Generative, Adaptive Human-Computer Interaction by Conceptual Analogy

Katy Börner

University of Bielefeld, Faculty of Technology
PF 10 01 31, 33501 Bielefeld, Germany
katy@TechFak.Uni-Bielefeld.de

Abstract

This paper describes research on *generative* and *adaptive* human-computer interaction (HCI). Generative interaction refers to the generation of useful knowledge structures (e.g., a config- uration or a plan) to satisfy a user's goals. Adaptive interaction entails the personalization of *content* and *presentation* of it (Langley 1997). Content involves the knowledge structures and inferences used to support human problem solving. The presentation of content refers to the implemented human-computer interface. The research combines Artificial Intelligence (AI) techniques and Virtual Reality (VR) technology. In particular the approach of *Concep- tual Analogy* (Börner 1997) is applied to provide generative and adaptive support for human problem solving in an assistant-like fashion. VR technology enables the computer interface to be adapted to the perceptual and effector systems of human users.

1   Introduction

By *intelligent HCI* we mean *generative, adaptive* human-computer interaction to support tasks like navigation or manipulation. Generative interaction refers to the generation of useful knowl- edge structures (e.g., a configuration or a plan) to satisfy a user's goals. Adaptive interaction entails the personalization of *content* and *presentation* of it (Langley 1997). Content involves the knowledge structures and inferences used to support human problem solving. The presentation of content refers to the implemented human-computer interface.

Note that *adaptive* refers to the *static* adaptation of the input/output devices used for human- computer interaction (HCI) as well as to the *dynamic* update of the software used to support human users. Virtual Reality (VR) technology can be applied advantageously to adapt computer interfaces to the perceptual and effector systems of human users. In order to achieve adaptive user support approaches developed in Artificial Intelligence (AI) can be exploited.

This paper proposes a system to be built which uses a VR interface for HCI and successively adapts its knowledge structures and reasoning as well as the presentation of support to user preferences.

The organization of the paper is as follows. Section 2 starts out by introducing some major problems of todays HCI to achieve generative, adaptive interaction. Section 3 introduces an abstract application domain that will be used to illustrate the adaptive support of navigation and manipulation tasks. In Section 4 we give an overview about *Conceptual Analogy*, a *knowledge lean, efficient* approach that may be applied advantageously to achieve adaptive HCI based on past user traces. The approach will be applied to support navigation and manipulation tasks

by suggesting preferred paths and subassemblies. Adaptive interaction via a Virtual Reality interface is exemplified in Section 5. The paper concludes with a discussion of this research.

2   The Problem of Generative, Adaptive HCI

Todays WIMP graphical user interfaces are designed for a single user who controls objects that have no autonomy and at most react to mouse manipulations. Interaction uses only one visual channel at a time, is half duplex, and involves a sequence of mouse events in the most complex scenario. Post-WIMP interfaces are under development and test since the early 1990s but are not yet in wide circulation (van Dam 1997). They support human problem solving in a variety of tasks by "delegating" these tasks to the system. In contrast to previous generations of interfaces, post-WIMP interfaces applying VR technology integrate a number of modes (e.g., speech, hearing, touch, gesture, facial expression, bio-feedback) and media (e.g., graphic, animation), thereby involving several senses in parallel. Multiple users can work on a shared task via systems providing viewer-centered perspective and wide-field-of-view stereo immersive displays controlled via head and hand tracking. Speech and gesture recognition devices can exchange continuous signals for operand and operation specification in full-duplex.

There exist a number of VR systems like *Sculptor* (Kurmann 1995) or *Sketch* (Zeleznik, Herndon & Hughes 1996) that allow for very intuitive and direct virtual interaction within a 3D world. However, these systems provide only rudimentary support on navigating and manipulating virtual worlds. On the other hand, there are examples of generative and adaptive AI systems like *Clavier* (Hennessy & Hinkle 1992), that support human users, e.g., in generating autoclave loads. However, these systems still use WIMP GUI's for human-computer interaction. To our knowledge there exists no system that provides generative, adaptive support, employs VR technology, and is able to adapt the support and its presentation to a user's personal style.

3   Application Domain and Task

To motivate, illustrate, and evaluate our approach to generative, adaptive human-computer interaction we will use an abstract labyrinth-like world. This rectangular, equally spaced world is enriched with virtual objects. The labyrinth-like world can be *navigated* in three dimensions and the objects can be *manipulated* (selected and assembled). User paths as well as object assemblies are represented by sets of places/objects plus the temporal/spatial relations among them (e.g., 'place_1 before place_2' or 'object_1 below object_2').

The general task is to build certain object assemblies at different locations within a restricted time. Therefore, the user has to navigate to particular places and manipulate objects correctly. The abstract domain and task may be instantiated into concrete domains and tasks such as architectural design, robot navigation, manipulation and navigation of potentially huge information spaces, or production planning.

Examples: A simple navigation task is depicted in Figure 1, left. Reachable places (e.g., the places labeled A to D) correspond to the intersecting points of a fixed grid and can be uniquely identified by their x/y coordinates. Steps connect places horizontally or vertically. Despite the simplicity of this example there exists a variety of paths connecting, e.g., the places labeled C and B (see Fig. 2). A navigation problem consists of arbitrary places and path pieces connecting these places partially. There is no map or optimization criteria (like find the shortest path)
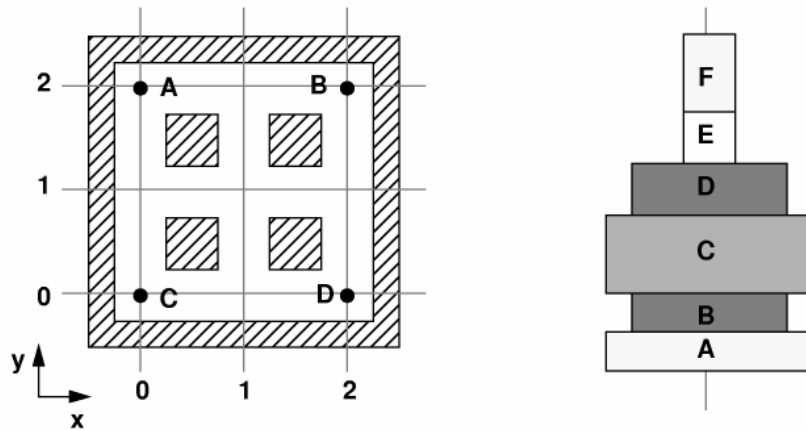
Figure 1: Simple navigation task (left) and manipulation task (right)

available. Figure 1, right shows an simple sample manipulation task. Again, there exist several possibilities to built, e.g., a toy tower using the stones labeled A to F. Both tasks are given in 2D but extend to 3D naturally.

Navigation examples can be represented by a set of vertices and a set of edges representing places and the *before* relation among these places. For example the path $\mathcal{P}$ from starting point C at position 0/0 to end point B at position 2/2 labeled with (I) in Fig. 2 may be represented by:

$$\mathcal{P}_{(I)} = (\{0/0, 1/0, 2/0, 2/1, 2/2\}, \{(0/0, 1/0), (1/0, 2/0), (2/0, 2/1), (2/1, 2/2)\}).$$

Manipulation examples can be represented by a set of objects (represented by their type - in the example A to F) and the *below* relation among them. Thus the toy tower assembly $\mathcal{A}$ in Fig. 1, right may be represented by:

$$\mathcal{A} = (\{A, B, C, D, E, F\}, \{(A, B), (B, C), (C, D), (D, E), (E, F)\}).$$

By introducing other relations like *beside*, *in* etc. more complex object assemblies can be represented.

Given a large number of examples (also called cases) a *case-based approach* may be applied. A new navigation or manipulation problem is solved by selecting similar past cases and transferring (if needed adapting) their solution to generate an appropriate solution. For detailed surveys of case-based reasoning see (Kolodner 1993, Aamodt & Plaza 1994).

However, to support navigation and manipulation tasks not the position or type of single places/objects but their temporal/spatial relations are important and have to be considered during retrieval and adaptation. Cases need to be represented by graphs and a structural similarity measure working over case sets has to be defined. We are going to use the following graph-theoretic definitions.

A *graph* $g = (V^g, E^g)$ is an ordered pair of vertices V^g and edges E^g with $E^g \subseteq V^g \times V^g$.

A *set of graphs* is denoted by G.

The *entire graph* $g^{(G)}$ of a set G of graphs equals the union of the vertices/edges of the graphs in G, i.e., $g^{(G)} = (\cup_{i=1}^{|G|} V^{g_i}, \cup_{i=1}^{|G|} E^{g_i}) = (V^{(G)}, E^{(G)})$.
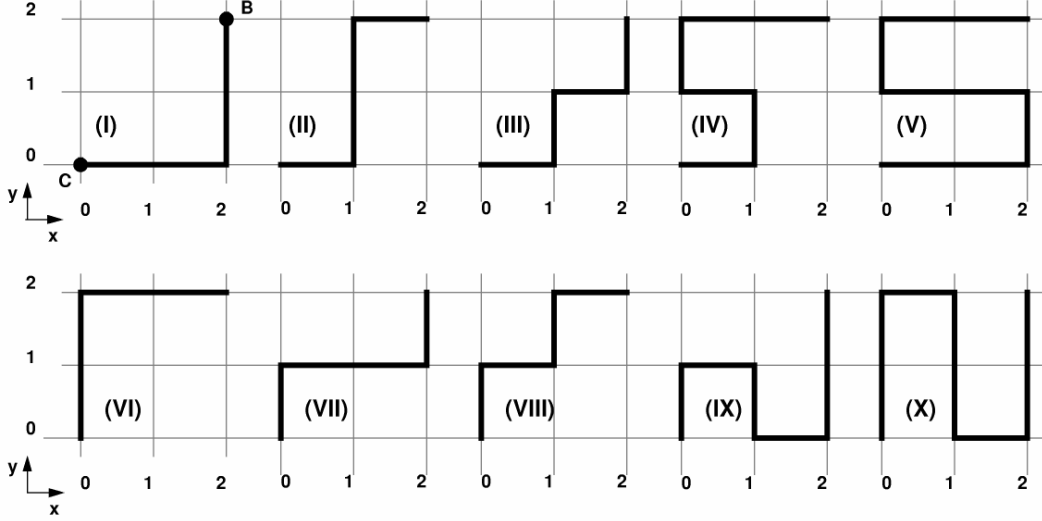
Figure 2: Example paths connecting places labeled C and B

Based on this we can define a case, case-base as well as a problem and its solution.

A structurally represented *case* $c = (V^c, E^c)$ is a graph.

A *case base* CB is a finite set of cases.

A *problem* provides a set of vertices and perhaps some edges; i.e., it is a forest.

A *solution* of a problem contains the problem vertices and edges and adds those vertices and edges from structurally similar cases that are needed to connect all problem vertices and edges.

The *structural similarity*[1] σ maps a set G of graphs into the interval $[0, 1]$:

$$\sigma(G) := \frac{\sum_{i=1}^{|G|} |E_i^{(G)}| * \frac{i}{|G|}}{|E^{(G)}|} \in [0, 1],$$

were $E_i^{(G)}$, $i = 1, ..., |G|$ is defined as $E_i^{(G)} = \{(v_l, v_k) \mid (v_l, v_k) \in E^{(G)} \ \wedge \ P_G((v_l, v_k)) = \frac{i}{|G|}\}$.

A *case class* CC is a non-empty subset of CB that groups cases of high structural similarity.

Additionally, we are dealing with a weak theory domain. Hardly any information about the relevance of features guiding the selection of similar cases is available. The adaptation of prior paths/arrangements mainly corresponds to adding, eliminating, or substituting places/objects and their relations. Because of the variety and the possible number of combinations of these modifications, adaptation knowledge is difficult to acquire by hand. Assuming that *structurally similar* cases satisfy similar constraints, a new problem may be solved by transferring the vertices and edges of the most similar *case class*. That way, constraints on navigation or manipulation tasks are transferred implicitly. On the assumption that relations which occurred often in the cases of a case class lead to solutions of higher *quality*, edges of high *relative frequency* are preferred to generate complete problem solutions.

---

[1]Note that the structural similarity function is commutative and associative. Thus it may be applied to a pair of cases as well as to a set of cases.

The *relative frequency* P_G of an edge (v_i, v_j) of the entire graph of a set G of graphs equals the cardinality of a set of graphs containing this edge divided by the number of graphs in G:

$$P_G((v_i, v_j)) := \frac{|\{g \in G \mid (v_i, v_j) \in E^g\}|}{|G|}.$$

The *relative frequency* P_G of a set of edges E relative to a set of graphs G equals:

$$P_G(E) := \frac{1}{|E|} \sum_{(v_i, v_j) \in E} P_G((v_i, v_j)).$$

The *quality* of a solution equals the relative frequency of its edges with regard to the cases used to generate it.

Taken together, past cases may be combined to support navigation as well as manipulation tasks. During reasoning, the temporal and spatial structure (which does not reflect the function or behavior of objects) inherent in such paths and object arrangements as well as its relative frequency needs to be considered. Complex case representations, however, cause increased computational expense to retrieve, match, and adapt cases. To guarantee answer times that are realistic for real world applications, efficient memory organizations directly tailored to the applied reasoning mechanisms are essential. Problems that relate to the amount and the structural complexity of the knowledge have to be addressed.

4 The Approach of Conceptual Analogy

Conceptual analogy (CA) is a general approach that relies on conceptual clustering to facilitate the efficient selection and combination of complex cases in analogous situations (Börner 1997).[2] CA divides the overall task into *memory organization* and *analogical reasoning*. Both subtasks process graph representations but are grounded on attribute-value representation of past cases. They are explained in detail subsequently.

Originally, the approach was designed to support the design of complex installation infrastructures for industrial buildings. In this domain cases correspond to pipe systems that connect a given set of outlets to the main access. Pipe systems for fresh and return air, electrical circuits, computer networks, phone cables, etc. are numerous and show varied topological structures.

The approach has been fully implemented in SYN, a module of a highly interactive, adaptive design assistant system. The implemented system interacts with users via the manipulation of CAD layouts describing pipe systems of real buildings. It supports architects in the design of new buildings and adapts its support to different needs based on past user interactions. See (Börner 1995, Börner 1997) for a more detailed description of the implementation.

4.1 Memory Organization

Memory organization requires a case base $CB = \{c_1, .., c_N\}$ providing a significant number of structurally represented cases as well as a structural similarity function σ. Nearest-neighbor-based, agglomerative, unsupervised conceptual clustering is applied to create a hierarchy of case classes grouping cases of similar structure.

---

[2] The research was strongly inspired by work of *Bipin Indurkhya* and *Douglas Hofstadter* to model human analogy-making in idealized domains (Indurkhya 1992, Hofstadter & the Fluid Analogies Research Group 1995).

Conceptual clustering starts with a set of singleton vertices representing *case classes*, each containing a single case c_i, $i = 1, ..., |CB|$. The two most similar case classes CC_k and CC_l over the entire set are merged to form a new case class $CC = CC_k \cup CC_l$ that covers both. This process is repeated for each of the remaining $N - 1$ case classes, where $N = |CB|$. Merging of case classes continues until a single, all-inclusive cluster remains. At termination, a uniform, binary *hierarchy of case classes* is left.
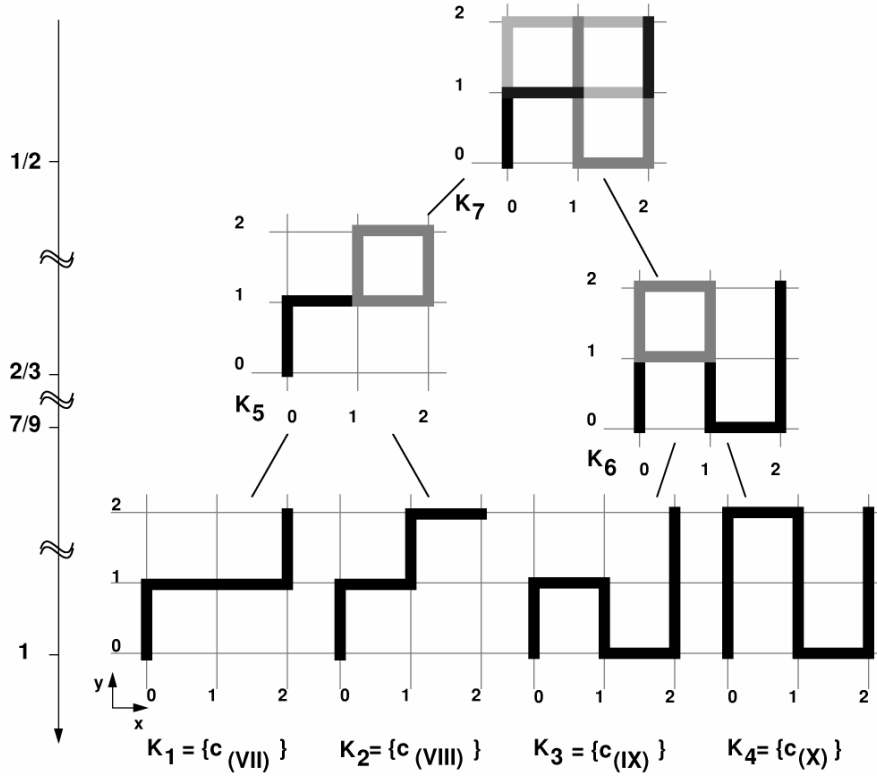

Figure 3: Concept hierarchy representing case $c_{(VII)}$ to $c_{(X)}$

Subsequently, a concept description $K(CC)$ is assigned to each case class CC of the case class hierarchy resulting in a *concept hierarchy*. The concept of a case class CC equals $n = |CC|$ (possibly empty) graphs showing the same relative frequency of their edges relative to the cases in CC.

A *concept* $K(CC)$ ist defined as $K(CC) := \{m_i^{(CC)} \mid i = 1, ..., |CC|\}$ where the vertices and edges of the graphs $m_i^{(CC)} = (V_i^{(CC)}, E_i^{(CC)})$ equal

$E_i^{(CC)} = \{(v_l, v_k) \mid (v_l, v_k) \in E^{(CC)} \wedge P_{CC}((v_l, v_k)) = \frac{i}{|CC|}\}$,

$V_i^{(CC)} = \{v \mid \exists (v_l, v_k) \in E_i^{(CC)} \ (v = v_l \vee v = v_k)\}$.

In such a way, large numbers of cases with many details can be reduced to a number of hierarchically organized concepts. The concrete cases, however, are stored to enable the dynamic reorganization and update of concepts.

Example: Figure 3 depicts the organization of four cases into a concept hierarchy. N cases are represented by $2N - 1$ case classes respective concepts $K(CC)$. Leaf vertices correspond to concrete cases and are represented by the cases themselves, e.g., $K_1 = \{c_{(VII)}\}$. Generalized concepts in the concept hierarchy are labeled K_5 to K_7 and are characterized by sets of graphs. Edges with a relative frequency of 1 are shown in black, edges with less relative frequency are drawn in grey.
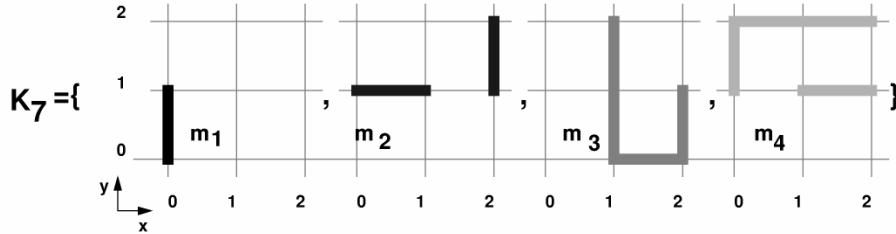

Figure 4: Graphical illustration of concept representation K_7

The similarity of these concepts is given on the left hand side. As for concept K_5 two edges with a relative frequency of 1 plus four edges with a relative frequency of $1/2$ are divided by 6 edges altogether resulting in a similarity of $2/3$.

Concept K_7 that represents $c_{(VII)}$ to $c_{(X)}$ is illustrated graphically in Fig. 4. It can be applied to generate $c_{(VII)}$ to $c_{(X)}$ as well as combinations thereof, e.g., $c_{(VI)}$ depicted in Fig. 2.

4.2 Analogical Reasoning

Analogical Reasoning is based on concepts exclusively. Given a new problem, the most applicable concept containing all problem objects is determined by searching the concept hierarchy in a top-down fashion.

The *applicability* α of a concept $K(CC)$ to solve a problem $p = (V^p, E^p)$ equals -1, if the entire graph of the concept does not contain the problem. Otherwise it equals the similarity of CC:

$$\alpha(K(CC), p) := \begin{cases} -1 & iff \ V^p \nsubseteq V^{(K(CC))} \wedge E^p \nsubseteq E^{(K(CC))} \\ \sigma(CC) & otherwise \end{cases}.$$

If $0 \leq \alpha(K(CC), p) \leq 1$ holds, then $K(CC)$ will allow to generate at least one solution of p. The concept showing the highest α value is called the *most applicable concept*[3]. It shows the highest structural similarity and solves the problem. Instead of *adapting* one or more cases to solve the problem, the concept representation $K(CC)$ of a case class CC is used to generate a set of adapted solutions $S_{CC,p}$ for a problem p. In general, there exists more than one applicable concept. The set of all solutions $S_{CB,p}$ of a CB for a problem p equals the union of solution sets $S_{CC,p}$. Finally, the set of solutions may be ordered corresponding to their quality.

The quality μ of a solution $s = (V^s, E^s)$ depends on the relative frequency of its edges with regard to the case class used. In order to determine $P_{CC}(E^s)$ every solution edge E^s need to be mapped with every edge of a case. Given the representation of the applied concept

$$K(CC) = \{(V_1^{(CC)}, E_1^{(CC)}), ..., (V_{|CC-1|}^{(CC)}, E_{|CC-1|}^{(CC)}), (V^{PT(CC)}, E^{PT(CC)})\}$$

---

[3]Note that the most similar concept may be too concrete to allow the generation of a solution.

the solution quality equals the sum of the relative frequency of solution edges divided by the total number of solution edges:

$$\mu(K(CC), s) := \frac{\sum_{i=1}^{|CC|-1} |E^s \cap E_i^{(CC)}| * \frac{i}{|CC|}}{|E^s|} \in [0, 1].$$

In this way the concept representation reduces the number of mappings required to determine the quality of a solution considerably.

Example: Figure 5 shows two problems, a number of applicable concepts taken from the concept hierarchy depicted in Fig. 3 as well as three solutions and their quality. Two solutions can be generated by applying concrete cases, i.e., $c_{(VII)}$ and $c_{(VIII)}$. The quality of the resulting solutions equals 1. The last solution (resembling the path labeled (VI) in Fig. 2) has to be combined out of case $c_{(VII)}$ to $c_{(X)}$. Its quality equals $1 + 3 * 1/4$ divided by 4 resulting in 7/16.
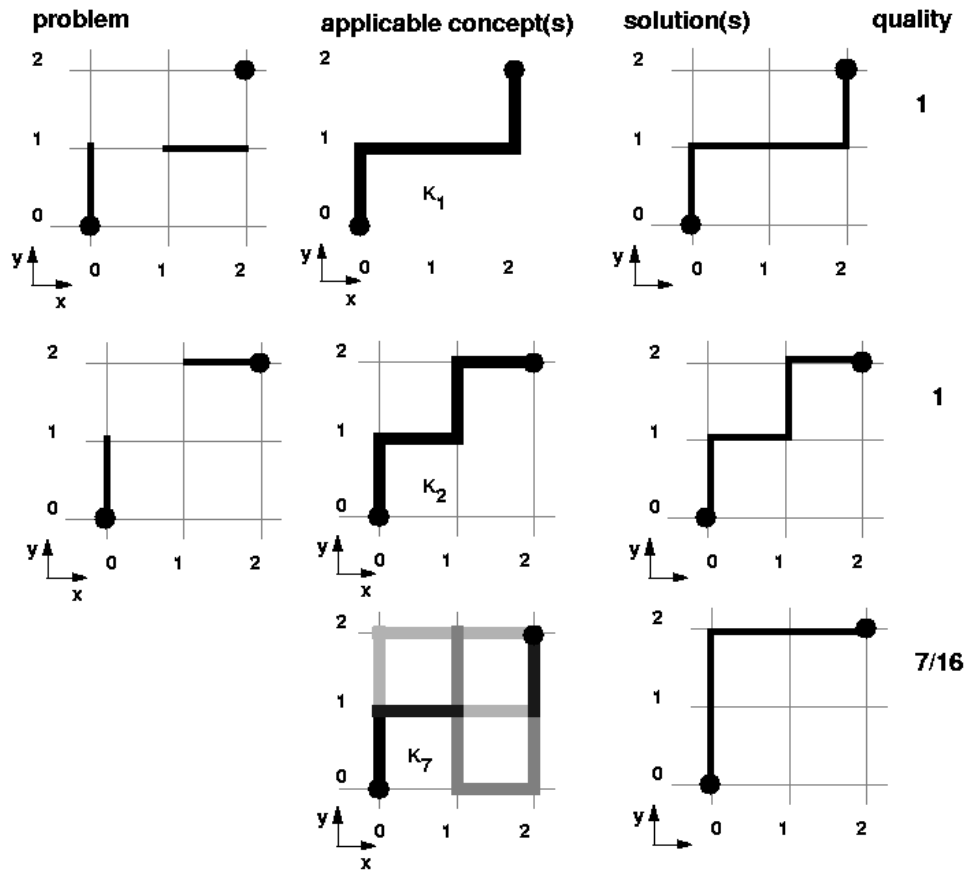

Figure 5: Problems, applicable concepts, solutions, and solution quality

If the solution was accepted by the user, its incorporation into an existing concept changes at least the relative frequency of edges. If the solution was not accepted, the case memory needs to be reorganized to incorporate the solution provided by the user.

Be aware, that the support a user gets is based on her/his past actions exclusively. So if s/he always takes a detour, the system will support this behavior and assign a high quality to these solutions.

5 Adaptive Interaction via Virtual Reality Interfaces

A major problem concerning assistance systems is the limited interaction capability of the currently used human-computer interfaces. Often, HCI is restricted to a non-intuitive use of keyboard, mouse, and screen. Extensive training is required to handle programs (e.g., a CAD tool) effectively.

Virtual Reality (VR) computer interfaces such as the *Responsive Workbench* (Krüger & Fröhlich 1994) or the Automatic Virtual Environment (CAVE) system (Cruz-Neira, Sandin & DeFanti 1993) allow multiple users to interact in a shared virtual and physical space. They support many of the perceptual channels of information that a person processes and the interface can be adapted statically to the human perceptual and effector systems. Additionally, multimodal VR interfaces permit extensive tracing of human that goes far beyond recording mouse events. The resulting behavioral protocols can serve as a basis to support human problem solving in complex tasks involving spatially organized information. VR interfaces combined with Artificial Intelligence techniques may lead to truly adaptable and generative human-computer interaction.
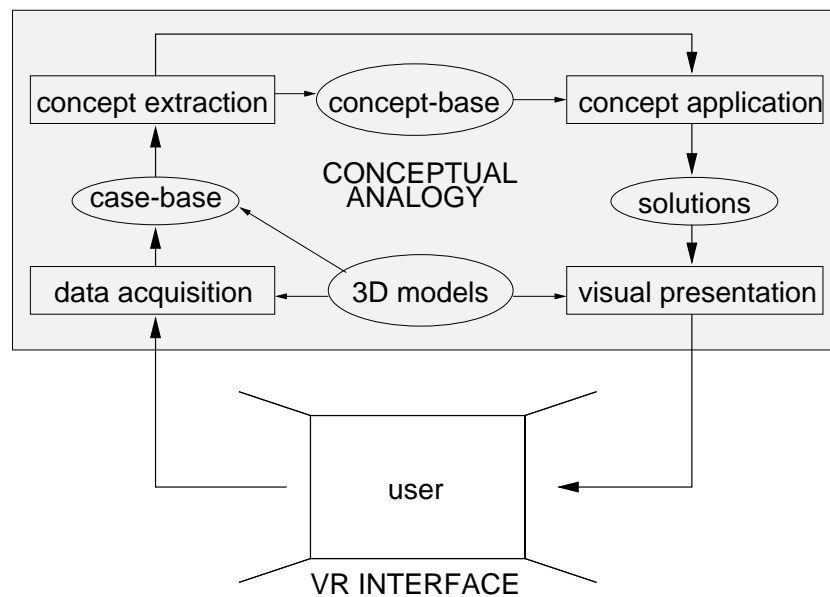

Figure 6: Merging *Conceptual Analogy* and a *VR interface* for generative, adaptive HCI

A combination of *Conceptual Analogy* with a CAVE human-computer interface is sketched in Fig. 6. In the CAVE, the user wears a set of shutter glasses that allow for time-multiplexing different images to different eyes. The glasses have a large angle of view, approximately 90 degrees per eye, and are synchronized by an infrared signal from emitters located near the scene. Using viewer-centered perspective, each eye's image is computed for that eye's exact location. Other participants see stereo, but from the tracked person's perspective. Input devices include gloves with position and orientation trackers as well as the tracked shutter glasses.

The application scenario is as follows. Entering the CAVE, the user experiences a world of virtual objects (uniform geometric solids) arranged at different places in the three dimensional labyrinth-like world. For example s/he may be asked to go from a starting point C at position 0/0 to an end point B at position 2/2 (see Fig. 1, left). The places can be reached using a variety of paths (see Fig. 2). The users task is to visit all places in a certain sequence. At each place s/he is asked to arrange the available objects by direct manipulation in a predefined way, for example, to assemble an arch, fence, tower etc. (see Fig. 1, right for an example of a tower). Simple snapping mechanisms by which objects that are placed close to each other are automatically attached are employed to ease the correct adjustment of objects without requiring force feedback. However, the variety of possible object assemblies is enormous.

In order to support navigation and manipulation the user's coordinates (place) as well as changing positions of typed objects will be time-stamped and are stored as navigation and manipulation examples in a case-base. Conceptual clustering is applied to extract a hierarchy of concepts representing knowledge about preferred navigation paths as well as about preferred object assemblies. Support will be provided by applying concepts to generate solution sets. Finally, solutions are presented to the user.

In such a way, *generative interaction* is achieved by applying the approach of *Conceptual Analogy* to support *navigation* as well as *manipulation* tasks. *Personalization of content* is done by acquiring the knowledge used for support automatically via *Conceptual Analogy.* Content is represented by concepts for navigation (e.g., about preferred paths) and concepts for manipulation (e.g., about preferred object assemblies). When the system first interacts with a user, it has no concepts about his/her preferences. During each session, new paths/assemblies are acquired, and if they are similar to previously acquired paths/assemblies, they are merged together to form more general ones. Later on these general paths/assemblies are used to support reasoning in an analogous fashion. *Personalized presentation of content* is achieved at increasing levels of generalization by using a VR interface. In the beginning, human-computer interaction for novices proceeds at a very concrete level. For example, objects to be selected might be highlighted and pathways to explore might be marked. However, as the system gains experience with a user, it collects data about his/her preferences and offers path macros and object assemblies.

Because content and its presentation are personalized together, preferred paths or assemblies can be used as well as communicated during problem solving on increasing levels of generalization. The interfaces' expressiveness scales along with the users' skill. Over time, less navigation and manipulation proceeds at higher levels of generalization, increasing the overall efficiency of human-computer interaction. Personalized abstract concepts are grounded on concrete human-computer interactions and the concrete interaction changes with the concepts that are built up.

Generative, adaptive support for navigation and manipulation is provided visually at increasing levels of generalization, for example, by concretely highlighting objects or the next place to visit, up to depicting the bounding boxes of preferred object assemblies or showing preferred paths for reaching remaining locations[4]. Instead of traversing a concrete path or manipulating single objects the user can employ *path macros* to change places instantaneously or manipulate larger *object assemblies*, thus delegating the concrete execution of the commands to the system. In such a way, the user's navigation and manipulation activities are constrained by, but not restricted to, past interactions.

---

[4]Note that there is always more than one way to guide navigation and manipulation interactions. For navigation, signs may be introduced, footsteps may visualize the number of users that went this path before or often used path may be simply broader or look nicer. On a more abstract level, one may shortcut the walk by introducing jumping points.

6   Discussion

Taken together the research extends the approach of *Conceptual Analogy* to support navigation and manipulation tasks visually at increasing levels of generalization. The approach itself can be very well used with todays WIMP GUI's. In fact, its first implementation used a CAD interface to support architectural design tasks. However, three dimensional navigation and manipulation tasks are hard to solve if two dimensional input and output devices are used. VR interfaces allow not only for three dimensional HCI but also merge haptic and visual space. The combination of VR interfaces with *Conceptual Analogy* enables generative, adaptive human-computer interaction that feels more intuitive and may provide easy-to-use support for novices and high effectiveness for experts.

In the selected domain and task, temporal and spatial constraints are used to represent object assemblies and paths. Given that temporal and spatial constraints can be seen as the basic constraints applied for human memory organization and reasoning, the results achieved in this project may transfer to other domains that can be mapped into space and time.

7   Acknowledgements

References

Aamodt, A. & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches, *AICOM* **7**: 39–59.

Börner, K. (1995). Interactive, adaptive, computer aided design, *in* M. Tan & R. Teh (eds), *The Global Design Studio. Proceedings of the 6th International Conference CAAD Futures*, Singapore: CASA, School of Architecture, pp. 627–634.

Börner, K. (1997). *Konzeptbildende Analogie: Integration von Conceptual Clustering und Analogem Schließen zur effizienten Unterstützung von Entwurfsaufgaben*, DISKI 177, Universität Kaiserslautern.

Cruz-Neira, C., Sandin, D. J. & DeFanti, T. A. (1993). Sourround-screen projection-based virtual reality: The design and implementation of the CAVE, *in* J. T. Kajiya (ed.), *Computer Graphics (Proceedings of SIGGRAPH 93)*, Vol. 27, Springer Verlag, pp. 135–142.

Hennessy, D. & Hinkle, D. (1992). Applying case-based reasoning to autoclave loading, *IEEE Expert* **7**(5): 21–26.

Hofstadter, D. R. & the Fluid Analogies Research Group (1995). *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought*, Basic Books, Inc., New York.

Indurkhya, B. (1992). *Metaphor and cognition: An interactionist approach*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Kolodner, J. L. (1993). *Case-based reasoning*, Morgan Kaufmann Publishers, San Mateo, CA.

Krüger, W. & Fröhlich, B. (1994). The Responsive Workbench, *IEEE Computer Graphics and Applications* **14**(3): 12–15.

Kurmann, D. (1995). Sculptor - A tool for intuitive architectural design, *in* M. Tan & R. Teh (eds), *Proceedings of the 6th International Conference on CAAD Futures'95*, Singapore, pp. 323–330. See also http://caad.arch.ethz.ch/ kurmann/sculptor/.

Langley, P. (1997). Machine learning for adaptive interfaces, *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, AAAI Press, pp. 763–769.

van Dam, A. (1997). Post-WIMP user interfaces, *Communications of the ACM* **40**(2): 63–67.

Zeleznik, R. C., Herndon, K. P. & Hughes, J. F. (1996). Sketch: An interface for sketching 3D scenes, *Computer Graphics (Proceedings of Siggraph'96)* .