

# Anatomy of an Adaptive Scaffold for Analysts

Stephanie Santosa  
Oculus Info Inc.<sup>1</sup>

Judi McCuaig  
University of Guelph<sup>2</sup>

Joseph MacInnes  
Oculus Info Inc.<sup>1</sup>

Nathan Kronenfeld  
Oculus Info Inc.<sup>1</sup>

William Wright  
Oculus Info Inc.<sup>1</sup>

## ABSTRACT

Adaptive user interfaces offer the potential to improve the learnability of software tools and analytic methodologies by tailoring the operation and experience to a user's needs. Scaffolding is an instructional strategy that can be applied by adaptive interfaces to achieve this. Scaffolding theory suggests that the level of guidance should be adjusted to optimize learning and performance levels. This paper explores the use of adaptive techniques to scaffold user interaction and presents a taxonomy of techniques for adaptive scaffolds within complex software systems. The techniques identified in the proposed taxonomy can help software scaffolds select appropriate adaptations in response to the user's learning and operating needs. A scaffold called nAble was implemented to explore the application of adaptive techniques from the taxonomy to support an analysis methodology called the Analysis of Competing Hypotheses (ACH).

## Keywords

Adaptive Interfaces, Scaffolding

## ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation (e.g., HCI)] User Interfaces – *training, help, and documentation, user-centered design.*

## 1. INTRODUCTION

Introducing new software tools or methods to users in a workplace typically requires either a classroom based training program with simplified toy problems or computer-based training during which students are led through a set of lessons and exercises. In-house trainers may follow-up to reinforce lessons and to help the users perform actual tasks with the new tools and methods. The time required for someone to learn new software and associated techniques can be days to several weeks, which represents time away from normal work duties. Formal training can be a barrier to the use and exploitation of new capabilities. Adaptive scaffolding can assist users to learn both the methodology and the software tool “buttonology” and so reduce the need for formal training.

Scaffolding is a training strategy used by human instructors to

help learners achieve more than they could independently. A good scaffold helps the student to perform beyond current skill levels and fades away as the student gains expertise in the task [10]. The effectiveness of scaffolding lies in the ongoing diagnosis of the student's understanding and the tailoring of support given accordingly. An adaptive interface uses models of users, tasks, interface components and domains along with inference techniques to personalize interaction with the user [3],[7]. A software scaffold is a type of adaptive interface that can, like a human instructor, dynamically adapt interactions with the user [2],[6]. Users of any expertise level can interact successfully with a scaffolded application whether they are acquiring new skills or using well understood procedures.

This paper presents a taxonomy of techniques for adaptive scaffolding within complex software systems such as those used for visual analytics. Section two begins with a description of each of the adaptive techniques in the taxonomy. Section three follows with a description of “nAble”, an adaptive scaffold implementation within a tool for intelligence analysis. Section four concludes with a description of how the taxonomy fits into potential future research.

## 2. TAXONOMY OF ADAPTATION TECHNIQUES FOR SCAFFOLDING

This taxonomy was derived from an iterative process involving the examination of previously implemented scaffolding systems along with data from human tutors in an initial scaffolding experiment [9]. The categories in the taxonomy are shown in Table 1 below.

Table 1. Categories of adaptive scaffolding techniques.

1. Recommend Information	6. Scale Automation
2. Support Bootstrapping	7. Adjust Error Recognition
3. Clarify Meaning	8. Change Feedback
4. Manage Attention	9. Scale Complexity
5. Support Sub-Task	

The techniques identified in the proposed taxonomy help software scaffolds select appropriate adaptations in response to the user's learning and operating needs. These techniques can be performed by adapting different aspects of the interface, including GUI elements such as menus and toolbars, interaction methods such as gestures and hotkeys, navigational paths such

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
ECCE 2009, September 30 – October 2, 2009, Helsinki, Finland.  
Copyright 2009 ACM 1-58113-000-0/00/0004...\$5.00.

<sup>1</sup> Oculus Info Inc.

2 Berkeley Street, Suite 600, Toronto, Ontario, Canada  
{ssantosa, jmacinnes, nkronenfeld, bwright}@oculusinfo.com

<sup>2</sup> University of Guelph  
Guelph, Ontario, Canada  
judi@uoguelph.ca

as links, instructional content, and methods of content delivery such as popups and sounds. Models of user expertise, along with learning style, preferences, interests, tasks, interruptability, and other properties can drive the scaffold's adaptation decisions. Categories in this taxonomy are not exclusive. For example, a bootstrapping technique might also incorporate recommendations and attention management. The following discusses each category.

**Recommend Information.** Providing the user with tailored content at the appropriate times reduces information overload associated with learning. An adaptive system can make suggestions about where to find appropriate information or can modify the instructional content presented to the user. It could, for example, suggest collaborators with whom the learner could connect with, navigational paths for the learner to follow, or specific actions to perform. A scaffold could even provide step-by-step tutorials for raw novices and remove or tailor them as expertise grows.

**Support Bootstrapping.** Bootstrapping is brief, direct and specialized pre-task guidance which gets a user started with the task. Bootstrapping adaptations allow novice users to overcome the intimidation and frustration of starting with a blank slate. An example of this type of adaptation is an instructional note identifying possible actions for novice users on how to begin. Interface elements can also be highlighted to achieve this bootstrapping effect.

**Clarify Meaning.** Visual annotations can be used to elaborate meaning for elements of the methodology, instructional material or the interface. Exaggerated visual elements can implicitly introduce the meaning of a feature and then fade away to a more succinct form once the feature is understood. Fading annotations optimizes the screen space for both novice and expert users. For example, large icons with labels could be used in interfaces for novice users, which gradually transform into an expert interface with smaller, simplified icons.

**Manage Attention.** Visual signals, such as increasing size, flashing, animating, or highlighting, as well as auditory cues such as beeping, can direct user attention to the appropriate operating control or analysis step. The goal of attention management is to direct user focus. These techniques are disruptive by nature, so they are typically applied when it is certain that the user should be attending to something specific or is doing something incorrect. A scaffold can guide the user through a task by focusing the user's attention on relevant interface elements, thus prompting the user to perform, and learn, the required actions. Attention management is a natural solution for guiding novice users to useful functions.

**Support Subtask.** A task model can help an adaptive system determine the current and future steps of a user's process. Using a task model, adaptations can be made that specifically support the user's current objectives. An adaptive scaffold can limit the interaction area based on sub-task, provide guidance on a methodology being followed, and enable or disable specific links. For example, the nAble scaffold described in the next section uses a Hidden Markov task model to determine a user's current subtask and provides task-specific instructions on the methodology and tool.

**Scale Automation.** A novice user's tasks can often be automated, or partially completed with the help of the adaptive scaffold. The balance of control can shift from the system to the user as expertise is developed. A scaffold initially assumes more task responsibility, and then fades this support. This will allow

novices to immediately be productive and to gradually perform more tasks independently with flexibility. Adapting the extent of machine automation can improve efficiency and task performance. Automation may be disruptive to some users and beneficial to others, so tuning it to user properties can help achieve a correct balance.

**Adjust Error Recognition.** The actions or degree of inaccuracy for an error to be signaled in the system can vary by user. This allows for more appropriate error recognition since knowledge about the user can indicate what to register as an error. An adaptive scaffold can be more tolerant of errors from novice users and expect more precision and accuracy as users grow in proficiency with the system. This technique was used by the human tutors in the initial scaffolding experiments described in [9], who tolerated less thoroughness on evidence examination initially. As expertise developed, tutors increased expectations and consequently showed less tolerance before identifying an analysis error. Errors can be in methodology as in the experiment, or in 'buttonology' when related to tool operation.

**Change Feedback.** The type and method of feedback can be adapted to suit user characteristics such as learning style or expertise level. Novices might be provided with validation and encouragement for accomplishing basic tasks, while experts might only receive feedback about potential errors. The human tutors in the initial scaffolding experiments continuously validated the actions of complete novices, and this validation was reduced as the subjects gained competence. This type of adaptation is used in intelligent tutoring systems where the feedback given after each step is adapted to the learner [1],[4]. Another form of feedback can be at the interaction level. This can include providing marking menus to help novice users with learning gestures and removing the feedback as their expertise increases [12].

**Scale Complexity.** An interface's visual and control complexity can be adapted to match the user's skill level. For example, the interface could hide or gray out elements that are not currently useful, and then introduce more functionality when appropriate. Layering the complexity of an application can help reduce cognitive load. This is especially important for complex tools or during early learning when the user is overloaded. Alternatively, the workflow could be scaled to provide a simpler, streamlined methodology for novice users to follow, and progress to more demanding ones as competency is gained.

This taxonomy for adaptive interfaces can be used to guide the selection of techniques for implementing adaptive scaffolds. A system designer may systematically go through each category and decide which adaptations to implement based on what is possible given the original non-adaptive user interface for the system, what would be most useful for the task in question, and the models available to provide input to the reasoning system.

### 3. nABLE IN nSPACE2

nSpace2 is a rich, web-based integrated cognitive workspace used in information analysis and is comprised of TRIST [8] for information triage and the Sandbox [11] for evidence marshalling and visual sense making. The Sandbox is a flexible, visual thinking environment that supports both formal and informal analytic methods. The goal of nAble is to provide functionality and training, when needed and just as it is needed, using an adaptive scaffold. The nAble scaffolding system consists of sensors of user activity, Hidden Markov task models, Bayes reasoning engine for expertise and learning style, adaptive scaffolding techniques and dashboards for delivery of

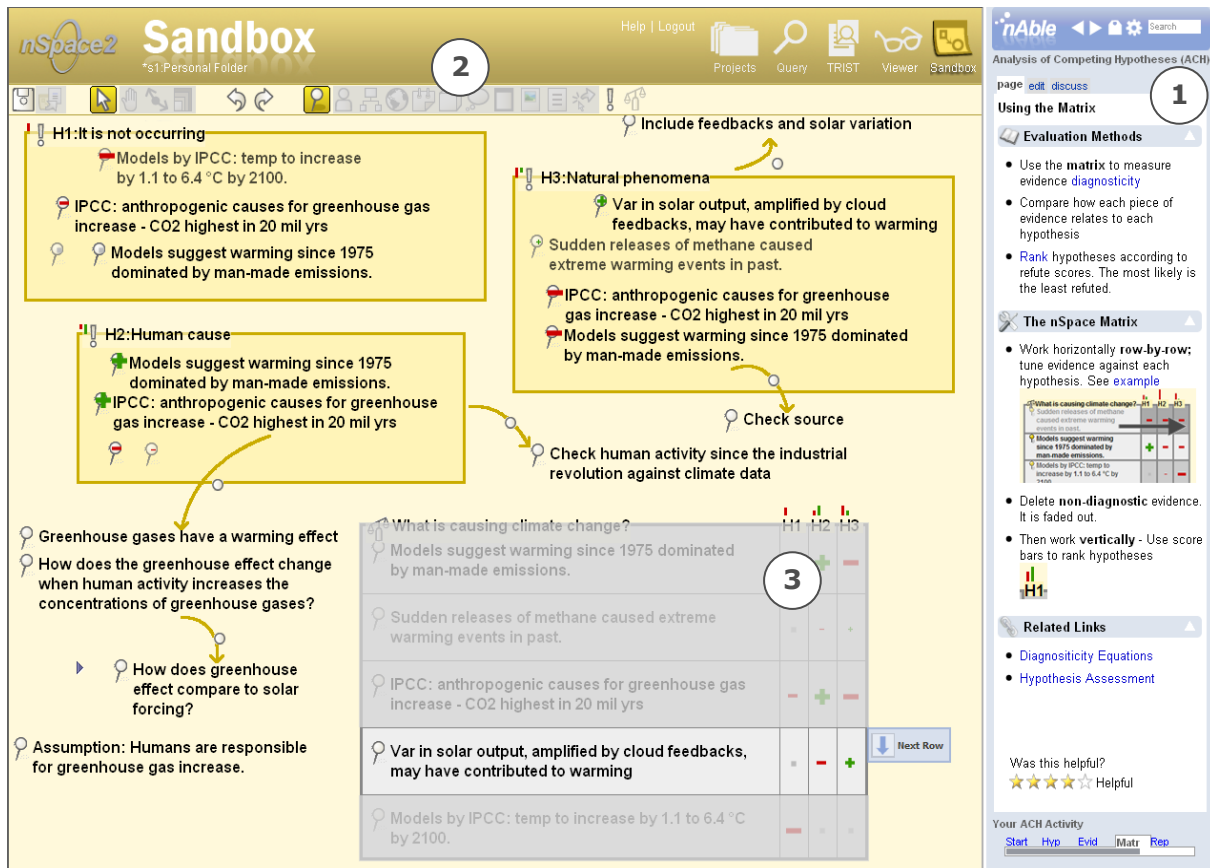


Figure 1. Examples of nAble's web-based adaptive techniques during the ACH subtask *assigning evidence*. Wiki-style sidebar (1) for recommendation of information and subtask support with automated scaffold progression. Manage attention and scale complexity of (2) interface and (3) task with masking techniques.

operational or methodology instruction. Results from the initial experiment using the Able scaffold suggest a significant improvement over traditional help and are described in [9].

The nAble scaffold described in this paper focuses on one formal analysis tool which implements the Analysis of Competing Hypothesis (ACH) methodology within the Sandbox. ACH is a well documented, formalized approach to weighing alternative explanations in order to minimize the likelihood of analytic errors and bias [5]. The nAble scaffold supports users in performing ACH subtasks, including *naming the issue, identifying hypotheses, gathering evidence, assigning evidence, assessing diagnosticity, reviewing the analysis and writing the report*.

nAble for nSpace2 is a set of network services providing the decision engine, task models and rich internet application for the adaptive scaffolding interface. The scaffold runs in the browser as a layer above the host application and queries the decision engine and task model using Service Oriented Architecture (SOA). User behavior is sensed through the browser, not the host application. Similarly adaptive techniques are sent directly to the browser. This gives the nAble scaffold autonomy from the host application, but consequently also requires a creative approach to implementing some of the adaptive techniques. Figure 1 shows the nAble scaffold for nSpace2 and highlights some of the techniques used in its delivery.

Scaling complexity required a new approach from the previous desktop nAble scaffold [9] since we could no longer directly

modify the host's interface to remove advanced features. We solved this problem using a masking technique to limit user interaction with advanced components while still providing visual information regarding the component's purpose. The semi-transparent widgets provide the user information about which features were being scaled, which prepares the user for interaction with that feature. In Figure 1, the system is masking items in the toolbar to reduce the interface complexity for the user. This is also an attention management technique, which guides user attention away from unneeded toolbar items in order to reduce cognitive load.

Recommendations and subtask support are presented to the user with nAble's Wiki-style sidebar. The sidebar is the focal point for new help content, links to sites of interest and suggestions for expert collaborators. Content is segmented into information chunks, which are organized according to user and task models. Thus, recommended content is adapted to expertise, learning style, and personality traits as detected by the user model. Users can read, rate, tag and even contribute to recommended content. User tags and ratings are used in future decisions about content chunk selection for adaptations. Figure 1 displays the Wiki-style sidebar populated by the information chunks that are relevant to novice users performing the ACH subtask of assessing diagnosticity.

Feedback is given to users about task progression through an interactive progress widget located at the bottom of the Wiki sidebar, (3) in Figure 1. This widget provides users with an overview of the overall objective, feedback on current steps, and a sense of progress as steps are achieved. The steps and

progression also adjust according to user expertise, as the steps for a novice user are simplified and initial progress is more visible.

Additionally, instructional pop-up dialogues are used to support bootstrapping, recommend information, manage attention and support sub-task. They actively push relevant information to the user by appearing in contextually determined locations directly in the user's workspace, allowing the system to effectively "point" things out to the user.

The problem of providing scaled automation as an adaptive technique was also solved using masking techniques. As shown by (3) in Figure 1, when the user's subtask is assessing diagnosticity, the scaffold masks the ACH matrix to force the user through a row-by-row examination of each item of evidence, which is an important step in ACH methodology [5]. This automation is scaled back when the user achieves sufficient competence.

nAble uses a network service exposing a Hidden Markov task model of the ACH methodology. The scaffold can send information to the network service in order to determine which subtask the user is working on. Most of the adaptive scaffolding techniques coordinate with the task model service to determine the timing of each adaptation.

#### 4. CONCLUSIONS

Adaptive scaffolding is a powerful instructional approach which allows novice users to perform at higher levels while they are learning new tasks and techniques. This paper has presented a taxonomy of adaptive scaffolding techniques that can be used by designers of adaptive interfaces to scaffold learners in complex software environments. A prototype scaffold for the ACH analytic methodology was also presented to demonstrate how key categories could be developed for a web-based analytic system. The nAble scaffold prototype leverages service-oriented architecture to create a scaffold that is autonomous from the host application.

Several avenues of future research are possible. While the generalizability of the taxonomy was tested through additional user interface design exercises and literature reviews, additional experimentation with implemented methods would confirm and refine the taxonomy. Further prototyping and experimenting is required to develop a broader portfolio of adaptive techniques and to evaluate the effectiveness of each. Although adaptive techniques from the taxonomy were implemented in the web environment to illustrate the particular challenges of scaffolding in web-based applications, the taxonomy can be applied across various technologies, including stand-alone systems and synthetic worlds. These adaptations will conceptually work across many domains; however variation in implementation will certainly be required to carry out these techniques.

This presented taxonomy and prototype are a part of a larger project for understanding and improving analytic workflow. We aim to develop a knowledge formalization for information analysis which allows sharing of knowledge between analysts and machine learning algorithms.

Next steps for the nAble scaffold are to build on the task recognition and training system to allow more sophisticated interface adaptations to analytic methodologies. More specifically, we are exploring machine-learning techniques to model various structured analytic methodologies, recognize when they are being used by an analyst, recommend additional methodologies, and perform aspects of the task automatically

where it would be beneficial. These adaptations would offer strong benefits to information analysis communities as new techniques could be captured and propagated as they are developed and used. We aim to develop a reasoning system which assists all users, novice and experts alike, in structured analytic techniques and informal critical thinking methods.

#### 5. REFERENCES

- [1] Alevan, V., and Koedinger, K. 2002. An effective metacognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science*, 26, 147-179.
- [2] Conati, C., and VanLehn, K. 2001. Providing adaptive support to the understanding of instructional material. In *Proceedings of the 6th international conference on Intelligent user interfaces*, (Santa Fe, NM, January 14-17, 2001). ACM Press, New York, NY, 41-47.
- [3] Fischer, G. 2001. User Modeling in Human-Computer Interaction. *User Modeling and User-Adapted Interaction*, 11(1-2), Kluwer Academic Publishers, Hingham, MA, 65-86.
- [4] Graesser, A., Chipman, P., Haynes, B., and Olney, A. 2005. AutoTutor: an intelligent tutoring system with mixed-initiative dialogue. *Education, IEEE Transactions on*, 48, 4, 612-618.
- [5] Heuer, J. 1999. *Psychology of intelligence analysis*. Center for the Study of Intelligence Central Intelligence Agency.
- [6] Jackson, S. L., Krajcik, J., and Soloway, E. 1998. The design of guided learner-adaptable scaffolding in interactive learning environments. In *Proceedings of SIGCHI conference on Human factors in computing systems* (Los Angeles, CA, April 18-23, 1998). ACM Press/Addison-Wesley Publishing Co., 187-194.
- [7] Jameson, A. 2002. Adaptive interfaces and agents. *Human-computer interaction handbook*, A. Sears, ed. Lawrence Erlbaum Associates Inc., Hillsdale, NJ, 305-330.
- [8] Jonker, D., Wright, W., Schroh, D., Proulx, P., and Cort, B. 2005. Information Triage with TRIST. In *International Conference on Intelligence Analysis* (May 2-4, 2005, McLean, VA).
- [9] MacInnes, J., Santosa, S., Kronenfeld, N., McCuaig, J., and Wright, W. 2008. nAble Adaptive Scaffolding Agent - Intelligent Support for Novices. In *Web Intelligence and Intelligent Agent Technology* (Sydney, Australia, December 9-12, 2008). *WI-IAT '08. 3*, IEEE Computer Society, Washington, DC, 365-368.
- [10] Wood, D., Bruner, J. S., and Ross, G. 1976. The role of tutoring in problem solving. *J Child Psychol Psychiatry*, 17(2), 89-100.
- [11] Wright, W., Schroh, D., Proulx, P., Skaburskis, A., and Cort, B. 2006. The Sandbox for analysis: concepts and methods. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (April 22-27, 2006, Montreal, Canada). ACM Press, New York, NY, 801-810.
- [12] Zhao, S. and Balakrishnan, 2004. R. Simple vs. compound mark hierarchical marking menus. In *Proceedings of the 17th annual symposium on User interface software and technology* (October 24-27, 2004, Santa Fe, NM), ACM Press, New York, NY, 33-42.