Analyzing a Process of Collaborative Game Design Involving Online Tools

Sandra B. Fan[1], Brian R. Johnson[2], Yun-En Liu[1], Tyler S. Robison[1], Rolfe R. Schmidt[1], Steven L. Tanimoto[1]
1. Department of Computer Science and Engineering
2. Department of Architecture
University of Washington, Seattle, WA 98195, USA
1: {sbfan, yunliu, trobison, rolfe, tanimoto}@cs.uw.edu; 2: brj@uw.edu

Abstract

We explore modeling problem solving and design using state-space-search methodology by engaging in the design of an educational game. We also explore how online communication tools (OCTs) could be used to support collaborative design using two online tools: 1) CoSolve, a collaborative problem-solving environment we developed, and 2) INFACT, a discussion forum that we built for use in education. We used these tools to design GoAtom, a chemistry game. Using our game design experience as an example, we present a method for modeling design processes using state-space-search, and reflect on our use of OCTs.

1. Introduction

We expect the role of online communication tools (OCTs) in collaborative problem solving to grow in importance as new affordances make OCTs more effective and participatory. Hence, we seek better and more novel ways that this technology can be used to support online collaborative problem solving and design.

We used OCTs to collaboratively design a face-to-face educational children's game: GoAtom. For the design we used our own INFACT system [5] as an OCT for GoAtom. We took a state-space-search approach to modeling the design process, inspired by the classical AI theory of problem solving, and used our own online CoSolve environment for modeling the problem state-space.

We first describe existing OCTs and the state-space-search design methodology. Then we describe our specific design experience creating this game. Next, we describe our state-space model of design as applied to our creation of this game, and present results of our analysis. Finally, we briefly report on our usage of the tools and offer suggestions for improving the design process we engaged in.

2. Background

There are many systems that address improving online communication for collaboration. These include CSILE and Knowledge Forum (for collaborative learning), HyperNews (a general threaded discussion forum tool), instant messaging and chat programs, and wikis. Strengths and weaknesses of such systems as representational affordances for collaborative learners are discussed in [4].

One OCT we used, INFACT, offers a threaded asynchronous discussion forum. The second tool, CoSolve, is a web-based system for collaborative problem solving, inspired by applying classical AI state-space-search theory to design [3]. State spaces are commonly taught in artificial intelligence courses as a means to automatically solve puzzles and play games. We use state-spaces at a "meta" level; each state of the design space represents a description of a possible design (e.g., a game), rather than a snapshot in the process of solving a puzzle. We then apply operators to a state to introduce new axes (*state variables*) of a design, or specify their formal types.

A *design step* means adding or removing a state variable from an existing state, changing its formal type, or changing its value. The formal type might be Boolean, or Enumeration over a set of n elements. A *design iteration* consists of one or more design steps.

CoSolve provides affordances for both posing and solving problems. When posing, users perform a kind of end-user-programming to create *problem templates*. A poser expresses a template by defining an initial state, and a set of operators. Next, to solve a problem, users instantiate a template and explore solution paths during a *solving session*. In this sense, CoSolve shares some of the goals of Meta-Design in that users not only solve problems, but are able to create and modify the very descriptions of the problems they are trying to solve [2]. For this activity, we created a problem template with a set of operators that manipulate parameters specially selected to model game design.

978-0-7695-4206-5/10 $26.00 © 2010 IEEE
DOI 10.1109/VLHCC.2010.19

3. The Game Design Process

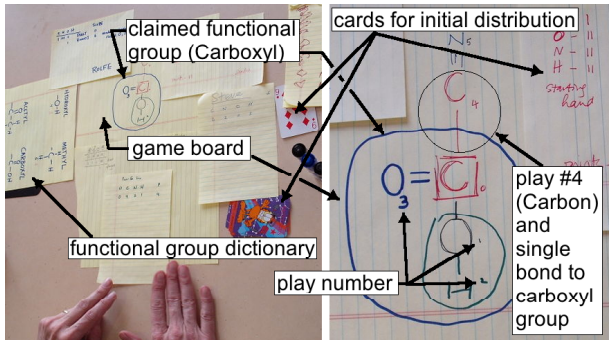

Figure 1. Example of a GoAtom game in progress. In the photo on the right, groups of atoms that form a functional group are circled by the player that won points for that group.

Our design team consisted of four graduate students (computer science) and two faculty (architecture and computer science), participating in a 10-week seminar on the study of problem solving and design. We engaged in the design of the GoAtom game to explore our ideas about design and OCTs.

GoAtom involves basic organic chemistry concepts. Players are dealt a "hand" of Carbon, Oxygen, Hydrogen and Nitrogen atoms. An initial Carbon atom begins play. In turn, each player bonds an atom, from his own hand, to an atom currently in play, creating a molecule consistent with chemistry bonding rules. Players gain points for each atom or bond they add to the structure, and the game ends when no more bonds or atoms can be added that follow these chemistry bonding rules. The player who ends the game gets to "claim" the molecule, and may receive points for it.

The team also designed an additional card game called Eco-avelli. Eco-avelli is a card game meant to model the political processes related to global warming; we used Google Wave and CoSolve for communication in this design. Due to space constraints, more details can instead be found in [1].

The team met face-to-face over several roughly hour-long design sessions; we also used OCTs between sessions and during sessions. In each session, we completed one or two design iterations, beginning with discussion of problems in the current version of the design, brainstorming and deciding on a modified set of game rules, playing the game, and analyzing how well the new rules enhanced or detracted from the game. The design process began with initial game ideas posted as messages to INFACT; there were nine such messages. Once we decided on the GoAtom idea, 40 additional messages were posted to INFACT under nine different threads. We held two face-to-face design sessions and played four design iterations: Iteration 1 and Iterations 2A & 2B (simultaneous iterations involving half the group in each iteration) during Session 1, and Iteration 3 during Session 2. Nineteen messages were posted before Session 1, three during Session 1, five between Sessions 1 & 2, and 13 messages during Session 2.

Before Session 1, a problem template was created in CoSolve to describe the state-space of the design, consisting of: (i) a scoring scheme of point values for each atom or bond, (ii) a scoring scheme for claiming different types of completed molecules, (iii) specification of the number and types of atoms in the initial hand of each player, (iv) whether or not the initial hand is random, (v) if random, minimum allocations of atoms in the initial hand.

We tested our designs by playing the game on pen and paper. Figure 1 shows two photographs from Iteration 3. Playing cards were used to randomly pick each player's initial hand of atoms. On the right, a play number is shown depicting the order of moves for recording purposes.

In designing this game, we explored how adjusting the rules affected game-play while maximizing the playability of the game (i.e., is it fun, easy-to-learn?), and teaching chemistry concepts. We hypothesized that by changing our scoring scheme, we could produce the right game-play dynamics to achieve this balance.

Some iterations were devoted to fixing game-play issues, such as balancing scoring (e.g. Iteration 1). Others focused on the educational value of the game. For example, in Iteration 3, we brainstormed ideas to increase the chemistry content in the game. We decided on a Scrabble-like version of GoAtom, with points awarded for building molecules that were real chemical compounds, rather than simply any that fit the basic bonding rules. We tested this idea by adding a small "dictionary" of functional groups from organic chemistry, and awarded extra points for building or identifying these groups. We found that this did increase our ability to identify functional groups, so we further discussed ways a more general "dictionary" of molecules could be developed for the game. This is an example of a design decision could not be accounted for in the initial state space; it required adding additional variables to the state space.

4. State Space Analysis / Iterations

We used CoSolve's state space representation to explicitly model the current state of our design and the space of all alternatives under consideration, allowing

us to study the evolution of our design. We will now use two simple metrics to quantify these states to help us understand the extent to which typical design decisions can be understood as state space explorations. The simplicity of these measurements may seem crude, but is offset by the advantage that it is easy to understand any artifacts they introduce.

Formally, our design process consisted of a set of iterations, I, and a relation **predecessor**(η_1, η_2), $\eta_i \in I$. that indicates when the iteration η_1 is the direct predecessor of iteration η_2 . In all cases we considered, the predecessor relation induced a directed tree structure on the set of iterations.

Each iteration $\eta_i \in I$ is specified by the set of state variables and their values at that iteration. Given a state variable v, we denote the value of that state variable at iteration η by $\eta(v)$. If v is not in the set of state variables for η , define $\eta(v) = \perp$. We will need to refer to the set of state variables used in an iteration, and denote this by **vars**(η). In our application, each state variable was either boolean, integer with a fixed number of bits, or a finite enumeration of options. Thus each state variable v can be assigned a size $s(v)$ which is the number of bits needed to specify the value of v.

When moving from one iteration to another, the following *elementary design operations* changes may be applied:

1. The value of a state variable may change.
2. A new state variable may be added.
3. An existing state variable may be removed.
4. An existing state variable may be refined (split into multiple, new state variables).
5. Multiple existing state variables may be combined into a new, single state variable.
6. The domain of an existing state variable may be changed.

Changes of type 1 can be explored using standard state space exploration, as CoSolve was designed to support. Changes of type 2-6 involve modifications of the underlying state space and must be explored in a broader framework.

4.1 Design Process Metrics

At each design iteration, we measure the size of the state space under consideration and the size of the design change from the previous iteration. In particular we have:

Definition 1. The *effective state space size*, **Size**(η), of an iteration $\eta \in I$ is the number of bits needed to specify the state of the design to the design team at that iteration. Formally:

$$\mathbf{Size}(\eta) = \sum_{v \in \mathbf{vars}(\eta)} s(v)$$

We can also measure the size of the change in an iteration by counting the number of state variables that changed to produce this iteration:

Definition 2. At an iteration η let

$$\Delta(\eta) = \{\ v \mid \eta'(v) \neq \eta(v) \wedge \mathbf{predecessor}(\eta', \eta)\ \}$$

denote the set of changes state variables. We will call $|\Delta(\eta)|$ the *iteration magnitude at* η. It is the number of state variables that have different values in η than in its predecessors.

With this framework in place, it is straightforward to define other more sophisticated metrics of design evolution. We propose these two as a simple first glance of the design process.

5. Results & Discussion

The effective state space size and iteration magnitude were computed per design iteration of GoAtom. First we observe the evolution of the state space size for the game (Figure 2).

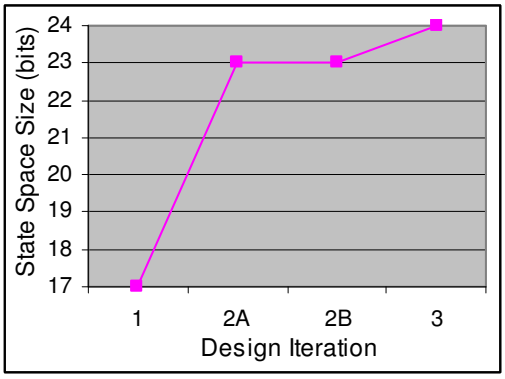

Figure 2. Evolution of State Space Size

We see that the state space grew at each iteration. In particular this means we never saw an iteration characterized entirely by Type 1 changes. Thus pure state space search would not have been effective at this phase of the design process, a fact that became clear to us as we saw our CoSolve state space specifications become obsolete after a few minutes of discussion in each round. This suggests that early stage design will better benefit from a more general state space designer, based on the elementary design operations 1-6. We also measured each iteration's magnitude (Figure 3),

but found that this slight increase does not seem to be correlated with increasing state size (see [1]).

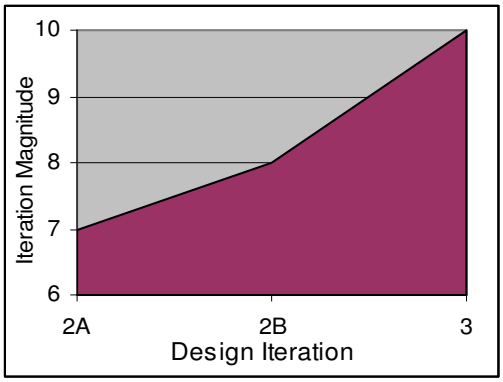

Figure 3. Iteration Magnitude

We also note that the state space encoding of an iteration is not unique—variables may be given different names, enumerated variables may be factored into Cartesian products of smaller variables, and unused variables may be added or removed. While these factors will not affect the qualitative nature of the results, inconsistent state space design will exaggerate the importance of some changes while understating others. Because of this, consistent state space design is important for effective quantification of design evolution.

In regards to our other OCT, INFACT, we used it primarily to record specific goals for each face-to-face meeting, to summarize the rule set played in each iteration, as well as capture comments about game play during that iteration. Given a chance to design another game, we would have made several changes. For example, it would have been useful to store online collaborative documents such as an overall design criteria specification—so that we can keep track of our larger goals, or keep a brainstorming list of new ideas to try. As it was, we occasionally forgot good game mechanisms proposed but not used in previous iterations. We also found it would have helped to tag design steps with a label indicating the purpose of the change. Finally, we would also have collected hidden-response surveys after each iteration, to obtain unbiased feedback regarding that iteration from each member. This type of data could also serve as a quantitative measure of progress; over time, do the designers feel the game design is meeting its goals better than in previous iterations? If enough designers rate an iteration poorly, the design path could revert to a previous iteration.

6. Conclusion

Our analysis of the use of the OCTs INFACT and CoSolve provided us with many new ideas for ways to support collaborative design. We found that the state-space-search model, as provided by CoSolve, not only gave us a way to communicate and think about the design process explicitly, but we were able to quantify our design changes in a concrete way through analysis of the state space variables of a design. In this analysis, we found that in almost all iterations the state space grew. This suggests that in this early stage of design we engaged in, a simple state space search is not enough; rather, the construction of the state space through elementary design operations such as adding new state variables plays a bigger role.

While this work in still in its preliminary stages, we hope our observations will help spur the creation of more efficient tools for communication and collaborative design.

Acknowledgments

Partial support for this research by the National Science Foundation under grant 0613550 is gratefully acknowledged.

7. References

[1] Fan, S., Johnson, B., Liu, Y., Robison, T., Schmidt, R., Tanimoto, S. 2010. Analyzing a Process of Collaborative Game Design Involving Online Tools (long version). http://www.cs.washington.edu/ole/collab-design-long-2010.pdf

[2] Fischer, G., and Giaccardi, E., 2004. Meta-Design: A Framework for the Future of End-User Development. In Lieberman, H., Paternò, F., Wulf, V. (Eds): *End User Development - Empowering People to Flexibly Employ Advanced Information and Communication Technology*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

[3] Simon, H. 1996. *The Sciences of the Artificial*, 3rd ed. Cambridge: MA: MIT Press.

[4] Suthers, D., Vatrapu, R., Medina, R., Joseph, S., and Dwyer, N. 2008. Beyond threaded discussion: Representational guidance in asynchronous collaborative learning environments. *Computers & Education*, Vol. 50, No. 4, pp. 1103-1127.

[5] Tanimoto, S., Carlson, A., Husted, J., Hunt, E., Larsson, J., Madigan, D., and Minstrell, J. 2002. Text Forum Features for Small Group Discussions with Facet-Based Pedagogy, presented at *CSCL 2002*, Boulder, CO.