Supporting Flexible Business Processes with a Progression Model

Nicole Stavness
Department of Computer Science
University of Saskatchewan
Saskatoon, SK S7N 5A9 Canada
+1 306 966 4886
nicole.stavness@usask.ca

Kevin A. Schneider
Department of Computer Science
University of Saskatchewan
Saskatoon, SK S7N 5A9 Canada
+1 306 966 4891
kas@cs.usask.ca

ABSTRACT
Users require flexibility when interacting with information systems to contend with changing business processes, and to support diverse workflow. Model-based user interface design needs to accommodate flexible business processes by integrating workflow activities with other model approaches. We present a progression model to help in developing systems that support flexible business processes. The progression model tracks a user's interaction when using a transaction–based information system. The steps a user takes to create a transaction and the state of the transaction at each step is made explicit. This provides the user with a mechanism to deal with partial information, interrupted and concurrent transaction entry, and the processing of multiple transactions. The paper describes the progression model along with the potential workflow benefits of recording progressions.

Keywords
Business Process, Workflow, Task Model, Use Cases, Interactive System, Information System, Process Model, Data Model

INTRODUCTION
Business organizations perform processes to ensure that work progresses towards business goals. A *business process* is a set of one or more linked activities that collectively realize a business goal, normally within the context of an organizational structure defining functional roles and relationships. Business processes are a required part of everyday business, whether explicit or implicit, as their fulfillment results in consistent and reliable work. Interactive software systems are developed to automate part or all of the process. A common term used to describe the orderliness and control of business processes is *workflow*.

This paper presents preliminary work on a model-based approach (the progression model) for integrating transaction–based information systems with workflow activities. The progression model captures a user's interaction with an information system by recording both the data the user enters and the steps the user takes to enter the data. In this way, the progression model combines both process model and data model information. The next section discusses related work. We then present the progression model, followed by an example scenario using the progression model. Finally, we conclude with a discussion of a prototype implementation and the future direction of this research.

RELATED WORK
A number of approaches have been used to model business processes. Workflow models are used to describe the flow of work in an organization and with external organizations. Task models define the possible actions available for a user to accomplish a goal. Use cases describe stories that define ways a user can use a system. Each of these approaches is discussed in this section.

Workflow Models
The workflow model is used to represent the flow of work within a department, across a company or to external agents. The Workflow Management Coalition (WFMC) defines workflow as the automation of a business process, in whole or part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules [13].

Workflow research focuses on approaches to making changes during the business process. Procedure-like, routine processes that are statically supported are on one end of a continuum, and highly unspecified, dynamic processes are on the other end [2]. In adaptive workflow management systems [10] the procedural rules can be changed or created during the process. Some research proposes a cooperative hypermedia system, with process support through a meta-model, to integrate the efforts towards communication, coordination, and cooperation in workflow systems [5]. A two-part classification defines types of possible flexibilities that may be desired in workflow management applications [7]. 'Flexibility by selection' provides the user some leniency in executing a process by offering multiple execution paths. Alternatively, 'flexibility by adaption' provides the ability to add extra execution paths through additional functionality and tools that allow the workflow type to change and integrate during runtime.

Although these approaches facilitate dynamic intervention of business processes, they tend to be difficult for the user to control. The intricate workflow model usually requires developer's intervention to make appropriate changes that uphold the constraints of the system.

Task Models
Task Models are logical descriptions of activities that are to be carried out in reaching a user's goals in an interactive system. There are many different approaches to task

modeling such as Hierarchical Task Analysis [1], GOMS [4], UAN [6], and ConcurTaskTrees [11].

Hierarchical Task Analysis (HTA) is based on describing the set of goals, tasks and operations in logical structures of different levels. GOMS (goals, operators, methods, selection) depicts procedural knowledge or 'how-to-do-it" knowledge through fine-grained operators that are performed to reach a goal. UAN (User Action Notation) also follows a hierarchical structure. It provides a notation for designers to describe the dynamic behaviour of graphical user interfaces, where the tasks are represented asynchronously with operators that denote the temporal relationships. The ConcurTaskTrees notation was created to support engineering approaches to task modeling. Temporal relationships are also incorporated for enabling, concurrency, disabling, interruption, and optionality.

In relation to business processes, task models describe the paths of activities available to reach the user's goals. Task models have also been looked at in relation to workflow [12]. Unfortunately, task models often result in large specifications with more detail than is necessary for a designer. Recent research has investigated annotating task models with data artifacts to better support information systems and extracting dialog models from the task model to better support automated generation of user interfaces [9].

Use Cases

Uses cases [8] (or scenarios) are used to organize individual user interactions with the system. The main purpose of use cases is to build system requirements. To bridge the gap between use cases and more detailed design views, Use Case Maps are identified for depicting scenarios [3].

An aspect particularly beneficial for business processes is the ability to combine and integrate scenarios to model dynamic runtime behaviour. The stubs can be static, containing one plug-in map, or dynamic, containing multiple plug-in maps. The selection of a plug-in map in the dynamic stubs can be determined at runtime according to selection policies. The selection policies are usually defined in a formal language. Although the various paths are statically defined, the selection of the path is indicated dynamically.

THE PROGRESSION MODEL

The progression model makes explicit the steps and transactions a user makes when using a transaction–based information system. As the user progresses towards accomplishing a task or goal, the progression model infrastructure records each step and the state of the transaction.

Making the steps and transactions explicit allows the user to group transactions into batches for later processing, to store partial transactions for later editing, and allows the user to browse historical progressions. Linking the steps in the workflow directly to the transaction provides a means to integrate the process model and the data model in one coherent model. This enables the support of the flow of work for an individual user by supporting new interactions. A series of definitions outline the basic aspects of a progression. Consequently, new interactions are enabled to provide flexible business process support.

The following definitions describe the key elements of the progression model and how they relate to each other. These items are graphically depicted in Figure 1.

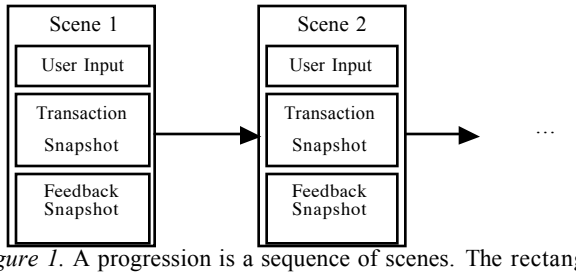

Figure 1. A progression is a sequence of scenes. The rectangles represent scenes that encompass user input, a transaction snapshot, and a feedback snapshot.

Progression. A progression is a sequence of scenes (or steps) in a process to create a transaction.

Progression Interval. A progression interval is a subsequence of a progression, that is, one or more steps.

Scene. A scene provides the user with the current state of their interaction. A scene contains the user's input, the current state of the transaction, and feedback to the user through the user interface. A scene captures the process and associated data as a user performs actions throughout a progression.

User Input. The user input is a series of zero or more user actions interleaved with one or more workflow actions.

Workflow Action. A workflow action is an action that affects the flow of the progression. A workflow action may change the user interface to a new scene, recall a past progression, replay a progression, save a progression, and so on.

Feedback. Feedback is the information provided to the user through the user interface. It is captured in a feedback snapshot, a representative rendering of the user interface.

Transaction. A transaction is a data structure that holds the accumulation of information at each point in the progression. Each transaction is made up of a set of elements that are accumulated throughout the progression by user actions. As the scenes change, the element additions, deletions, and changes are reflected in the transaction. For instance, if the user is filling out a wizard form, at every submission the new information is added to the transaction. A transaction snapshot is the state of the transaction at a specific step in a progression.

EXAMPLE PROGRESSION SCENARIO

This section provides a simple example that demonstrates a process where the user can benefit from the progression model. The scenario consists of a customer buying a fish from an online pet store. The web-based system Java Pet Store is a sample application from the Java 2 Platform, Enterprise Edition ("J2EE") BluePrints Program from Java Software, Sun Microsystems. It is important to note that the Pet Store software first needs to be modified to support progressions, however, this is a minimal change achieved

by composing the progression mechanism with the application.

Pet Store Example

The scenario is described in a series of steps as follows. Figure 2 depicts a progression interval of steps 4 and 5 with the user input, transaction, and the feedback snapshot.

1. The user enters the website where the feedback snapshot is the homepage. The elements in the transaction are initially empty.
2. The user selects a Fish pet.
3. The user chooses an Angelfish product.
4. The user decides on the Small Angelfish item.
5. The user selects the "Add to Cart" button.
6. The user edits the quantity text box of the Small Angelfish to 6 and selects "Update Cart."
7. The user continues to the checkout by selecting the "Checkout" button.

The interface to the user appears the same as the regular online pet store, with the exception of the additional transaction snapshot being displayed. As the user performs their regular web browsing actions, they are provided with the transaction information concurrently.

As the user steps through a purchase transaction, a number of benefits result from using progressions. Some of the benefits are more apparent or have higher returns in more complex examples. However, for the simplicity of this paper, this example provides initial insight.

Transaction Review and Edit

The collected transaction data is available for review by the user in the form of a transaction snapshot. Any information editable by the user is displayed. For example, the initial quantity has a default value of 1 provided by the system; however, it is included in the transaction as the user can change the quantity.

Progression History

The user can use the progression history to change previous decisions. For example, the user may decide just before they are about to "check out" that instead of the Small Angelfish, they prefer the Large Angelfish. They can easily undo the actions and move back to an earlier point in the progression history.

Additionally, users can utilize the replay features to assist their shopping. For instance, infrequent users of certain progressions may desire to see what they bought and how they shopped the last time they made an order. The replay feature would allow them to view the path they previously took and the information that resulted from it. Alternatively, the user may wish to directly submit the same order they used in a previous shopping experience.

If a user must leave their shopping session prior to completing the transaction they can save their position within the progression, as well as the accumulated information, including the history and transaction, to return to at a later time.

<progression name="buyFish" time="8:50"**>**
…
<scene name="selectItem" time="9:05"**>**
<userInput>
<userAction> … </userAction> …
<workflowAction value="selectItem">
<item>SmallAngelfish</item></workflowAction>
</userInput>
<transaction>
<category value="Fish" />
<product value="Angelfish" />
<item value="Small Angelfish" />
</transaction> …
<feedback>

Small Angelfish

List Price: $16.50

Your Price: $10.00

Add to Cart

</feedback>
</scene>
<scene name="addToCart**"** time="9:07"**>**
<userInput>
<userAction> … </userAction> …
<workflowAction value="addToCart" />
</userInput>
<transaction>

Buy Fish Progression	
CATEGORY	Fish
PRODUCT	Angelfish
ITEM	Small Angelfish
QUANTITY	1

</transaction>
<feedback>

Shopping Cart

Small Angelfish	Remove	1	@ $16.50
Update Cart		Total:	$16.50

Check Out

</feedback>
</scene>
…
</progression>

Figure 1. Pet Store Progression Interval. The progression interval represents two scenes depicting the information recorded from the user input along with the resulting output in a markup language.

Multiple Progressions

The user may record a number of progressions. In our example, providing the shopper with the ability to manage multiple progressions allows the shopper to comparison shop.

Managing batches of progressions is useful if a user decided they wanted to submit a number of progressions at one time. For instance, the user may play out a number of different scenarios without officially checking out of the store. However, at a later point in time, she could decide that she wanted to follow through with three of the saved progressions. She could simply select the progressions and indicate that all should be checked out.

CONCLUSION

For successful model-based design of user interfaces it is important that the models are integrated. Our approach integrates workflow activities (progressions) with the data schema (transactions) of an information system. Initial indications show that traditional model-based approaches may be integrated using a workflow model, such as the one discussed here. We are currently developing a more complete prototype to demonstrate the applicability of our model.

REFERENCES

1. Annett, J., Duncan, K.D., Stammers, R.B., & Gray, M.J. (1971). Task analysis. London: Her Majesty's Stationery Office.
2. Bernstein, A. How Can Cooperative Work Tools Support Dynamic Group Processes? Bridging the Specificity Frontier. (CSCW'00), 2000, pp. 279-288.
3. Buhr, R.J.A. Use Case Maps as architecture entities for complex systems. IEEE Transactions on Software Engineering, 24(12):1131-1155, December 1998.
4. Card, S. K., Moran, T. P., and Newell, A., The Psychology of Human-Computer Interaction , Lawrence Erlbaum, 1983.
5. Haake, J., Wang, W. Flexible Support for Business Processes: Extending Cooperative Hypermedia with Process Support. (GROUP'97), 1997, pp. 341-350.
6. Hartson, H. R, Siochi, A. C., Hix, D. The UAN: a user-oriented representation for direct manipulation interface designs. ACM Transactions on Information Systems, Volume 8, Issue 3, July 1990, pp. 181-203.
7. Heinl, P., Horn, S., Jablonski, S., Neeb, J., Stein, K., Teschke, M. A Comprehensive Approach to Flexibility in Workflow Management Systems. (WACC'99), 1999, pp. 79-88.
8. Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G., Object-Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley, Harlow, England, 1992.
9. Luyten, K., Clerckx, T., Coninx, K., Vanderdonckt, J. Derivation of a Dialog Model from a Task Model by Activity Chain Extraction. (DSV-IS'2003), Funchal, Madeira Island (Portugal), 2003, ©Springer-Verlag 2003.
10. Narendra, N. C., Adaptive Workflow Management – An Integrated Approach and System Architecture. (SAC'00), 2000, pp. 858-865.
11. Paternò, F., Mancini, C., Meniconi, S. ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. (Proceedings Interact'97), Chapman&Hall, 1997, pp.362-369.
12. Traetteberg, H.: Modeling work: Workflow and Task modeling. In: Vanderdonckt, J., Puerta, A.R. (eds.): Proc. of 3 rd Int. Conf. on Computer-Aided Design of User Interfaces CADUI'99 (Louvain-la-Neuve, 21-23 October 1999). Kluwer Academics, Dordrecht (1999) 275–280.
13. WfMC. Workflow Management Coalition Terminology & Glossary, WFMC-TC-1011, Document Status- Issue 2.0, June 1996.