Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2005), *19*, 277–288. Printed in the USA.

DOI: 10.1017/S0890060405050195

Word graphs in architectural design

BAUKE DE VRIES, JORAN JESSURUN, NICOLE SEGERS, AND HENRI ACHTEN
Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands

(Received October 13, 2004; Accepted May 24, 2005)

Abstract

In computer-aided architectural design, words are an underemployed source of information. Through a series of case studies, we deduced a design annotation data model. All entities in this model can be captured from the design draft, except one: the word relation. Therefore, a system was developed that generates word graphs using single words from the draft as input. The system searches for semantic relations between words and for new intermediate words that can connect two existing words. The system has filters that select only those graphs that are considered interesting by the designers. The envisioned applications of word graphs in the context of computer-aided architectural design are to contribute to the architect's design and to enhance the fluency of the design. These expectations are met, but must be considered in relation to the architect's drafting behavior.

Keywords: Computer-Aided Design; Design Semantics; Word Associations

1. INTRODUCTION

Words have not enjoyed the same amount of attention as the support of graphic representations in research on computer-aided architectural design (CAAD). Research on CAAD has been focused since the introduction of the first drawing systems, mainly on the development of intelligent drawing objects. CAAD is firmly established for the production of the final design, and is now moving toward support of the early design phase, for example, in the area of sketch recognition (Do et al., 2000; Leclercq, 2001). The input of words in CAAD is treated as graphical entity or at most reinterpreted into the character format. In architectural design sketches, annotations are used for clarification of, or commenting on, the design at hand. In our view, if architects write in the act of designing, then words should be considered as part of the design process just like sketches. Words complement the sketch and provide information about the design (Lawson & Loke, 1997). Other research even demonstrated that there is a relationship between the design quality and the use of words (Wong & Kvan, 1999). Jakobsen et al. (1991) propose verb–noun pairs in software design for the formulation of functional requirements. Further, in the field of psychology, it is suggested that showing semantic associations causes people to come up with more semantic and episodic associations (Silberman et al., 2001). Words are a valuable, yet underemployed source of information that can help us better understand the design process and thereby provide better support for the designer.

Based on this observation, we have set out to develop a method to interpret words while designing and process them into a semantic representation. The form in which the use of words has been implemented in semantic representations is by presenting the captured words in graphs to the designer. Relationships between annotated words are inferred, and new words are generated that are associated to these captured words. We have investigated what filtering and presentation of generated graphs is necessary to produce output in a format that can be readily interpreted by the designer. Using a test implementation, we explored the possibilities of word graphs in the context of architectural design. The findings show that it is possible to interpret annotated words and to positively stimulate the architect by structuring these words and by offering new related words.

The outline of the article is as follows. We briefly discuss the outcome of the design case studies. The outcome is organized in a design annotation data model that includes the design entities that make up a design. In the next section, the word graph system and the technologies used in the implementation are described in detail. The feedback filtering section discusses which output of the word graph

Reprint requests to: Bauke de Vries, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. E-mail: B.d.Vries@tue.nl

system is considered interesting and how the filters operate. A list of typical examples demonstrates how word graphs can shift the interpretation of words. Finally, we report on some results of the experiment and discuss the potential of word graphs in the context of CAAD.

2. DESIGN ANNOTATIONS

2.1. Case studies

Although we had indications from our own experience and from other research (Lawson & Loke, 1997; Wong & Kvan, 1999) that words play an important role during the design process, we did not know what this role is, why it is important, and how it fits within the whole design process. Therefore, we organized a short series of case studies, that is, we observed design sessions, introspective design sessions, and student assignments, to analyze the use of words in the early phase of architectural design. The case studies are extensively reported in Segers (2004). Here, we only describe the results that contribute to the understanding of annotated words. From the case studies, we deduced the entities that are used in design: words, sketches, images, and marks.

2.1.1. Words

Words are used for different purposes and in different combinations. Types of writing that were found are annotations, list of items (with numbers or bullets), diagram-like placement of keywords, and words that comprise complete sentences. Annotations often clarify or comment on a sketch, image, or mark. A list of items is used to make a list of attributes to an idea, while a diagram-like placement of keywords is often an abstraction. A statement consisting of complete sentences explains ideas more thoroughly.

2.1.2. Sketches

In the studies we found that there are different types of sketches: a small icon or diagram representation, an isometric or perspective representation, and a projection (facade), section (vertical), or plan (horizontal section) representation. Sketches never seem to be finished. Architects edit them in a later phase or use them as underlying sketch to trace some of the old lines and add new ones through transparent sheets.

2.1.3. Images

Images are pictorial examples that are retrieved from an outside source. Some images are included in the assignment, such as photos from the site or maps, while other images are taken from books, magazines, or the Internet. Images are used to clarify or illustrate an idea, to give inspiration to the architect, or to trace over.

2.1.4. Marks

Marks are any graphic element that are not construed as a word, sketch, or image, but that indicate a relationship between two entities or that single out a particular entity. The type of the mark varies from arrows, lines, and encircling to framings. Marks appear to have a different function, depending on what they connect, and how and where they are placed. The arrow with a single arrowhead, for instance, mostly points at conclusions, solutions, questions, or other important issues. In or near a sketch, this arrow indicates an entrance, a line of sight, or a movement.

Based on the design entities used in the design draft, we construct a data model in the next section to specify how the design entities, including words, are related to each other.

2.2. Word associations in design

In Figure 1, a UML class schema (Fowler & Scott, 2000) shows how design data are structured as classes. For each design entity, a unique time and place stamp attribute is assigned to the entity with each design action, that is, when an entity was created, moved, related, or reused. Extra classes

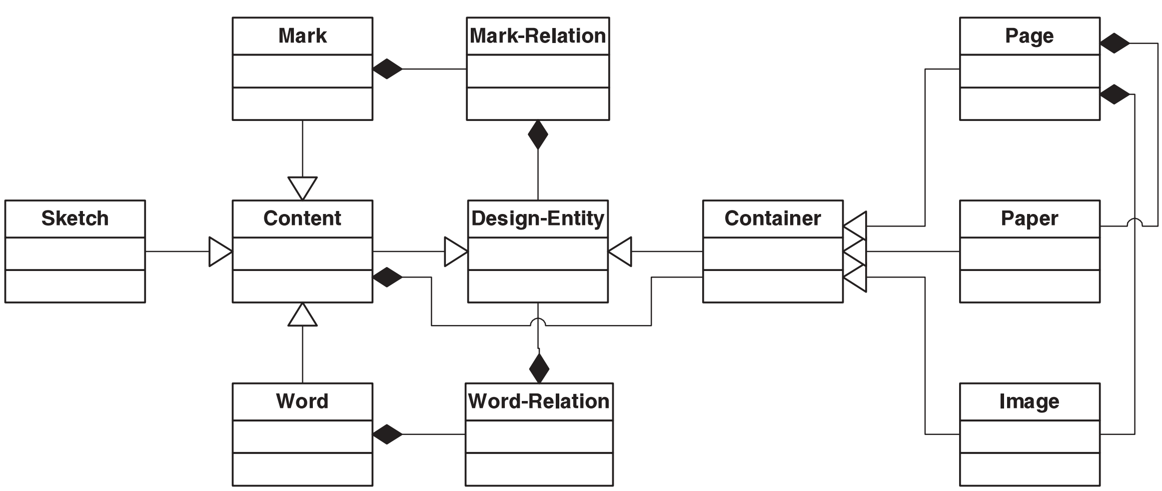


Fig. 1. The design annotation data model.

https://doi.org/10.1017/S0890060405050195 Published online by Cambridge University Press

are added to construct the complete design annotation model: paper, page, content, container, mark relation, and word relation. When an architect uses pen and paper to work, the paper can be a roll of transparent paper, or a workbook with empty pages. Turning to a new page has several functions. A new empty sheet prevents the designer from being distracted by an earlier design draft or provides more space for drafting when continuing the design draft with the same or a new idea. On each page are multiple objects: containers such as papers and images, and contents such as words, marks, and sketches. A container is an abstract class for images, pages and papers, which is a placeholder for (new) content. Content is an abstract class for words, marks, and sketches. Content is generated during a design session, whereas a container is imported into the design. The mark-relation and the word-relation classes represent relationship types. In case of the mark relation, the relationship type can sometimes be deduced from the mark, such as an arrow with the meaning "leads to," but architects do not have a univocal symbol language. Word relations are often not explicitly defined by the architect. Although implicit relationships are inferred from the design by the designer during the design process, they do not become part of the design data because they are not explicitly stated.

In this research project, we have focused on the role of words and we have investigated the possibility to infer relationships between words. More specifically, we have researched lexical associations as a method to stimulate architectural design by associative reasoning. Silberman et al. (2001) describes two types of associations: semantic and episodic. Semantic associations are theoretical: there is a common understanding about the relationship. An example of a semantic association is for instance "mother" with "child." A tree and a branch are related to one another in theory; it is a relation not depending on an individual's perception. Episodic associations are associations that are not theoretical, but exist because something happened in time or space: an episode. An example of an episodic association is "yellow" with "submarine." The relation exists because the Beatles made a song called "yellow submarine."

3. WORD GRAPHS

Episodic associations occur through personal experience, and are therefore difficult to capture, and nearly impossible to infer "after the fact." For this reason, we have focused on the generation of semantic associations to describe the word relations explicitly. We developed a system to test whether the generation of semantic associations can fill in the word-relation class of Figure 1 and thereby contribute to the understanding of the design process.

The word graph test system is implemented as a standalone system that can be used by the designer on demand (Fig. 2). The system reads words that are typed by the user and outputs word graphs. In this procedure, a Lexicon component processes the words and searches for semantic associations. The words and their semantic associations are transferred into a graph structure by the Visualize component. As explained in the following sections, two existing software libraries were used, WordNet and DOT.

3.1. Lexicon

In the case studies, we tested existing lexical systems (e.g., WordNet browser: www.cogsci.princeton.edu/cgi-bin/webwn) to investigate their value-added potential in a design context. We found that the words that the architect writes down during the early phase of the design process do not include much jargon (Achten et al., 2004). Therefore, an existing lexical ontology was used instead of developing a design ontology specifically. We presented the designers with the result from the word graph test system on the basis of the words written down during a design session in which they had participated. During this process of reinterpretation, the designers indicated that they were interested in the ability of these systems to bring forward *new related words*. They found missing the possibility to *structure the words* that were entered in a meaningful way. Therefore, we developed a Lexicon component that generates structures that are intrinsically semantic by exploiting semantic relationships between words that are specified in lexical libraries. The relationships that are subject to our area of research are listed in Table 1.

The structure that emerges from the words entered by the designer can be conceptualized as a *graph* with the words as the nodes and the relations between words that are found by the system as the edges of the graph. The system takes all words written down on one page (i.e., a sheet of paper) into consideration in the graph. Consequently, each time a new word is entered, the system will search for new *graphs*:

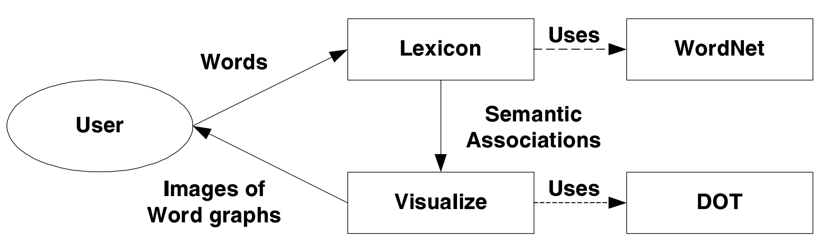


Fig. 2. The word graph test system.

https://doi.org/10.1017/S0890060405050195 Published online by Cambridge University Press

Table 1. *Relationships*

	Lexical Meaning	Implemented in Lexicon
Noun		
Synonym	Equivalent	Yes
Antonym	Exact opposite	Yes
Hyponym	Subordinate: is a kind of . . .	No
Hypernym	Superordinate: is a kind of . . .	Yes
Meronym	Is a part of	No
Holonym	Is a whole of	Yes
Verb		
Synonym	Equivalent	Yes
Antonym	Exact opposite	Yes
Troponym	Particular ways to do this	No
Hypernym	Superordinate: is one way to. . .	Yes
Entailment	Cannot without, . . . entails doing	Yes
Adjective		
Synonym	Equivalent	Yes
Antonym	Exact opposite	Yes
Similar	Not exactly the same	No
Pertainym	Relational adjective	No
Adverb		
Synonym	Equivalent	Yes
Antonym	Exact opposite	Yes

enter the word as a node, *search* for semantically related words, and *use* them as nodes and edges. These graph structures do not necessarily have to form one graph, but they can also consist of disconnected graphs.

The Lexicon component uses the WordNet library (Miller et al., 1990). The basic principle in WordNet is the so-called synset. A synset is a synonym set, a set of words that are interchangeable while maintaining the same meaning. Each sense of a word is in a different synset, and each synset has its own set of semantic pointers, indicating a relation between synsets (word meanings). Because one word mostly has several synsets with their own set of semantic pointers, there are usually many words related to one word.

Next to direct word relations, the Lexicon component also searches for words that connect two other words that are entered into the system. These intermediate words are especially meaningful in the case of hypernym and holonym relationships. To discriminate these relationships from the original ones, we add the superscript index 2 (^{2}). To give an example of a hypernym^2 relation, suppose that the input is "brother" and "sister." The system finds them related not only as antonyms but also with one intermediary word ("relative") as a hypernym^2 relation. The system will show the graph with all three words included, generating the word "relative," because the architect did not enter that word.

3.2. Word graph structure and processing

A word graph contains WordNodes and RelationEdges. A WordNode is a reference to one word in a WordNet synset. A RelationEdge makes a connection between two WordNodes. The relation type can be any type that is defined in WordNet. When a new word is entered into the system, the following happens: the Lexicon component looks up all synsets for the word in WordNet. Every synset is added as a "fresh" word node into the graph. If a new word is entered that is already in the Graph as a generated word, then the generated word is replaced by a fresh nongenerated version. Next, the Lexicon component searches for relations between each fresh word and all other words by pair wise comparison of semantic relations between the words. The system checks for synonym, antonym, hypernym, holonym, and entailment relations. If a relation is found a fresh edge with that relation type is created between those two words. For the hypernym^2 and holonym^2 relation types, the system searches for an intermediate word that has a hypernym or holonym relation with both original words. This intermediate word is added as a fresh generated word and the system searches again for relations between the added word and all other words in the graph as described above, but this time it does not search for hypernym^2 and holonym^2 relations again. When this process ends, the system locates all fresh Graphs with more than one word. These Graphs will be presented to the user.

Hypernyms are the reverse relations of hyponyms. When searching for a hypernym relation between two words, both relation types can be used. However, a word has more hyponyms than hypernyms. It takes less time to find a relation between words when looking at the hypernym relations. The same is true for meronyms and holonyms. In this case it is quicker to search for relations when following the holonym relations.

When checking for a holonym relation between two words, the system also needs to traverse hypernyms. Only direct holonym relations are contained in WordNet; however, a holonym relation can be connected to another hypernym, which therefore is also a holonym of the original word. For example a "door" is part of a "car." To find this relation the system first traverses the hypernym relation between "door" and "car door," followed by the holonym relation between "car" and "door."

The Lexicon component generates graph structures in a textual format. The following section describes the processing of the textual information into a graphical representation.

3.3. Visualization

For a fast and easy understandable representation, we developed a set of symbols that visualizes the (sub)graphs. Words written by the architect are displayed in a white textbox, and words generated by the system are displayed in a yellow text box. Additional information is obtained from the category of the word, that is, a noun, verb, adjective, or adverb. For this purpose, the shape of the word objects is used: a rectangle indicates a noun, an ellipse a verb, and a parallelogram an adverb or adjective. To indicate the type

https://doi.org/10.1017/S0890060405050195 Published online by Cambridge University Press

of relation, the edges between the nodes are used. These are displayed as arrows: a black arrow type indicates whether it is a (1) hypernym or hypernym^2 or a (2) hyponym or hyponym^2. The position of the arrowhead discriminates between relation type 1 and type 2. Color is used to denote less frequently occurring relations: blue stands for entailment, red means antonym, and green means synonym. For the implementation of the visualization component we used the DOT library (Koutsofios & North, 1993). DOT creates hierarchical layouts of directed graphs. A DOT language file specifies the objects in a directed graph and makes an optimal output layout of the objects.

Figure 3 displays the interface of the word graph test system. At the bottom part is a field where words are entered that should be included in the word graph. The left column shows the words that have been entered. In the main window, the subgraphs are subsequently added each time a new word generates one or more new subgraphs. The horizontal scrollbar allows for browsing through the history of subgraphs. Pushing the Complete Graph button displays a sheet with all subgraphs that are generated from the complete list of words. The filters visible in the bottom left can be checked on or off. These are discussed in the next section.

4. FEEDBACK FILTERING

When words are added, the process described in the previous section generates all possible semantic relations. The displayed graphs become very large, which makes them too complicated to understand quickly. Some of these relations are not interesting (as stated by the interviewed architects in Section 3.1) or can easily be deduced from other relations. Therefore, we implemented filters that reduce the size of the graph and that present only interesting relations and new words. We can distinguish between two classes of filters: redundancy filters and relevance filters (Table 2). The redundancy filters called "ignore transitive hypernyms," "ignore transitive holonyms," and "ignore secondary synonym" remove relations that can be deduced from other relations. The relevance filters "remove generated distracters" and "ignore same syntax words" remove relations and words that are not interesting for the user.

5. WORD GRAPHS IN CAAD

To test our system in an architectural design environment, we integrated the word graph system in a design aid system. In the following sections, the technical aspects are described including the user interface.

5.1. Platform

The Visual Interaction Platform (VIP-3; Aliakseyeu, 2003), is an augmented reality system developed for architects. VIP-3 preserves the naturalness of the traditional way of designing with pen and paper while at the same time adding new functionality. The system consists of a table with various projectors. On the table are a digital drawing board and an electronic pen. The digital drawing board is a Wacom tablet, which is calibrated with the overhead projector. Strokes can be made with the electronic pen and are then projected on the drawing board immediately; as a consequence, it feels like working with a normal workbook. The resolution is high enough for the architect to work with rather fine lines and to retain a sketchlike quality. The pen can also be used to manipulate items. The system offers transparency of the virtual papers, which can also be rotated, scaled, and edited. Working with VIP-3, the architect sketches as usual, writes things down, makes marks, and performs searches in an image database. Feedback from the word graph system is shown in real time on a separate vertical screen.

5.2. Word recognition and word graph presentation

For the generation of word graphs, the text that is written down during the design session must be captured by the system and recognized as words. Although handwriting recognition software has improved rapidly over the last years, it fails for our purpose for the following reasons:

- The quality of text in design drafts is often very poor.
- Not just a specific set, but all words need to be recognized.
- Text in design drafts is written under various angles.
- Text must be isolated from other strokes such as marks and sketches.

No sufficiently robust software was found available. A pragmatic solution to this problem of word recognition is the Human Handwriting Recognition Server module. This module basically functions as a "Wizard of Oz" by a human observer who recognizes the words that are made in the design draft. The observer is located in a different area from the design system. All strokes appear on the observer's monitor. The observer selects the strokes that are part of a word, as displayed in Figure 4, and types in the word for the CAAD system.

As soon as a new word is entered, new word graphs (if any) are generated and displayed to the system user on the vertical screen. Word graphs are displayed on the vertical screen in a striplike manner. An average of five word graphs is visible on the vertical screen to prevent displaying too much information. New word graphs appear on the right and the old ones "slide" to the left. With the word graph menu, the architect can see all word graphs by scrolling back and forth through the strip of word graphs. One word graph from the strip is highlighted, indicating the selected word graph. With the button Get Graph the selected word graph is copied to the horizontal work field to become part of the design draft (Fig. 5).

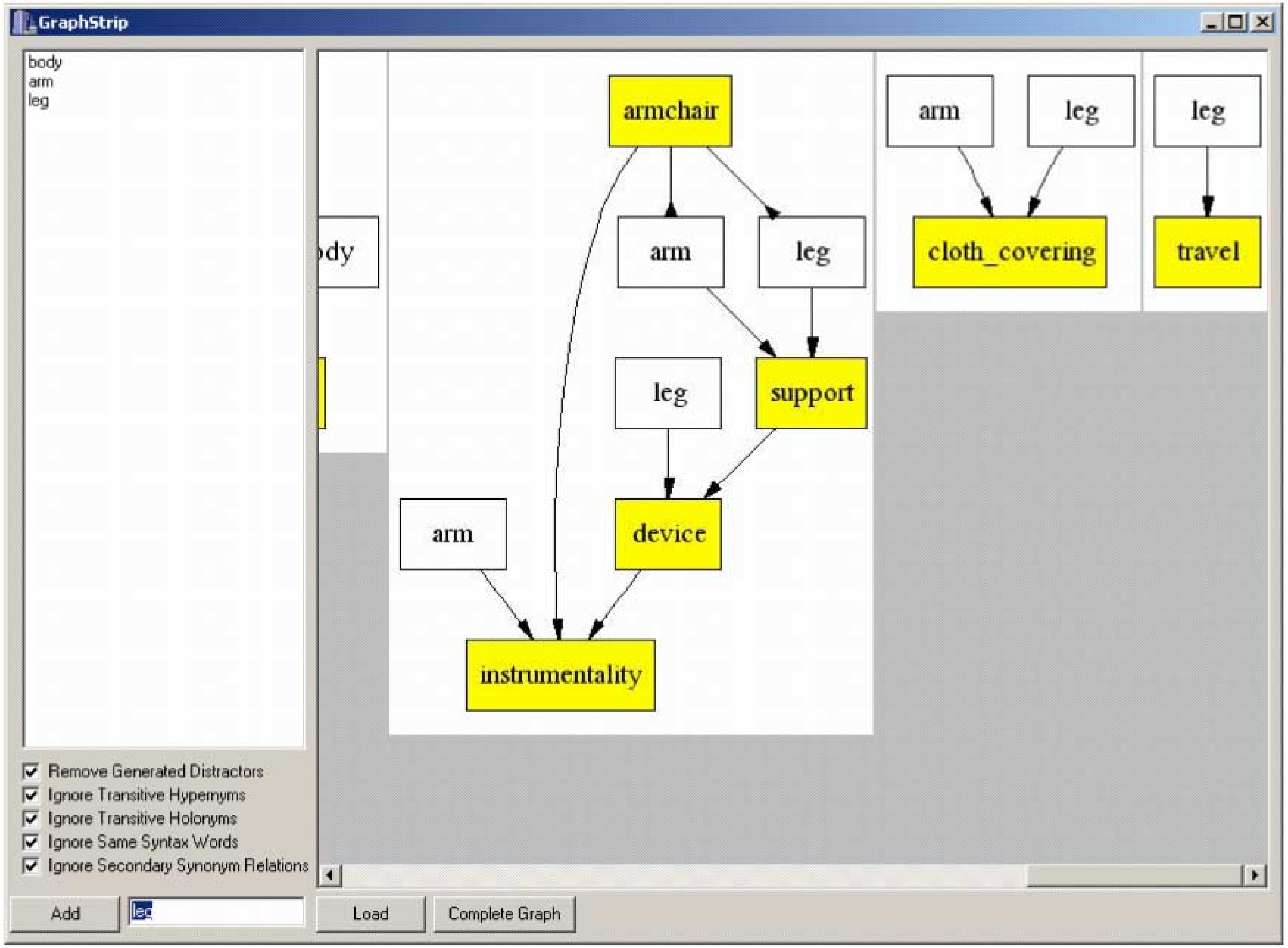
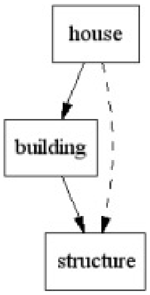
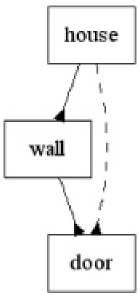
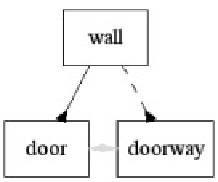
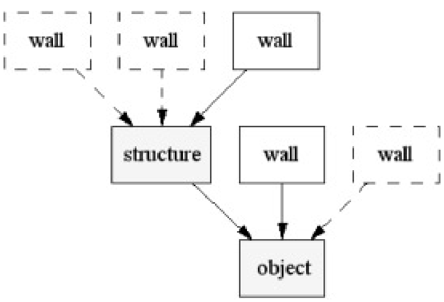
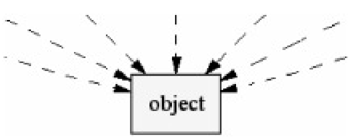


Fig. 3. The word graph test system interface. [A color version of this figure can be viewed online at www.journals.cambridge.org]

Table 2. *Feedback examples*

	*Ignore transitive hypernyms* This filter looks for transitive hypernym relations between the words and removes the longer routes.
	*Ignore transitive holonyms* This filter looks for transitive hypernym relations between words and removes the longer routes.
	*Ignore secondary synonym* All synonyms have the same relations with other words, so only the relations of one synonym to the other words need to be displayed. The other relations are made invisible.
	*Ignore same syntax words* All synsets of a word are added to the graph. Even when these synsets have different meanings, they often have the same relation to other words. This filter looks up words that are written the same and have exactly the same relations in the current graph. These duplicates are removed, however, to the loss of information that there were different meanings of that word.
	*Remove generated distractors* This filter looks for generated words with more than seven relations and removes them from the diagram. This number is proved in the case studies to be a practical value when trying to maintain the overview of the graph. The system does not exclude words that the architect writes down, even if there are more than seven relations with that word.

In the snapshot from the VIP desktop of Figure 5, we see in the top left corner a vertical slider bar with images, in the lower left corner a drawing tool menu, some sketches and writings in the working area, and an included word graph in the top right corner.

The word graph in the design draft was selected by the architect from all generated word graphs on the vertical screen. The words written by the architect are: "eye," "hair," and "light." Using these words the system generates (among others) the word graph as shown in Figure 6. Next to the various relationships presented by arrows, the following intermediate words have been generated: mammal, body-part, and process. As can be seen in Figure 5, the architect encircled the word mammal, and it inspired her to make some sketches that look like animals.

6. EXPERIMENTAL RESULTS

In the previous sections we explained how word graphs can be generated and presented. The aim of the research is to

https://doi.org/10.1017/S0890060405050195 Published online by Cambridge University Press

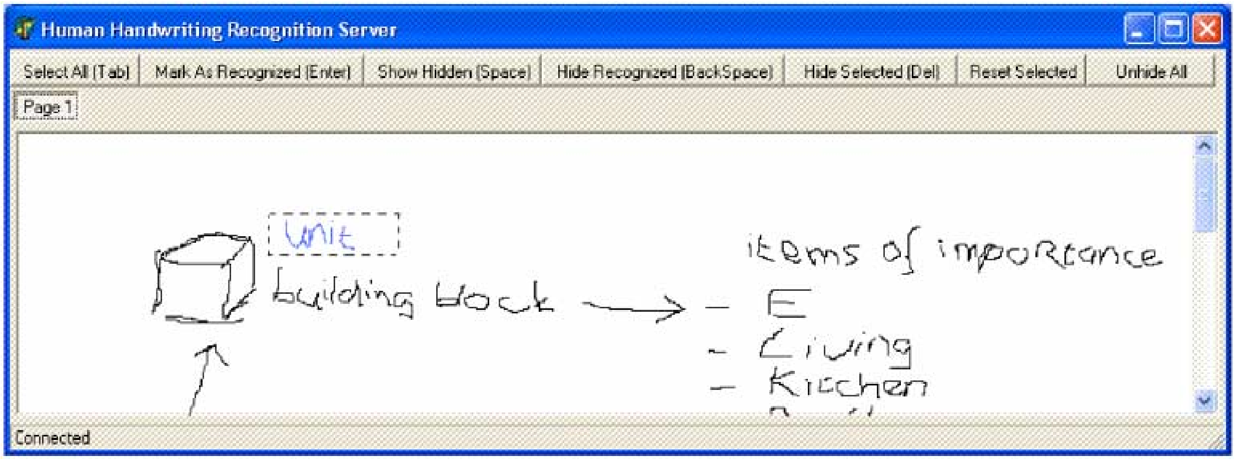


Fig. 4. The Human Handwriting Recognition Server module. [A color version of this figure can be viewed online at www.journals.cambridge.org]

use the underemployed potential of words in design in the context of CAAD. In this section, an experiment is reported that researched two aspects of word graphs in CAAD: graph and word usage and periods of (no) activity. For the experiment, the prototype system described above was used. The experiment is a two-condition (with and without feedback in word graphs) repeated-measures design with two independent variables (task and order of the tasks) with the use of

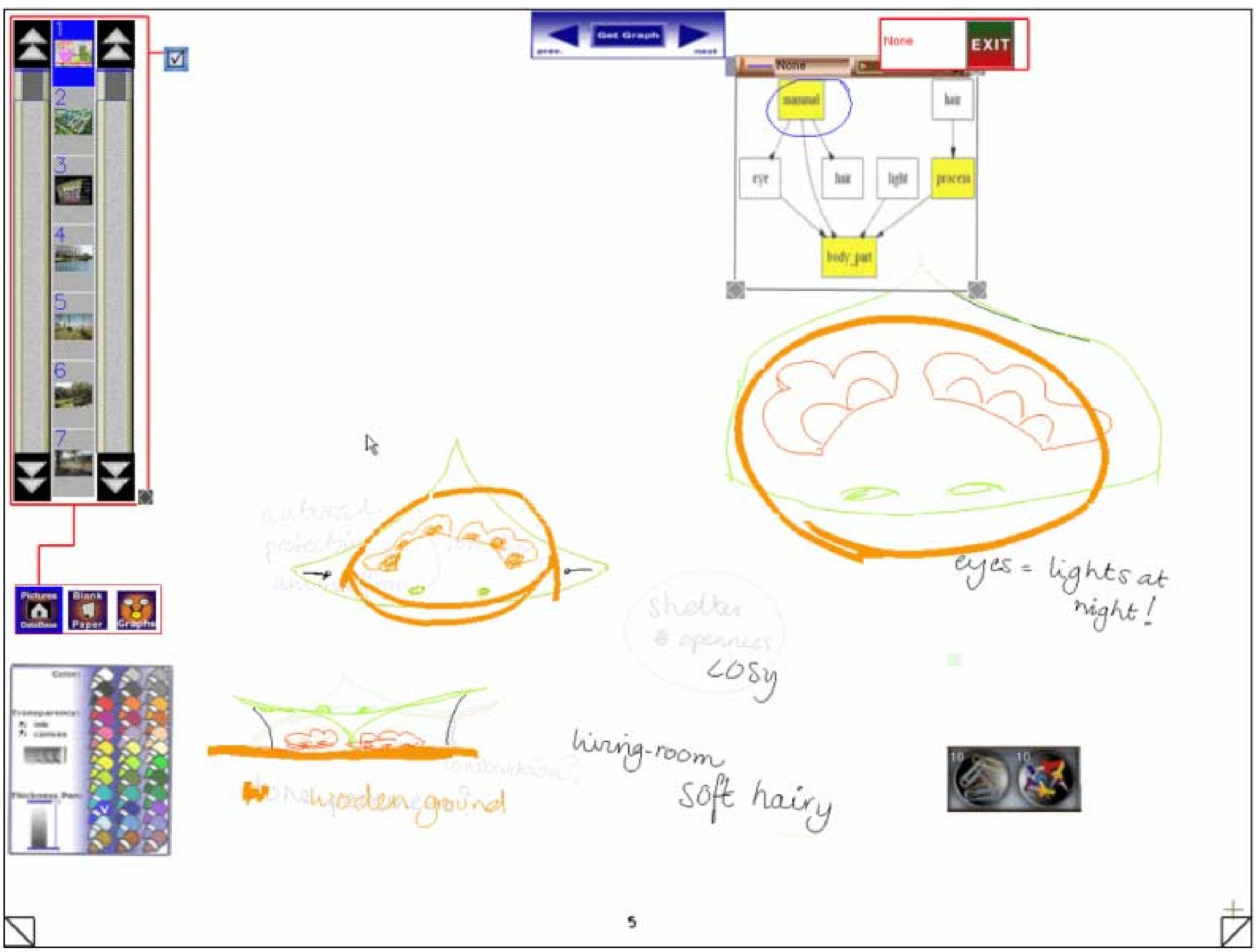


Fig. 5. A snapshot from the Visual Interaction Platform design desktop. [A color version of this figure can be viewed online at www.journals.cambridge.org]

https://doi.org/10.1017/S0890060405050195 Published online by Cambridge University Press

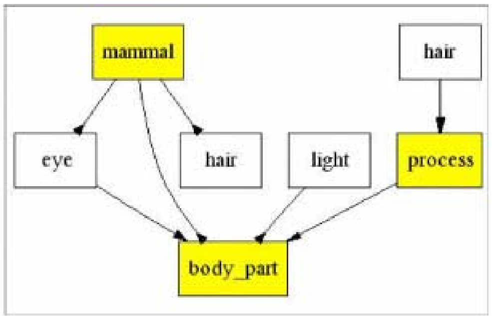


Fig. 6. The word graph included in the design draft. [A color version of this figure can be viewed online at www.journals.cambridge.org]

randomized blocks to avoid the effects caused by standard order. A tutorial with a small task tackled the learning effect of working with the system. Eighteen professional architects participated in this experiment. The prototype application provides an environment that is equivalent for all subjects except for the feedback in word graphs. Consequently, the difference that is measured is the effect of the feedback. Further details about the prototype setup and experimental design have been published by Segers (2004, pp. 67–81, 83–91).

6.1. Means of assessment

For the assessment, two data logs are generated by the system during a design session, via a word graphs log and an activity log. The prototype keeps track of the word graphs, that is, word input, generation, acceptance, and constituent words and relations. Architects are requested to accept a word graph during the design process when they find a word graph interesting, or when they think that it somehow contributes to their design. The prototype also keeps track of the pen activity on the tablet. Pen activity consists of moving only a short distance above the tablet, making strokes, and inserting images or word graphs. If the pressure of the pen tip drops below a certain threshold, a record entry is made indicating inactivity. Similarly, if the pressure increases above the threshold, activity is recorded.

6.2. Graph and word usage

A measure of applicability of the presented word graphs is the number of times that word graphs were accepted by the architect. A measure of applicability of the newly generated and presented words is the number of written words by the architect that had been generated by the system earlier.

Figure 7 shows for each architect the numbers of acceptances of word graphs in relation to the number of times an architect wrote a word that was generated by the system. From the figure, there is evidence that both graph usage and word usage occur, which indicates the applicability of newly generated words and the applicability of word graphs. The spread of the data in Figure 7 indicates that architects appreciated the feedback differently. It also shows that three archi-

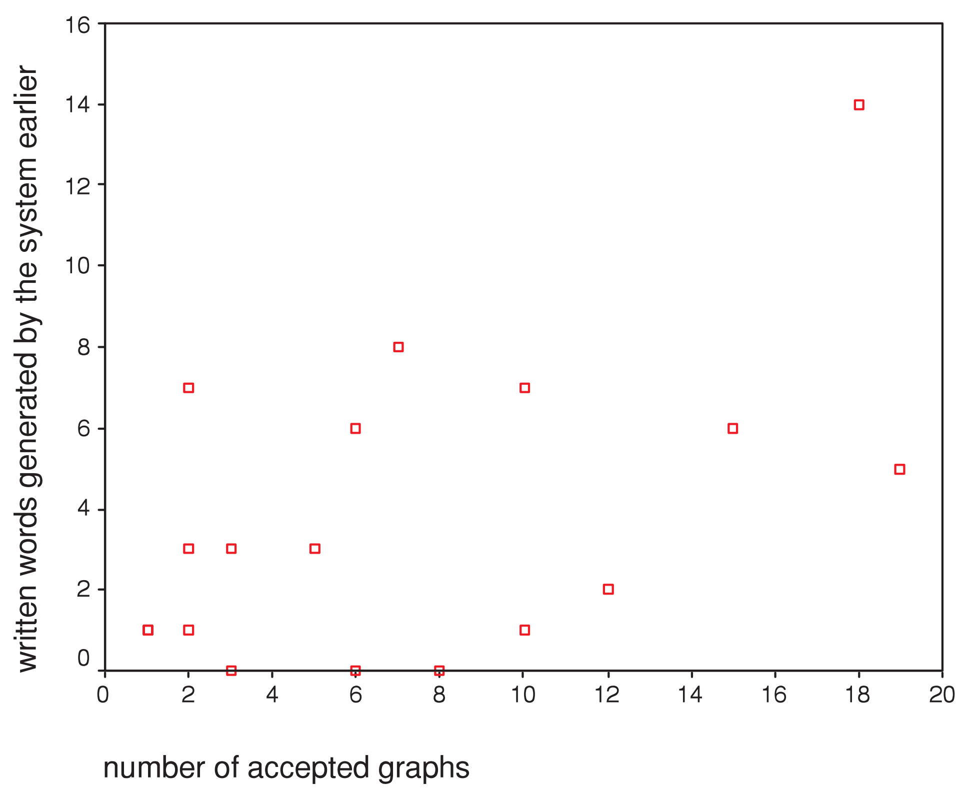


Fig. 7. Acceptance of word graphs versus the architect's writing a word generated earlier by the system. [A color version of this figure can be viewed online at www.journals.cambridge.org]

https://doi.org/10.1017/S0890060405050195 Published online by Cambridge University Press

Table 3. *Standard linear regression between effect (Y) and the number of written words (X)*

Effect $Y = aX$	a	Sig. a
Number of associations in *accepted* word graphs	0.422	0.088
Percentage of the number of associations in *accepted* words graphs from the total number of associations in generated words	−0.016	0.745

tects did not write down any word that was generated by the system earlier.

However, analysis of the design draft revealed that architects did write down several of the earlier generated words although they did not accept the word graphs that contained those words. This indicates that one word was interesting in those word graphs, yet these word graphs as a whole were not interesting enough to include them in the design. That is, single generated words can also be interesting apart from the word graph.

To know whether an architect's tendency to write many words means he will find more utility in the system, we analyzed the relationship between the number of written words generated by the system earlier and the total number of written words by the architect. From this, it appears that the architect does not write down more computationally generated words when he writes more overall. This indicates that the architect's act of writing down more words did not influence the number of written words that were generated by the system earlier. The conclusion here is that the architect does not necessarily have to write more words to find more use in the feedback in generated words. Further, if an architect finds the word graphs useful, then the architect may be more likely to find more use in the generated words as well. We must also consider the possibility that at a given moment, there is a limit to the novelty of the prototype's newly generated words.

Table 3 shows that a significant relationship exists between the number of associations in accepted word graphs (Y) and the number of written words (X). In other words, if the architect writes more, the word graphs that he accepts are also more complex. An analysis of the relationship between the percentage of the number of associations in the accepted word graphs from the total number of associations in generated word graphs (Y) and the number of written words (X) indicates that there is no dependency. More word graphs do not lead to a higher acceptance of them.

6.3. Periods of (no) activity

If the architect accepts a word graph after a period of no activity, it points at the usability of a word graph, in the sense that architects might accept a word graph to stop the fixation and continue the design. We wanted to investigate if design activity increases or decreases when the architect is given feedback in word graphs. A state of inactivity is deduced from the activity log. When the architect does not move the pen above or on the tablet for 20 s or more, the architect is considered to be in an inactive state. The duration of activity and no activity is calculated (minutes) for the situations of working with and without the prototype system (Table 4). In the last row of the table, the fractional inactivity time is calculated from the duration of the assignment. The design fluency is dependent on the fractional inactivity time, which is a relative measure of inactivity to activity.

It is found significant at a 5% probability level that giving feedback through the word graphs has an effect on the design fluency (Table 4). Using the prototype system, the architects have longer periods of both activity and of no activity. Even when the percentage of no activity and the total duration of the session are considered, the results are negative. The design fluency is not enhanced. Looking at the presented word graphs requires time. This action may contribute to longer periods of no activity. From the activity log, we can also observe longer periods of activity in those cases where architects did not accept word graphs.

Table 5 provides the numbers of the accepted words graph after a period of inactivity, the total number of word graphs, and the percentage of acceptance. The last column shows that most of the architects experience substantial support from the prototype system in overcoming periods of inac-

Table 4. *Comparison of the means of the duration of the design activity and no activity (paired-samples t test)*

	With Prototype		Without Prototype			
Effect	M	SD	M	SD	ΔM	Significance
Duration no activity	7.50	2.48	5.17	3.84	−2.33**	0.016**
Duration activity	51.61	4.88	47.94	5.82	−3.67**	0.050**
Percentage no activity/duration total session	12.6	4.0	9.1	5.4	−3.5**	0.010**

$^{**}p \leq 0.05$ (two tailed).

Table 5. *Accepted word graphs after a period of no activity*

Architect	No. of Accepted Words After Inactivity	Graphs	Acceptance After Inactivity (%)
1	5	15	33
2	2	7	29
3	3	10	30
4	1	5	20
5	1	1	100
6	0	2	0
7	1	2	50
8	9	12	75
9	2	6	33
10	1	6	17
11	1	10	10
12	0	3	0
13	1	3	33
14	1	2	50
15	0	1	0
16	6	19	32
17	4	8	50
18	1	18	6

tivity. An additional regression analysis confirms that the number of word graphs accepted after a period of inactivity is significantly dependent on the number of word graphs that were accepted (significance = 0.005). We can conclude that the word graphs positively affect architects who are more "into words," that is, architects who accept generated words.

7. DISCUSSION AND CONCLUSION

We presented an approach to utilize words that are written down in the design process. We showed how lexical relations can be used to construct relations among these words, and to generate new intermediate words. This work was implemented in a prototype design aid system that visually presents the generated graphs to the user. From experimental data we can conclude that designers actually use the word graphs by generating associations and words during their design process. Furthermore, we found that the word graphs contribute to overcoming periods of inactivity for most architects. It must be kept in mind, however, that the use of representations differs among architects; some of the architects were not accustomed to writing in their design draft. These architects will have less benefit from word graphs and newly generated words. Points of improvement according to the architects are a reduction of the amount of words, a reduction in the abstraction of words, an increase in the type of associations, and an increase in interaction possibilities with the system.

In understanding the full potential of using word graphs, and in further pursuing research on words and associations in CAAD, it is useful to consider the architects' remarks. Architects appreciated being helped with structuring ideas and making ideas abstract, expressing and explaining ideas in words, and changing the direction of thinking. One architect stated, "It could help you to think different, which was especially useful when you're stuck in the middle of the design process." Another architect pointed out, "When I don't know what to do anymore for a while, I just write down some words and who knows what comes of it? It is an extra possibility for me next to watching the images to get inspiration. I had a more constant flow of ideas. It is pleasant to work with the feeling more is coming out of me."

These findings lead us to believe that words graphs and newly generated words are an important means for increasing the design fluency of the user. As a whole, the results of both case studies and the experiment lead to the conclusion that words, word associations, and the use of word graphs with newly generated words are valuable design content in the context of CAAD.

REFERENCES

Achten, H.H., van Leeuwen, J.P., & Fridqvist, S. (2004). Communicating concepts for shared understanding—a multi-agent approach. In *AVOCAAD 2003—Proc. Fourth Int. Conf. Added Value of Computer Aided Architectural Design* (Stellingwerff, M. & Verbeke, J., Eds.), pp. 195–208. Delft: Delft University Press.

Aliakseyeu, A. (2003). *A computer support tool for the early stages of architectural design*. PhD Thesis. Eindhoven: Eindhoven University of Technology.

Do, E.Y., Gross, M., Neiman, B., & Zimring, G. (2000). Intentions in and relations among design drawings. *Design Studies 21(5)*, 483–503.

Fowler, M., & Scott, K. (2000). *UML Distilled, A Brief Guide to the Standard Object Modelling Language*. Amsterdam: Addison–Wesley.

Jakobsen, K., Sigurjonsson, J., & Jakobsen, O. (1991). Formalized specification of functional requirements. *Design Studies 12(4)*, 221–224.

Koutsofios, E., & North, S.C. (1993). *Drawing graphs with DOT* [Technical Report]. Murray Hill, NJ: AT&T Bell Laboratories.

Lawson, B., & Loke, S.M. (1997). Computers, words and pictures. *Design Studies 18(2)*, 171–183.

Leclercq, P. (2001). Programming and assisted sketching. In *CAADFutures 2001* (de Vries, B., van Leeuwen, J., & Achten, H., Eds.), pp. 15–31. Dordrecht: Kluwer.

Miller, G.A., Beckwith, R., & Fellbaum, C. (1990). Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography 3(4)*, 235–244.

Segers, N.M. (2004). *Computational representations of words in architectural design: development of a system supporting creative design*. PhD Thesis. Eindhoven: Eindhoven University of Technology.

Silberman, Y., Miikkulainen, R., & Bentin, S. (2001). Semantic effect on episodic associations. In *Proc. 23rd Annual Conf. Cognitive Science Society* (Moore, J.D., & Stenning, K., Eds.), Edinburgh: Cognitive Science Society.

Wong, W., & Kvan, T. (1999). Textual support of collaborative design. In *ACADIA '99* (Ataman, O., & Burmudez, J., Eds.), pp. 168–176. Salt Lake City, UT: Association for Computer-Aided Design in Architecture.

Bauke de Vries is Chair of the Design Systems Group of the Faculty of Building and Architecture at Eindhoven University of Technology. His main research topics are CAAD, product and process modeling, virtual reality technology and interfaces, and knowledge-based systems. Dr. de Vries

https://doi.org/10.1017/S0890060405050195 Published online by Cambridge University Press

is the author of publications that have been presented at international conferences and scientific articles that have been published in international journals. He has been a member of many PhD committees and Project Leader in European research projects.

Joran Jessurun is a Software Engineer who supports other researchers with the development of prototype software systems. The application domain is primarily in design support systems. His expertise areas are software development (coding), software specification and design, computer graphics, virtual reality, databases, and computer networks.

Nicole Segers' interest in art and science led to an MS in architectural design and a PhD in CAAD at the Technische Universiteit Eindhoven. Her work focuses on the early stages of the design process and especially on creativity and the use of (written) words by architects. This interest is developed further at De Twee Snoeken, which is a Dutch firm that combines an architectural office with software development. Dr. Segers is currently working on the participation of inhabitants in architectural design and urban planning. The use of words and semantic relations plays an important role in this project as well.

Henri Achten is Assistant Professor of design theory research and CAAD. His research work focuses on the architectural knowledge content of graphic representations, ways of formalizing this knowledge, and development of automated recognition and design support tools in this area. Dr. Achten's second area of research is the connection between design methodology, design theory, and CAAD, in particular, how the use of computation informs and changes design processes and design thinking.