Börner, Katy. (1998). Concept-based, adaptive human-computer interaction. Proceedings of the 9th Irish Conference on Artificial Intelligence and Cognitive Science, Dunnion, John, O'Hare, Gregory, Nuallain, Sean O., Reilly, Ronan, and Smith, Barry (Eds.), Dublin, Ireland, Published by the University College Dublin, pp. 103-109.•

Concept-based, adaptive human-computer interaction

Katy Börner

Indiana University, Computer Science Department
Bloomington, IN 47405, USA
katy@cs.indiana.edu

Abstract. Aiming at *intelligent interfaces* this paper describes research on human-computer interaction (HCI) that is based on concepts and evolves during the systems usage. The concepts are extracted out of past user interactions utilizing the approach of *Conceptual Analogy* [Bör97]. They are used to assist navigation and manipulation tasks visually and acoustically at increasing levels of generalization. Here we motivate the concept-based approach, describe the adaptive HCI that can be achieved with it, and exemplify the approach in an abstract domain.

1 Introduction

By *intelligent interfaces* we mean interfaces that allow for *generative, adaptive* human-computer interaction. Generative interaction (e.g., during *navigation* and *manipulation*) refers to the generation of useful knowledge structures (e.g., a plan or a design) to satisfy a user's goals. Adaptive interaction entails the personalization of *content* and *presentation* of it [Lan97]. Content involves the knowledge structures and inferences used to support human problem solving. The presentation of content refers to the implemented human-computer interface inclusive its interaction possibilities.

Artificial Intelligence (AI) techniques can be applied advantageously to personalize contents. *Virtual Reality* (VR) technology enables multiple channels and modes to be used for human-computer interaction. In this paper we present preliminary results of a system which combines AI techniques and a VR interface to achieve generative, adaptive human-computer interaction. We begin in Section 2 by introducing our approach which centers around enabling the system to learn unobtrusively from the user and to successively adapt according to user preferences. Section 3 sketches the approach of *Conceptual Analogy* that was described in detail elsewhere [Bör97] and is applied to achieve generative, adaptive HCI based on concepts. In section 4 we summarize our work this far and provide an outlook.

2 Adaptive human-computer interaction

Any computer system – like any person – makes certain (inter)actions easier to achieve and is more or less intuitive to understand. The systems interface and the software behind implicitly define a sort of *interaction landscape*, creating valleys that are easy to travel whereas other areas are separated by forbidding mountain ranges and are harder to reach. Ideally, the *interaction landscape* should dynamically adapt to the (kind and sequence) of operations that are used by its particular user(s).

Lets illustrate this in an abstract domain and task.[1] The abstract *domain* is a rectangular, equally spaced labyrinth-like world enriched with virtual objects. The users *task* is to *navigate* to several places and to *manipulate* (i.e., select and assemble) objects to built *object assemblies* like an arch, fence, tower etc. within a restricted time.[2] Places and objects are arranged on an underlying grid. Places can be reached using a variety of *paths* (see Fig. 1). The variety of possible object assemblies is enormous. There is almost no rule-based or model-based knowledge available to guide navigation and manipulation.

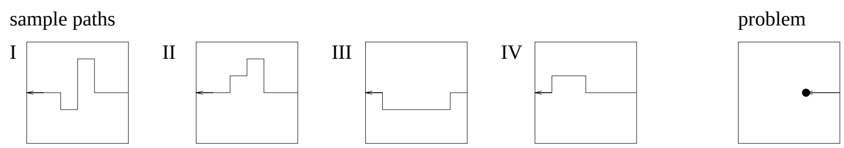


Fig. 1. Four different sample paths and a problem

A problem corresponds to a set of places/objects and some relations among them. Its solution is a path or object assembly that contains the problem places/ objects and combines past examples (i.e., paths/object assemblies) to connect all problem places/objects. Myriad combinations of past examples are possible resulting in a very large search space and usually more than one solution. Here, solutions that share a high structural similarity with past examples (i.e., which probably satisfy similar sets of constraints) are preferred.

A system aiming at an unobtrusive, adaptive support of such navigation and manipulation tasks should personalize its *knowledge structures* and *inferences* as well as the *presentation* of support to the users preferences.

Personalization of content may proceed as follows. In the first run, the system has not knowledge about the preferences of its user. Thus, no guidance can be provided during navigation or manipulation. Tackling a navigation task a second

[1] The domain and task may be instantiated into diverse domains and tasks such as architectural design, robot navigation, generating animation sequences of virtual actors, or production planning.

[2] Other research on multimodal adaptive interfaces combines automatic speech recognition, computer vision for gesture tracking, and machine learning techniques [RP97]. Our system restricts the primary mode of interaction to the navigation and manipulation of virtual places/objects.

time, the system may suggest the path used the last time. In a third run the system may combine the first and second path to support navigation. Analogously, manipulation of objects can be eased by grouping objects to object assemblies. That is, the system collects past examples of paths/object assemblies.[3] It combines them to support subsequent navigation and manipulation tasks preferring those examples that show a high structural similarity.

Personalization of presentation proceeds visually and acoustically via a VR interface. It consists of tracked glasses to experience the three dimensional world that is projected on a larger screen of about 8' by 8' size, named Wall, a tracked stylus glove to navigate and manipulate objects, and a stereo audio system.
Visually, preferred paths may be broader, lighter, contain blinking elements, show more footsteps, or more virtual actors may go on them (see Fig. 2). Here the with, level of hue or the number of blinking elements/virtual actors corresponds directly to the number of previous path selections.
Acoustically, solutions generated by combining examples of low structural similarity will be presented with a high, shrill sound reminding a user to check the solution before accepting it. Solutions showing a high structural similarity to the transferred past example(s) are presented with low, humming sounds.

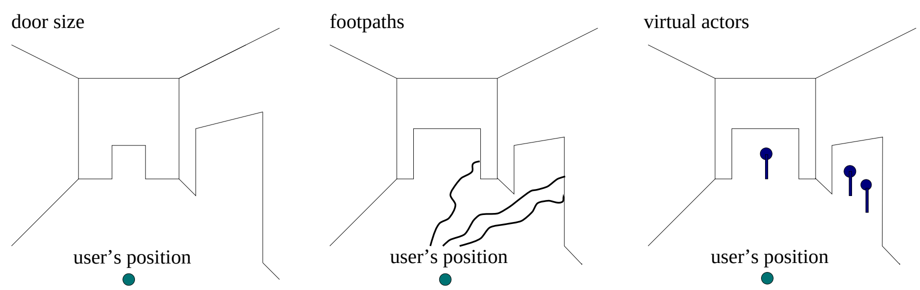


Fig. 2. Different visualizations of paths preferences at a given user position

Not only the presentation of support changes during system interaction but also the interaction itself. In the beginning the concrete next place to visit or the objects to assemble may be highlighted. Later on, the system combines examples such that the user can manipulate larger *object assemblies* or employ *path macros* to change places instantaneously. The concrete execution of the commands is delegated to the system. In such a way, generative, adaptive support for manipulation and navigation is provided visually and acoustically at increasing levels of generalization.

[3] Note that there are no negative examples.

3 Concept-based human-computer interaction

To achieve the adaptive HCI described in the last section, concepts extracted out of past user interactions may be employed. User interactions (i.e., the user's coordinates as well as changing positions of typed objects manipulated via a stylus glove) will be time-stamped and are recorded via position sensors one at the side of a pair of shutter glasses and one on the back of the stylus-glove respectively. They are internally represented by a set of *attribute vectors*:

```
place(x,y,z=4.0,3.0,1.0; time=0.2)
object(x,y,z=0.0,0.0,0.0; type=cube; time=0.3)
```

to facilitate learning of navigation and manipulation tasks in terms of these attribute primitives. For example, if a user puts several cubes over each other to built a tower, the goal would be to learn the (general) structure of a tower.

Paths and *object arrangements* are represented by sets of places/objects (subsequently denoted by x) and the temporal/spatial relations (e.g., 'place_1 before place_2' or 'object_1 below object_2') among them:

```
example( (x_1, ..., x_n), (<x_1,x_2>, ..., <x_n-1,x_n>) )
```

The temporal/spatial relations may conform to *Allens* relations [All84]. For navigation/manipulation support past examples have to be combined. Solutions sharing many relations with past examples are preferred. Case-based reasoning (CBR) [Kol93,AP94] allowing for the retrieval and adaptation of past examples (called cases) seems to be the appropriate reasoning method. However, it does not support case retrieval based on *structural similarity* and case adaptation by *case combination*, see [BCPT96] for a detailed discussion. The approach of *Conceptual Analogy* (CA) was developed to overcome these limitations [Bör97]. CA is a general approach that relies on conceptual clustering to facilitate the efficient use of past cases in analogous situations. The approach divides the overall design task into *memory organization* and *analogical reasoning* both processing graph-based case representations.

To explain *Conceptual Analogy* 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., $g^{(G)} = (\cup_{i=1}^{|G|} V^{g_i}, \cup_{i=1}^{|G|} E^{g_i}) = (V^{(G)}, E^{(G)})$. The *relative frequency* P_G of an edge (v_i, v_j) of the entire graph of G equals the cardinality of the graphs in G containing this edge divided by the cardinality of G:

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

Conform to the terminology used in case-based reasoning we define: A *case* $c = (V^c, E^c)$ is a graph (vertices represent either places or objects and edges denote temporal/spatial relations among places/objects). A *case base* CB is a finite list of (possibly identical) cases. A *problem* $p = (V^p, E^p)$ provides a set of vertices and perhaps some edges; i.e., it is a *forest.* A *solution* $s = (V^s, E^s)$ of a problem contains the problem vertices and edges and adds those vertices and edges from cases in CB that are needed to connect all problem vertices and edges.

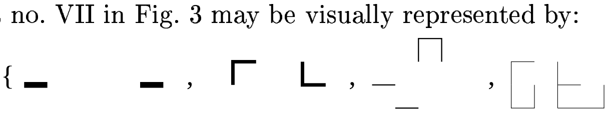
Memory organization: applies nearest-neighbor-based, agglomerative, unsupervised conceptual clustering to create a hierarchy of concepts representing cases of similar structure.

Clustering starts with a set of singleton vertices representing each case of CB by a *concept* K, i.e., a set of graphs with edges showing an identical relative frequency. The two most similar concepts and over the entire set are merged to form a new concept that covers both. The *structural similarity*[4] σ maps a set K of graphs into the interval $[0, 1]$:

$$\sigma(K) := \frac{|\cap_{i=1}^{|K|} E_i^{(K)}|}{|\cup_{i=1}^{|K|} E_i^{(K)}|} = \frac{|E_{|K|}^{(K)}|}{|\cup_{i=1}^{|K|} E_i^{(K)}|} \in [0, 1]$$

were $E_i^{(K)}$, $i = 1, ..., |K|$ is defined as $E_i^{(K)} = \{(v_l, v_k) \mid (v_l, v_k) \in E^{(K)} \wedge P_K((v_l, v_k)) = \frac{i}{|K|}\}$. This process is repeated for each of the remaining $N - 1$ concepts, where N equals the number of cases in CB. At termination, a uniform, binary *hierarchy of concepts* is left (see Fig. 3).

The concept of four cases equals four (possibly empty) graphs showing the same relative frequency of their edges relative to the cases in the case class. For example, concept no. VII in Fig. 3 may be visually represented by:

$K_{VII} = \{$  $\}$

In such a way, large amounts 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.

Analogical reasoning: is based on concepts exclusively. Given a new problem, it is classified in the most applicable concept. The *applicability* α of a concept K to solve a problem p equals -1, if the entire graph of the concept does not contain the problem and thus the concept cant solve the problem. Otherwise it equals the similarity of K. If $0 \leq \alpha(K, p) \leq 1$ holds, then K will allow to generate at least one solution of p. The concept showing the highest α value is called the *most applicable concept.*

Each solution connects all problem objects by using those vertices and edges that show the highest probability in the concept applied. Instead of *adapting* one or more cases to solve the problem as in CBR, an applicable concept K is applied to generate a set of adapted solutions $S_{K,p}$ for a problem p. In general,

[4] 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.

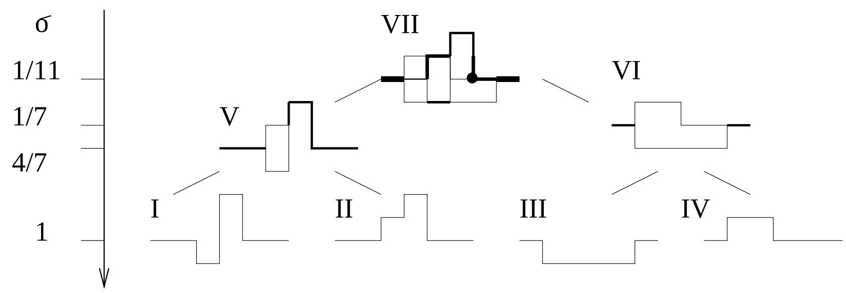


Fig. 3. Different paths and resulting concept hierarchy

there exist 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 $S_{K,p}$. It may be ordered corresponding to the *quality* μ of a solution, which corresponds to the relative frequency of its edges with regard to the concept K used to generate it, i.e.,

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

Solutions of high quality are presented with humming sound. Solutions of low quality - generated by more general concepts - are accompanied by shrill sound.

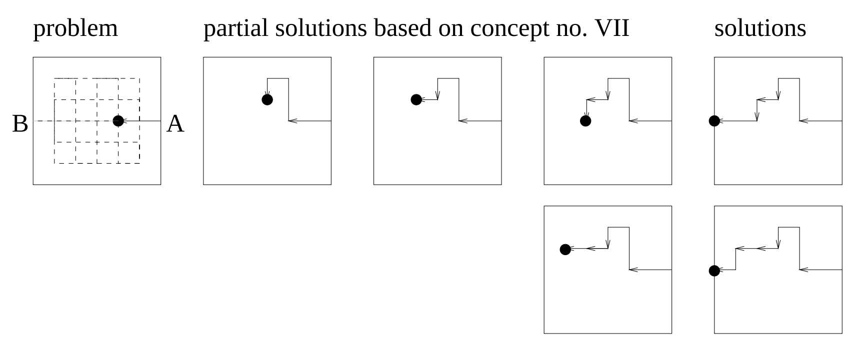


Fig. 4. Finding a path from point A to B based on examples I to IV

Figure 4 provides an example of navigation support at different user positions (denoted by a black dot). It is based on the sample paths depicted in Fig. 1 and their conceptual representation shown in Fig. 3. The upper solution corresponds to sample path no. II and shows a quality of one. The solution below was generated by applying concept no. VII. Its quality equals 0.625.

4 Summary

We have provided a snap shot of our work in progress on intelligent interfaces. The system provides an interface which learns to support manipulation and navigation tasks by natural interactions with people. The system uses a virtual world and virtual objects as natural reference for human-computer interaction. Preferred paths or object assemblies can be extracted, 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 manipulation and navigation proceeds at higher levels of generalization, increasing the overall efficiency of human-computer interaction. Personalized concepts are grounded on concrete human-computer interactions and the human-computer interaction changes with the concepts that are built up.

Some areas of our future work on concept-based, adaptive HCI include: The application of the approach to concrete domains and experiments with differently personalized presentations (visually and acoustically).

5 Acknowledgements

The author thanks the AI & VR Lab, University of Bielefeld for the resources and people that have influenced this work. The research was supported by the German Ministry for Research and Technology (BMBF) within the joint project FABEL under contract no. 413-4001-01IW104. It is now partially supported by the *German Academic Exchange Center (DAAD)*.

References

[All84] J. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[AP94] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AICOM*, 7:39–59, 1994.

[BCPT96] Katy Börner, Carl Helmut Coulon, Eberhard Pippig, and Elisabeth-Ch. Tammer. Structural similarity and adaptation. In Ian Smith and Boi Faltings, editors, *Advances in Case-based Reasoning, Proceedings of the Third European Workshop on Case-Based Reasoning (EWCBR-96)*, pages 58–75. Springer, 1996.

[Bör97] Katy Börner. *Konzeptbildende Analogie: Integration von Conceptual Clustering und Analogem Schließen zur effizienten Unterstützung von Entwurfsaufgaben.* DISKI 177, Universität Kaiserslautern, 1997.

[Kol93] Janet L. Kolodner. *Case-based reasoning.* Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[Lan97] Pat Langley. Machine learning for adaptive user interfaces. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 763–769. AAAI Press, 1997.

[RP97] Deb Roy and Alex Pentland. Multimodal adaptive interfaces. Technical Report TR 438, Vismod Publication, 1997.