USING AFFORDANCES TO SHAPE THE INTERACTION IN A HYBRID SPOKEN DIALOG SYSTEM

*Timo Baumann, Maike Paetzel, Philipp Schlesinger, Wolfgang Menzel**

Natural Language Systems Group, Department of Informatics, Universität Hamburg
baumann@informatik.uni-hamburg.de

Abstract: Affordances manifest possibilities of interaction with a spoken dialog system. For example, the act of asking a question affords to the recipient the possibility of answering. In the system we present, the observable act of maneuvering affords the possibility of controlling a motion. Our system thus uses the affordance principle to shape the interaction: to trigger the usage of instructions that are easy to understand and process, the system gives immediate visual feedback to afford user commands that can then be reacted upon. This tightening of the interaction loop requires an incremental processing paradigm to allow fast reactions and to be able to alter ongoing system actions. Our system is a hybrid of incremental and non-incremental processing components, combining conventional, state graph-based processing, which has the advantage of widely available toolkits and well-understood dialog management, with incremental dialog processing which allows for the tight feedback loop that provides for quick reactions. We tested our approach in a small user study and found that users used simpler and setting-independent commands more often and were more efficient when faced with the affordance-based version of our system.

1 Introduction

The concept of affordances [11] is widely used in human-computer interaction to model the ways in which human users react to observed system attributes. Specifically, attributes are assigned their meaning by the context of the observer and many attribute-meaning pairs are conventionalized: for example a blinking cursor, by convention, manifests the possibility of entering text. Of course, in dialog, one participant ending a turn affords the other participant to take over the turn. Using the affordance principle, we build a system that uses the affordance of motion. The system exhibits a more complex (and potentially irritating) incremental feedback behavior compared to a standard system which, however, requires more complex user behavior in the form of utterances that are potentially difficult to understand. We show that the affordance-based system outperforms a conceptually simpler system in task-efficiency and that it leads to radically more simple user utterances that simplify porting the implemented system to other settings within the domain. Overall with the affordance-based system, complexity is shifted away from speech recognition and natural language understanding (NLU) towards the timing of interaction and the tightness of the *feedback loop* between user and system. Speech recognition is still the bottleneck for many speech based systems, and hence our approach may result in improved interaction quality.

We place our system in the Pentomino domain, shown in Figure 1 (a), a puzzle game based on 12 distinct puzzle pieces (only 6 pieces were shown in our experiment at a time). In contrast to the

*The authors would like to thank Radu Comaneci and Mircea Pricop for helping with the implementation of the described system.

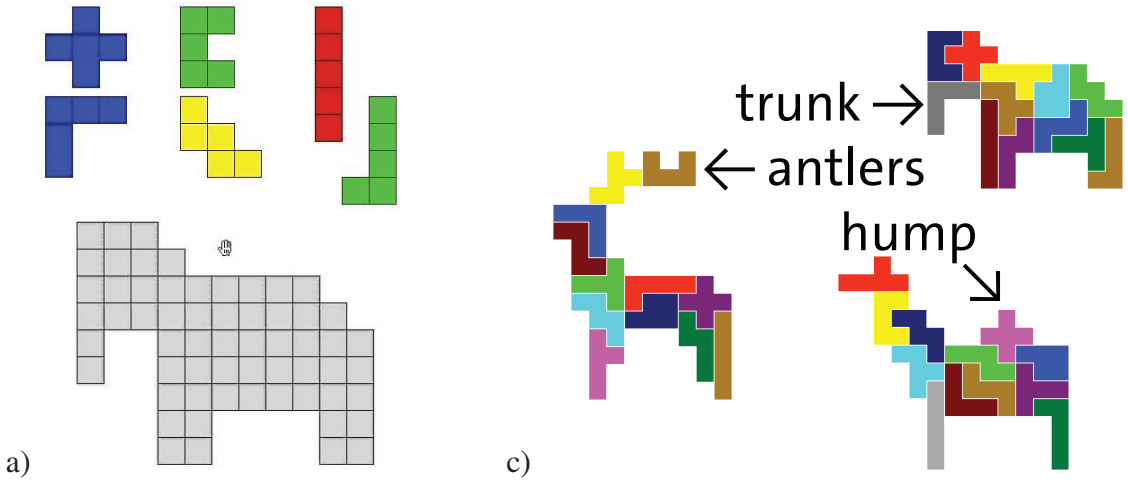

Figure 1 - The Pentomino domain: (a) The user orders the system to select and move pieces to puzzle an animal shape. (b) Different animal shapes afford radically different vocabulary for describing puzzle piece target positions.

fixed set of puzzle pieces (which already incur a large variety of references [13]), the number of all target positions (including all positions not part of the solution) is extremely large, rendering the development of NLU to resolve all possible referring expressions for all target positions a daunting task. Furthermore, the vocabulary used for referring to targets can be expected to vary widely for different animal shapes used in the puzzle, as can be seen in Figure 1 (b). (In fact, the vast majority of participants in experiments on human-human dialog that used a setting similar to Figure 1 (a) [10] referred to the figure shape as "the elephant" – thus, the puzzle target positions *afforded* to these participants vocabulary specific to elephant anatomy.)

We resolve the issue of positioning by starting to maneuver a puzzle piece right away (once it has been selected), instead of only asking where to put it. Our assumption is that the maneuvering *affords* to the user the possibility of controlling the ongoing motion and that this affordance is stronger than the affordance opened up by the system's verbalized question where to put the piece. Steering the motion radically reduces the NLU problem to understanding a fixed inventory of directional instructions instead of the open-ended problem of target position descriptions.

However, for efficiency, and to hold up the affordance of *steering*, the directional instructions need to be followed swiftly by the system, favoring an incremental approach. Incremental speech recognition is able to recognize words with little delay [2] and conceptual models [15] as well as implemented architectures [14] exist to handle incrementally updated hypotheses. Full incremental dialog management, however, is still in the early stages [7], and off-the-shelf dialog management components for complex (multi-state) domains are missing in incremental processing toolkits. To combine the best of both worlds, we take a novel *hybrid approach* where the overall control as well as the puzzle piece selection is performed using a conventional state-based system [5] and only the piece positioning is performed with incremental processing.

A system that mixes continuous with discontinuous interaction, incremental and non-incremental system components, as well as showing initiative in different modalities (speech and visual feedback) need not necessarily be a success. A small-scale user study, however, showed that the affordance-based system outperforms its baseline in efficiency without hurting user satisfaction and proved the system's robustness. The remainder of this paper is structured as follows: we point out some related work in Section 2 before we describe the interplay between the different subsystems, and their inner workings in Section 3. We then present our user study in Section 4 and discuss our observations and draw some conclusions in Section 5.

2 Related Work

Buss et al. [6] showed an interface to puzzle piece selection that incrementally understands Pentomino piece descriptions and (partially) acts as soon as possible. Their experiments showed that acting early improved task efficiency and the resulting interactions were rated as more human-like. However, the selection task is simpler than the positioning task and seldomly requires corrections of actions (as there are only four pieces and the system only acts on unambiguous references). In our system, correction occurs frequently and is handled gracefully when incrementally positioning pieces.

Soeda and Ward [16] presented a system for one-dimensional motion control (controlling the classic `xlander` game) based purely on prosody. They noted that their system was unintuitive to users who only performed well when told about the (prosodic) analyses conducted by the system and their intended meanings. Our system uses verbal information from speech recognition instead of prosody, which potentially scales much better to more complex commands and (as we show below) was very intuitive to use.

Our application of the affordance principle which shifts complexity from referring expressions towards tighter feedback loops can be compared to the approach in the *NA Generation System* [9] for the GIVE-2 challenge [12], where the system's task is to give commands and generate referring expressions. The strategy employed by the system [8] is to give simple (but ambiguous) commands. For example, when the user faces two blue buttons, one of which is to be pushed, the system simply generates "Push *a* blue button." (instead of trying to fully specify the button to be pushed). If the user (randomly) pushes the wrong blue button, the system adds "No, not this one! Look for the other one!" [9, p. 5]. That is, their simple commands afford quick actions (that are easy to repair, solving the disambiguation task through interaction); our system performs actions that afford simple commands (which are easy to understand for the system).

3 Implemented System

In our system, playing the Pentomino puzzle game consists of the two alternating sub-tasks of selecting a puzzle piece and then placing the selected puzzle piece at some target position. Our system by design does not take over initiative from the user (as some participants in human-human dialogs do [10]) and does not, for example, place pieces autonomously as soon as it becomes obvious where in the target shape they fit in.[1]

Central to our system is a domain reasoning component which contains a model of the domain: the puzzle pieces on the board including their positions, color, and shape, the target shape that the pieces should be arranged in, and the cursor which may grab and move puzzle pieces. The domain model is visualized by a view component as in Figure 1 (a). When moving, the cursor's speed indicates the distance to the target at which the cursor will stop. This allows the user to easily see whether the cursor is about to stop or will go on moving in the same direction (unless ordered otherwise).

Controllers can connect to the component using network ports and effect actions or queries. Actions influence the cursor (move, stop, grab, release), and queries about puzzle pieces indicate if an attribute-value frame matches one puzzle piece, matches none, or if it matches multiple pieces, what attribute would be best suited to disambiguate between the multiple pieces.

The dialog system proper is subdivided into two subsystems for piece selection and positioning. We will briefly describe the structure of each and then explain the control flow between the subsystems in the following subsections.

[1] In fact, the system could just position the Pentomino pieces on its own as there is only one possible arrangement to form the target shape. This, of course, would render the system quite non-interactive.

Table 1 - Example interaction consisting of (a) piece selection and (b) piece positioning. The affordant system and its baseline differ only in position strategies.

a)

	S: Which piece should I take next?
	U: The green piece.
	S: What shape is the green piece?
	U: It is shaped like a C.
	S: Did you mean this piece?
	U: Yes.

b)

baseline system	affordant system
S: Where shall I put the piece?	
U: In the left part of the head.	S: *starts and keeps moving*
S: *no reaction (not understood)*	U: In ... uh ... further left ...
U: In the forehead.	go on ... a little higher ... stop.
S: *moves piece to target*	S: *stops at target*
S: Is this the right position?	S: Is this the right position?
U: Yes.	U: Yes.

3.1 State-based Piece Selection

As layed out above, selecting a puzzle piece is a relatively easy task: the domain of puzzle pieces is small (twelve pieces) and puzzle pieces have easily distinctive features. We hence opted for a conventional, state-based system implemented in DialogOS [5]. DialogOS provides a graphical development environment for implementing state-based dialog systems, and provides nodes for speech input, speech output, for setting and querying variables, and for executing scripts. Speech recognition is based on grammars which can be annotated with semantic interpretations.

The system first uses an open question to elicit a piece description from the user and uses a sophisticated recognition grammar (including semantic interpretation) based on a corpus of puzzle piece descriptions [13] to fill slots in its attribute-value frame which is sent to the reasoning component for evaluation. Depending on whether one single piece is found, the system confirms, or asks more specific follow-up questions to disambiguate the piece (as can be seen in the example in Table 1 (a); this mode is also entered after repeated non-understanding of the user's open descriptions). The system also stores information about how the user referred to a piece, so that it can refer to that piece in the same way the user did (as in the example: "green piece").

DialogOS is limited to unweighted grammars, so the larger the grammar, the larger the probability of ending in some unwanted state. The system thus had to be limited to frequent descriptions for puzzle pieces (which is problematic as human descriptions for Pentomino pieces can be long an complex [13]). Furthermore, to avoid the recognizer getting lost, we added a time-out to reset the recognizer if it hasn't recognized the user utterance within 8 seconds. In this case, a signal is output (using a different voice) signifying to the user to restart the utterance.

To conclude, the state-based Pentomino selection subsystem is able to understand user descriptions, resolve them via the domain reasoner and to send commands to select the corresponding piece. The domain of puzzle pieces is small and the most frequent descriptions are relatively simple. In contrast, the task of positioning pieces on the board requires a much broader vocabulary. We hence decided to handle positioning differently, as described next.

Table 2 - Concepts understood and generated by the NLU component; directional actions have an attached modifier that indicates the strength of the motion.

Actions	
concept	example words
LEFT/RIGHT	"links [left]"
UP/DOWN	"höher [further up]"
CONTINUE	"nochmal [again]"
REVERSE	"zurück [back]"
STOP	"halt [stop]"
CANCEL	"von vorne [start over]"
DROP	"ablegen [release]"

Modifiers	
concept	example words
WEAK	"bisschen [a little]"
NORMAL	*default modifier*
STRONG	"weit [far]"
MAX	"ganz [to the very]"

3.2 Incremental Piece Positioning

Piece positioning uses the affordance of motion which manifests to the user the possibility of controlling that motion. Thus, the task of piece positioning is reduced from the daunting task of understanding all the possible descriptions of target positions (which even requires setting-dependent vocabulary) to the limited inventory of direction-giving commands. The aim is to let the user take control of the cursor instead of arguing with the system about how to identify the requested target position.[2] However, this approach requires an incremental dialog system so that user commands are followed swiftly, allowing to keep up the affordance of steering.

We use INPROTK [3, 4], a toolkit for incremental spoken dialog processing as the basis for our incremental piece positioning component, and implemented an NLU component tailored towards the task of steering, and a domain controller that turns NLU actions into cursor movement.

The simple incremental NLU component is based on the keywords shown in Table 2. The NLU greedily creates actions from incrementally recognized words. Actions may be modified through certain words that have preceeded the action's trigger word. The NLU must be prudent not to be too greedy (e. g. "weiter" could mean CONTINUE but also be the start of "weiter rechts", potentially the opposite direction). Thus, a more eager second pass is conducted when the user turn has ended. Actions are passed to the domain controller, and effected on screen immediately.

NLU, and domain handling are implemented as *incremental processors* and actions are *IUs* as specified by the IU model [15] that forms the basis of INPROTK. This means that the architecture flexibly and transparently recovers from intermediate speech recognizer errors: when the speech recognizer 'changes its mind' about a word in light of more context, these changes are automatically passed on to NLU and further on to the domain controller. Thus, if the recognizer intermittently hypothesizes STOP, this only leads to a short interruption in motion, which is transparently corrected when this false hypothesis is revoked.

Incremental speech recognition often hypothesizes a word before the speaker has even finished speaking it [2]. This allows for very timely system behavior, and is a requirement for commands such as STOP that need to be executed immediately in order to be intuitive.

3.3 Control Flow and Cooperation in the Hybrid System

The difficulty in the integration of two alternative subsystems is to transition between them without any apparent break in the user experience. Given the task, we are primarily interested in incrementalizing one part of the dialog (the positioning task) while the overall structure can be modelled by the conventional state-graph approach. Thus, the state-based system starts in primary control initially and only transfers control when the corresponding part in the dialog

[2]This also opens up the possibility of placing pieces along continuous dimensions, rather than in the box-grid used in the Pentomino game. Our system does not make use of this principled advantagem, though.

graph is reached. Communication between the systems uses what we call *fat states* which use script calls in DialogOS that are served by the incremental system.[3]

When the state-based system reaches a *fat state* in the graph, it calls the incremental system, supplying all necessary arguments, and waits until the script returns. Upon receiving the call, the incremental system activates its audio input, handles the subdialog with the user (in our case, positioning) and interaction with the domain controller. The incremental system returns when positioning has completed, at which point the state-based system continues processing. Return values can be used to influence the continuation. The states are 'fat' in that they may contain long and potentially complex sub-dialogs without intervention by the host system.

In our system, switching happens after the initial question of where to put the puzzle piece has been output (cmp. Table 1). In our task, the incremental system does not give any verbal but only visual feedback. However, it would be possible to intermittently return back to DialogOS, passing the request to utter a prompt and to then come back to the incremental system. That way, the system always uses the same voice, regardless of which component is currently in charge.

It appears to us that fat states would be (fairly) easy to add to any VoiceXML interpreter (or other state-based system) and allow for systems that are *selectively incremental*. This may yield advantages whenever the incrementalization of small portions of an overall dialog already results in an improved system. Switching between the two systems that we use, which were built to be used in isolation, works quickly and seemlessly.

4 User Study

We performed a small-scale user study to test our assumptions regarding the affordance of system-initiated maneuvering, the seamless integration of system components and to test the overall usability of the system.

As a baseline, we extended the state-based system to cover some typical names used for target positions in the elephant shape ("hind leg", "forehead", ...). The system then waits for a reply to the question before it starts to position the piece. Once motion has started, steering is also possible in the baseline system – it is just not afforded by the initial motion as in the full system.

The baseline system is limited (to a few targets in the elephant) and does not allow free positioning. Thus, users were given tasks which consisted of selecting a given piece and positioning it on the target. Tasks were presented visually, so as to not prime the user's verbalization.

4.1 Procedure

8 participants (university students at the department, not involved in the research) conducted two tasks with one version of the system, and then two tasks with the other version. The ordering of tasks and system versions was balanced between the participants. After each condition, users marked success, understanding, transparency, reactivity, and naturalness of selection and positioning on five-point Likert scales. The users were not told that they were subjected to two versions of the system, or how the versions differed. In a third, final questionnaire users marked differences in understanding, transparency, reactivity and naturalness between the two rounds.

We recorded the interactions to be able to investigate timing differences between systems. Two participants had severe understanding problems with the DialogOS part of the system, and were unable to finish their tasks (for either condition) within 4 minutes (after which the trial was aborted; most of the time was spent trying to select a puzzle piece). One additional recording was unavailable due to technical difficulties, so that timing information is available (and valid) for 5 participants only; the questionnaires were filled out by all 8 participants.

[3] DialogOS requires to connect to all components that it interacts with at startup, so our domain controller and INPROTK (which both reside in one process) need to be started before DialogOS (and INPROTK starts muted).

4.2 Analysis

On their first round, the four participants that started with the affordance-based system all reacted to the affordance of motion by giving directional commands instead of answering with a target description to the verbalized question. In total, seven participants reacted to the affordant system by giving instructions before the system-initial motion had completed, indicating that they were not overwhelmed by the rate of interaction. In contrast, only two participants gave direction commands to the baseline system (and these two were primed from the affordance-based system that they had experienced in the first round). This indicates to us, that the affordance of motion is indeed working as expected.

Users were significantly faster when using the affordance-based system, for both the overall interaction time (paired t-test, $p < .05$) for two tasks, and when considering the time spent in piece positioning (t-test, $p < .05$) compared to the baseline (non-affordant) system.

The questionnaires indicate an advantages for the full system but the results were not significant (due to the small number of participants): overall, the mean difference in ratings was 0.3 favoring the affordance-based system and it was rated better or identically in all ratings on average, especially in transparency and reactivity. User ratings for piece selection and positioning were strongly interdependent and the affordance-based system also outperformed the baseline in piece selection (even though this was unchanged between systems). The full system was also preferred in the direct comparison of conditions where it was rated as more transparent and more reactive.

There were no effects of task or condition ordering, another indication that the system was easy to use without any training phase.

5 Discussion and Conclusion

We have presented a hybrid dialogue system that uses the affordance of motion to shape the interaction in a way that avoids extensive, complex (and setting-specific) referring expressions, instead affording to the user the possibility of steering the cursor for piece positioning. This shifts the complexity of the interaction from the user's utterances to the way that system and user interact, specifically, to timely interaction in a tight feedback loop.

The user study supports our hypothesis that immediate maneuvering causes users to give simpler directional instructions instead of describing target positions, and that users are able to keep up with the faster paced interaction loop. Users needed significantly less time to complete tasks (possibly interaction went more smoothly and/or maneuvering time was folded into interaction time) and on average gave better ratings in a questionnaire compared to a baseline system that did not make obvious the affordance of motion. We observed that users would increase their usage of "continuous" directional commands as already noted by Aist et al. [1, p. 763]. As expected, incremental maneuvering leads to shorter, simpler utterances, which however did not result in lower naturalness ratings, indicating that naturalness is more strongly determined by the way of interaction than the complexity of utterances and understanding.

Overall, recognition and interaction worked much better in the incremental part of the system, despite that system's speech recognition models (same as in [3]) very likely being inferior to those integrated in DialogOS (Nuance version 9). This helps to show the degree to which shifting workload from speech recognition towards a tighter interaction loop can be profitable.

User testing also showed (once more) the wide variability of expressions used to name Pentomino pieces, which turned out to be the bottleneck in our system). Future work could focus on improving the piece selection part of the system, or try to transfer the results of incremental positioning to other task domains. We must note that incremental understanding need not be limited to directional commands but could include understanding of target positions, or any other user content for further performance improvements. Integration of prosodic analysis to analyze urgency of commands or to disambiguate meaning (as in [16]) is another area for future work.

A video demonstrating the system is available at `http://youtu.be/3sXh2L8Rjkc`.

References

[1] AIST, G., J. ALLEN, E. CAMPANA, C. G. GALLO, S. STONESS, M. SWIFT and M. K. TANENHAUS: *Incremental Dialogue System Faster than and Preferred to its Nonincremental Counterpart*. In *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, pp. 761–766, Nashville, USA, 2007.

[2] BAUMANN, T., M. ATTERER and D. SCHLANGEN: *Assessing and Improving the Performance of Speech Recognition for Incremental Systems*. In *Proceedings of NAACL-HLT 2009*, pp. 380–388, Boulder, USA, 2009.

[3] BAUMANN, T., O. BUSS and D. SCHLANGEN: *InproTK in Action: Open-Source Software for Building German-Speaking Incremental Spoken Dialogue Systems*. In *Proceedings of ESSV*, Berlin, Germany, 2010.

[4] BAUMANN, T. and D. SCHLANGEN: *The INPROTK 2012 Release*. In *Proceedings of SDCTD*, Montréal, Canada, 2012.

[5] BOBBERT, D. and M. WOLSKA: *Dialog OS: an extensible platform for teaching spoken dialogue systems*. In ARTSTEIN, R. and L. VIEU (eds.): *Decalog 2007: Proceedings of the 11th Workshop on the Semantics of Dialogue*, pp. 159–160, Trento, Italy, 2007.

[6] BUSS, O., T. BAUMANN and D. SCHLANGEN: *Collaborating on Utterances with a Spoken Dialogue System Using an ISU-based Approach to Incremental Dialogue Management*. In *Proceedings of SigDial 2010*, Tokyo, Japan, 2010.

[7] BUSS, O. and D. SCHLANGEN: *DIUM – An Incremental Dialogue Manager That Can Produce Self-Corrections*. In *Proceedings of SemDial 2011 (Los Angelogue)*, Los Angeles, USA, 2011.

[8] DENIS, A.: *Generating Referring Expressions with Reference Domain Theory*. In *Proceedings of INLG*, pp. 27–35, Trim, Ireland, 2010.

[9] DENIS, A., M. AMOIA, L. BENOTTI, L. PEREZ-BELTRACHINI, C. GARDENT and T. OSSWALD: *The GIVE-2 Nancy Generation Systems NA and NM*. Techn. Rep. INRIA-00541578, INRIA, 2010.

[10] FERNÁNDEZ, R., T. LUCHT, K. RODRIGUEZ and D. SCHLANGEN: *Interaction in Task-Oriented Human–Human Dialogue: The Effects of Different Turn-Taking Policies*. In *Proc.s of the First International IEEE/ACL Workshop on Spoken Language Technology*, 2006.

[11] JENKINS, H. S.: *Gibson's "Affordances": Evolution of a Pivotal Concept*. Journal of Scientific Psychology, pp. 34–45, Dec. 2008.

[12] KOLLER, A., K. STRIEGNITZ, A. GARGETT, D. BYRON, J. CASSELL, R. DALE, J. MOORE and J. OBERLANDER: *Report on the Second NLG Challenge on Generating Instructions in Virtual Environments (GIVE-2)*. In *Proceedings of INLG*, 2010.

[13] SCHLANGEN, D., T. BAUMANN and M. ATTERER: *Incremental Reference Resolution: The Task, Metrics for Evaluation, and a Bayesian Filtering Model that is Sensitive to Disfluencies*. In *Proceedings of SigDial 2009*, London, UK, 2009.

[14] SCHLANGEN, D., T. BAUMANN, H. BUSCHMEIER, O. BUSS, S. KOPP, G. SKANTZE and R. YAGHOUBZADEH: *Middleware for Incremental Processing in Conversational Agents*. In *Proceedings of SigDial 2010*, Tokyo, Japan, Sept. 2010.

[15] SCHLANGEN, D. and G. SKANTZE: *A General, Abstract Model of Incremental Dialogue Processing*. In *Proceedings of the EACL*, pp. 710–718, Athens, Greece, 2009.

[16] SOEDA, S. and N. WARD: *Design for a System able to use Time-Critical Spoken Advice*. In *Proceedings of the 15th Annual Conference of JSAI*, Matsue, Japan, 2001.