir d'une perception globale du document. *Revue Traitement du Signal* 15(2), 1–20.
- Rajan, P., & Hammond, T. (2008). From paper to machine: extracting stokes from images for use in sketch recognition. *Proc. 5th Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBIM'08)*, pp. 41–48.
- Robertson, B.F., & Radcliffe, D.F. (2009). Impact of CAD tools on creative problem solving in engineering design. *Computer-Aided Design* 41, 136–146.
- Rodgers, P.A., Green, G., & McGown, A. (2000). Using concept sketches to track design progress. *Design Studies* 21, 451–464.
- Safin, S., Boulanger, C., & Leclercq, P. (2005). Premières évaluations d'un bureau virtuel pour un processus de conception augmenté. *IHM 2005*, pp. 107–114.
- Safin, S., Juchmes, R., & Leclercq, P. (in press). Du crayon au stylo numérique: influences des IHM à stylo et des interprétations numériques sur l'activité graphique en tâches de conception. *Journal d'Interaction Personne-Système*.
- Saund, E. (2003). Finding perceptually closed paths in sketches and drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 475–491.

- Saund, E., & Moran, T. (1994). A perceptually-supported sketch editor. *Proc. 7th Annual ACM Symp. User Interface Software and Technology*. Marina del Rey, CA: ACM.
- Schenk, P. (1991). The role of drawing in the graphic design process. *Design Studies 12*, 168–181.
- Schon, D.A., & Wiggins, G. (1992). Kinds of seeing and their functions in designing. *Design Studies 13*, 135–156.
- Shilman, M., & Viola, P. (2004). Spatial recognition and grouping of text and graphics. *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pp. 91–95.
- Sutherland, I. (1963). SketchPad: a man-machine graphical communication system. *Spring Joint Computer Conf.*, pp. 329–345.
- Suwa, M., Purcell, T., & Gero, J. (1998). Macroscopic analysis of design processes based on a scheme for coding designers' cognitive actions. *Design Studies 19*, 455–483.
- Tian, C., Masry, M., & Lipson, H. (2009). Physical sketching: reconstruction and analysis of 3D objects from freehand sketches. *Computer-Aided Design 41*, 147–158.
- Tovey, M., & Richards, C. (2004). Computer representation for concept design and maintenance instruction. *TMCE 2004*, pp. 107–114, Lausanne.
- Tversky, B. (2002). What do sketches say about thinking. *Proc. 2002 AAAI Spring Symp.*, pp. 148–151.
- Ullman, D.G., Wood, S., & Craig, D. (1989). The importance of drawing in the mechanical design process. *NSF Engineering Design Research Conf.*
- Visser, W. (2006). *The Cognitive Artifacts of Designing*. London: Erlbaum.
- Wuersch, M., & Egenhofer, M. (2008). Perceptual sketch interpretation. *Headway in Spatial Data Handling*, pp. 19–38.

Catherine Elsen is a BAEF postdoctoral affiliate at the Massachusetts Institute of Technology (Ideation Lab) and teaching assistant at LUCID, University of Liège (ULg). She received her PhD in engineering sciences in 2011 (ULg, funded by F.R.S.-FNRS), a Master's in working and social sciences (Research in Ergonomics) in 2009 (CNAM, Universities of Paris 5 and 8 and Bordeaux 2, France), and a MS degree in architecture and building engineering in 2007 (ULg). Her research interests cover design processes (in architecture and industrial design), the impact of design tools on cognitive processes, as well as creative philosophies such as design thinking.

Jean-Noël Demaret received a Bachelor's degree in computer graphics in 2004 and a Master's degree in computer science from ULg in 2007. His Master's thesis was about artificial intelligence and games. He joined the LUCID-ULg research team and started a PhD funded by the Belgian National Fund for Scientific Research (F.R.S.-FNRS). His main research interest covers the use of multiagent and adaptive computer systems for automatic sketch recognition, specifically in the field of architectural design.

Maria C. Yang is the Robert N. Noyce Career Development Assistant Professor of mechanical engineering and engineering systems. She earned her BS in mechanical engineering from MIT and her MS and PhD from Stanford University's Department of Mechanical Engineering. She is the 2006 recipient of an NSF Faculty Early Career Development (CAREER) award. Her industrial experience includes serving as the Director of Design at Reactivity, a Silicon Valley software company that is now part of Cisco Systems. Dr. Yang's research interest is in the process of designing products and systems, particularly in the early phases of the design cycle. Her recent work explores various forms of design information in representing the design process and their role in design outcome.

Pierre Leclercq is a Professor in the Department of Architecture, Faculty of Applied Sciences, ULg. He received his PhD in applied sciences in 1994 and MS degree in architecture and building engineering from ULg in 1987. Dr. Leclercq managed many research programs in CAD over 16 years, and he founded LUCID at ULg in 2001. He has led various fundamental and applied programs that all relate to a multidisciplinary approach of design engineering. His primary research interests are design computing and cognition, artificial intelligence in design, human-computer interaction in design, and sketching interfaces.