Knowledge Engineering with Didactic Knowledge First Steps towards an Ultimate Goal

Rainer Knauf

Faculty of Computer Science and Automation
Ilmenau University of Technology
PO Box 100565, 98684 Ilmenau, Germany
rainer.knauf@tu-ilmenau.de

Abstract. Generally, learning systems suffer from a lack of an explicit and adaptable didactic design. Since e-learning systems are digital by their very nature, their introduction rises the issue of modeling the didactic design in a way that implies the chance to apply AI Techniques. A previously introduced modeling approach called storyboarding is setting the stage to apply Knowledge Engineering Technologies to verify and validate the didactics behind a learning process. Moreover, didactics can be refined according to revealed weaknesses and proven excellence. Successful didactic patterns can be explored by applying Data Mining techniques to the various ways students went through the storyboard and their associated level of success.

1 Introduction

University instruction often suffers from a lack of didactic design. Since universities are also research institutions, their professors are usually hired based on their topical skills. Didactic skills are often underestimated in the recruiting process. So far, the ad hoc application of didactic skills in teaching situations is not formally modeled for use by less experienced instructors. Moreover, much of such skills are not represented at all, but just “implemented” in the heads of experienced teachers [Chiang and Chung 2006].

To make didactic design explicit, a modeling approach called storyboarding is outlined here. Besides providing didactic support, a (semi-) formal model such as storyboarding is setting the stage to apply Knowledge Engineering Technologies to verify and validate the didactics behind a learning process. Moreover, didactics can be refined according to revealed weaknesses and proven excellence. Successful didactic patterns can be explored by applying Data Mining techniques to the various ways students went through a storyboard and their associated level of success. As a result, future instructors and students may utilize these results by preferring successful ways through a storyboard.

The paper is organized as follows. The next section briefly outlines the storyboard concept including the present state of the current development. This is followed by an overview on Knowledge Engineering Technologies, which have been developed and implemented for storyboards. Finally, we summarize the research undertaken so far and outline current work as well as research horizons.

2 Storyboarding

Our storyboard concept is built upon standard concepts which enjoy (1) clarity by providing a high-level modeling approach, (2) simplicity, which enables everybody to become a storyboard author, and (3) visual appearance as graphs.

A storyboard is a nested hierarchy of directed graphs with annotated nodes and annotated edges. Nodes are scenes or episodes. Scenes denote leaves of the nesting hierarchy. Episodes denote a sub-graph. There is exactly one Start- and End- node to each (sub) graph. Edges specify transitions between nodes. They may be single-color or bi-color. Nodes and edges have (pre-defined) key attributes and may have free attributes.

A storyboard is the authors' (instructors') design document representing various expectations of the users' (learners') behavior. Storyboards on educational processes can be traversed in different manners according to (1) users' interests, objectives, and desires, (2) didactic preferences[1], (3) the sequence of nodes (and other storyboards) visited before, i.e. according to the educational history, (4) available resources (like time, money, equipment to present material, and so on) and (5) other application driven circumstances.

A storyboard may be seen as a model of an anticipated reception process that is interpreted as follows:

- *Scenes* denote a non-decomposable learning activity that can be implemented in any way.
- *Episodes* are defined by their sub-graph.
- *Graphs are* interpreted by the paths, on which they can be *traversed*.
- A *Start Node* / *End Node* of a (sub-) graph defines the starting / target point of a legal graph traversing.
- *Edges* denote transitions between nodes. There are rules to leave a node by an outgoing edge: (1) The outgoing edge must have the same color as the incoming edge by which the node was reached. (2) If there is a condition specified as the edge's key attribute, this condition has to be met for leaving the node by this edge.
- *Key attributes of nodes* specify application driven information, which is necessary for all nodes of the same type, e.g. actors and locations. *Key attributes of edges* specify conditions, which have to be true for traversing on this edge.
- *Free attributes* specify whatever the storyboard author wants the user to know: didactic intentions, useful methods, necessary equipment, e.g.

3 Knowledge Engineering with Storyboards

3.1 Formal Verification of Storyboards

Our concept of storyboarding is a semi-formal one. The graph hierarchy is completely formal and below the level of scenes is is completely informal. Thus, the scenes form the interface between the formal and the informal levels. The formal levels are the

[1] In the authors' experience, some students understand better by presenting illustrations, others by providing a small example and others by providing formal descriptions.

key feature to support the logical reliability such as consistency, completeness, non-redundancy, and so on. To ensure consistency and completeness of our storyboards, we developed and implemented several verification procedures:

1. An *Episode - Hierarchy - Test* focuses questions such as whether every episode has exactly one related graph and vise versa.
2. Also, the *reachability* of each node (in particular, of the *End Node*) from the *Start Node* is checked.
3. Furthermore, *completeness* and *non-contradictoriness* of alternative outgoing edges (with the same beginning color) is checked.
4. Edge colors, which express the *interdependence of incoming/outgoing edges*, are also a subject of formal verification by checking, whether (1) there is a unique (beginning) color of the Start node's outgoing edges and (2) at least one outgoing edge with the same (beginning) color for each incoming edge's (finishing) colors.

The above mentioned anomaly tests are implemented for our storyboard development environment [Sauerstein 2006,Duesel 2007].

3.2 An Inheritance Concept

Additionally, an inheritance concept within the graph hierarchy was implemented, which distinguishes several inheritance types such as sum, maximum, or set union for inheritance within the graph hierarchy.

1. In some applications it makes sense to inherit annotations from nodes (both scenes and episodes) to their related super-graph. For example, material that is used to teach a particular lecture is also material to teach the complete course the lecture is part of.
2. In other cases it makes sense to inherit the arithmetic sum of a key annotation of all nodes to the related super-graph. For example, an upper limit of the time needed to teach a course can be estimated by the sum of its components (lectures) and a maximum cost of a university study can be estimated by the sum of the fees for all recommended subjects.
3. In other cases it makes sense to inherit the maximum value of a key annotation of all odes to the related super-graph. For example, the educational difficulty (basic/easy, medium, advanced, very difficult) of a study needs to be communicated as the maximum value of all mandatory subjects.

Thus, for each key annotation an appropriate inheritance method can be selected in our in our storyboard development environment [Xu 2006]. From a Knowledge Engineering point of view, inheritance in the storyboard hierarchy is some sort of deductive inference over the knowledge represented as storyboards.

3.3 Towards a Storyboard Development Environment

For an a priori approach to ensure such logical features, a set of operations were defined, which's exclusive use automatically leads to a “legal storyboard” [Sauerstein 2006].

These operations are (1) adding paths, (2) adding nodes, (3) turning a scene to an episode, while introducing a related sub-graph, (4) adding a concurrent path, and (5) merging (equivalent) nodes by introducing related bi-colored edges, which make sure that the linkage with the remaining graph isn't changed (see figure 1).

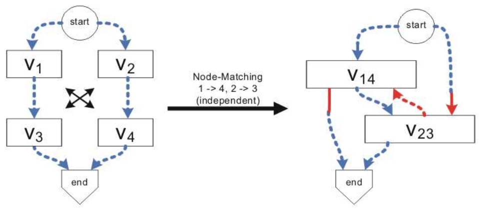

Fig. 1: Merging equivalent nodes

In figure 1, V_1 and V_3 as well as V_2 and V_4 are equivalent. Since different users visit them in different sequences, they are represented as different nodes on the left hand side. By merging the equivalent nodes together, a new color needs to be introduced to express these different sequences.

Based on this operation set we developed a web based storyboard development [Kasperski 2007]. By using it, the structure tests as listed up in section 3.1 are obsolete.

3.4 Knowledge Mining over Storyboard Paths

A basic objective of this storyboard application is to use Knowledge Engineering technologies on the (semi-) formal process models [This paper's author]. The objective of this research is inductively "learning" successful storyboard patterns and recommendable paths. This is performed by an analysis of the paths, where former students went through the storyboard and it is based on their success that is associated with these particular paths.

To exemplary show the feasibility and benefit of this approach, a simple prototype was recently developed to evaluate curricula created or modified by the students in advance of their study [This paper's author]. Here, we implemented a concept to estimate success chances of curricula, which are composed by students at a Japanese university in their curriculum planning class in the first semester.

Construction of a decision tree The construction of a decision tree is based on the paths of former students through the storyboards. Each of those paths can be associated with the degree of success, which has been achieved by the student. In case a set of students went the same path, the degree of success can be estimated by a weighted average degree.

This path begins at the Start Node of the top level storyboard and terminates at its End Node. Each episode on this path is replaced by its sub-graph. This replacement continues throughout the entire hierarchy of nested graphs. Figuratively speaking, the decision tree is constructed on the basis of a "flatten" storyboard, which contains atomic scenes only.

The decision tree is based on the concept of bundling common starting sequences of the various paths to a node of the tree. Different subsequent following (next) nodes of the paths will result in different sub-trees right below the actual root on the last node of the common starting sequence. This continues for each lower level sub-tree accordingly. If there are different paths with a common starting sequence from the root to the actual root different in the next (subsequent) nodes, related sub-trees will be established.

The final node of the paths are followed by a label-node, which contains a list of marks that students received after going through this path. Each mark is along with the number of occurrences (the number of students getting the mark).

Since the courses of a semester are usually visited concurrently, we consider them as a single node containing a set of courses.

A new path is added to the tree by simultaneously traversing the path's courses sequence and the decision tree down from the root until (1) the path is finished or (2) there is a "next node" in the path that is different from all "next sub-tree roots". In the first case, the related label node is updated accordingly. In the latter case, a new sub-tree is made out of the remaining path and hooked into the tree.

Path estimation and completion by a decision tree If a submitted path is completely represented in the decision tree (as a path from its root to a node that is succeeded by a label node, i.e. with an *assess* - fact), the success estimation is very easily done through presenting the content of this label. Otherwise, the most similar sub-path in the decision tree will be identified.

Like in the tree construction procedure, this is performed by simultaneously traversing the path's course sequence and the decision tree down from the root until (1) the path is finished or (2) there is a "next node" in the path that is different from all "next sub-tree roots". In the first case, the related *assess* - fact at its leaf position provides the desired success estimation. In the latter case, the *assess* - fact of the current tree position is provides the desired information.

Additionally, we provide a supplement to the submitted path, which is the most successful rest - path starting at the last node of the tree traversing along with this optimal achievable success.

Also, the user is informed about the degree of similarity of his submitted path and the one found in the decision tree. We call this similarity *significance* and compute it as the number of nodes in sequence that are common in the submitted path and the decision tree, related to the entire length of the path.

Based on this information, the user (student) can make a decision on whether or not holding on to the submitted curriculum or modifying it in accordance with the optimal supplemental path.

An example Figure 2 shows a concrete storyboard path, which has been went by a particular student, along with the result of the flattening procedure. This student finalized his study with a Grade Point Average of 3.0.

Figure 3 shows the result of the decision tree construction in our application. As illustrated in the figure's left hand side, 25 students went through the storyboard on four different paths, namely (1) $[s4, s6, s7, s1, s9]$, indicated by red background color, (2) $[s4, s6, s7, s5, s8]$, indicated by yellow background color, (3) $[s4, s2, s3, s1, s5, s9]$, indicated by green background color), and (4) $[s4, s2, s3, s1, s5, s6]$, indicated by blue background color.

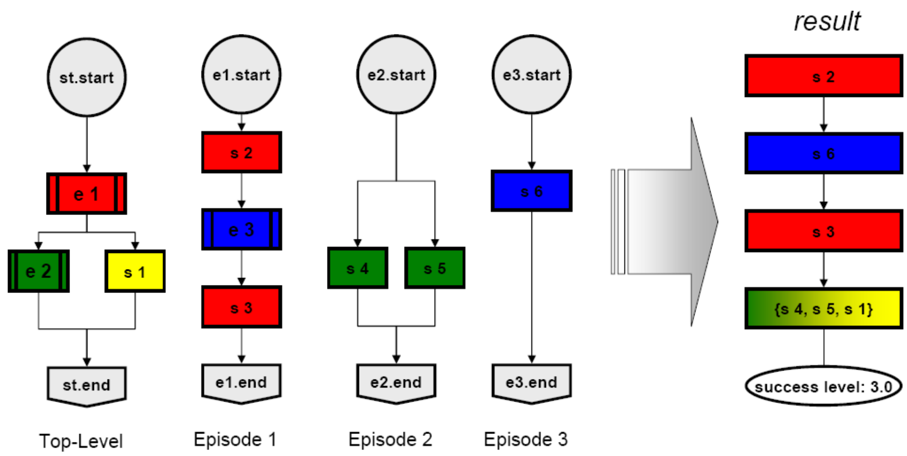

Fig. 2: A student's path through the (nested) storyboard

Figure 4 shows the usage of the decision tree for three submitted paths. For the blue path, for example, there is no identical path in the tree. Here, the estimation procedure looks for a path within the tree, which has the longest starting sequence in common with the submitted path. This is $[s4, \{s7, s6\}]$. Since this path has two nodes in common with the submitted one (having four nodes), the significance of the success estimation is calculated by $2/4 = 0.5$. Behind the node $\{s7, s6\}$, there are two different subtrees, which led to different success degrees by former students, $[s1, s9]$ and $[s5, s8]$. Since the latter is the better one, it is recommended as a rest path to optimize success chances.

By practicing this way to utilize a decision tree, we realized that we rarely found a path in the tree, which is completely equivalent to a submitted path, for which we wanted to have a success estimation. However, if an element of a node that contains a scene set in the tree is not in the related node of the submitted path, it still could be a subject that the student already passed successfully in a previous semester. If this applies to all set elements of a considered node in the tree that are not in the related node of the path, those nodes (the path node and the tree node) should be considered as "equivalent", too.

Therefore, the containment in the decision tree was extended with respect to the educational history of a student. A previously taken course may always be considered as an element of a subsequent node. Formally spoken, the new concept of path equivalence is as follows:

- Let $P = [P_1, P_2, \ldots, P_n]$ be a path submitted by a student with P_i being a set of courses taken in the i-th semester.
- Let $T = [T_1, T_2, \ldots, T_m]$ be a path that is represented in the decision tree.

P is represented by T ($P \sqsubseteq T$), iff all courses of all P_i are in any T_j with $j \leq i$: $\forall i \forall j \; P_i \subseteq \bigcup_{j=1}^{i} T_j$.

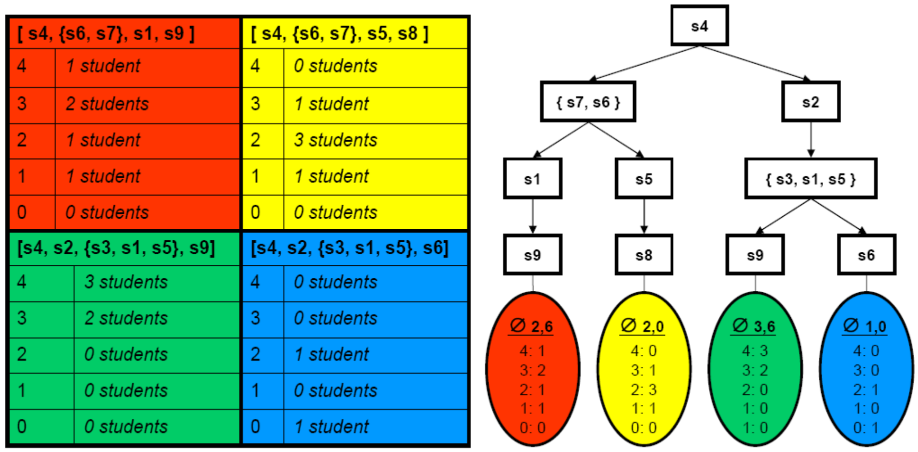

Fig. 3: Storyboard paths and a derived decision tree

4 Summary and Outlook

Storyboarding is a way of managing didactic knowledge for organizing learning experience. By storyboarding, didactic design became explicit and subject to evaluation and quality assurance:

1. Structure tests for verification are developed as a method to discover logical anomalies in storyboards.
2. An inheritance concept has been developed as a means of logical (deductive) inference over this knowledge representation.
3. Based on a set of operations that ensures logical correctness of storyboards, we developed a web based storyboard development environment for our storyboards.
4. As a first step towards inductive inference over this knowledge representation, we developed a method to estimate success chances of intended storyboard paths. Additionally, this approach also suggests a supplement to a given curriculum that leads to an optimum with respect to the success chances.

Our upcoming work focuses the following issues.

1. Storyboards have a high performance with respect to didactic issues of planning education processes. However, there is still no capability to manage these processes according to their resources (e.g. to concretely plan weekly timetables based on requests and available capacities like rooms, teachers, equipment and so on). Therefore, a desirable synergy effect is expected when incorporating the planning capabilities of the Dynamic Syllabus tool of the DLNRS [**?**] into the storyboards.
2. Most importantly, we learned that there is not the one and only one proper curriculum composition, which leads to success for every student. Students are different. A curriculum, which is good for one student may be bad for another one.

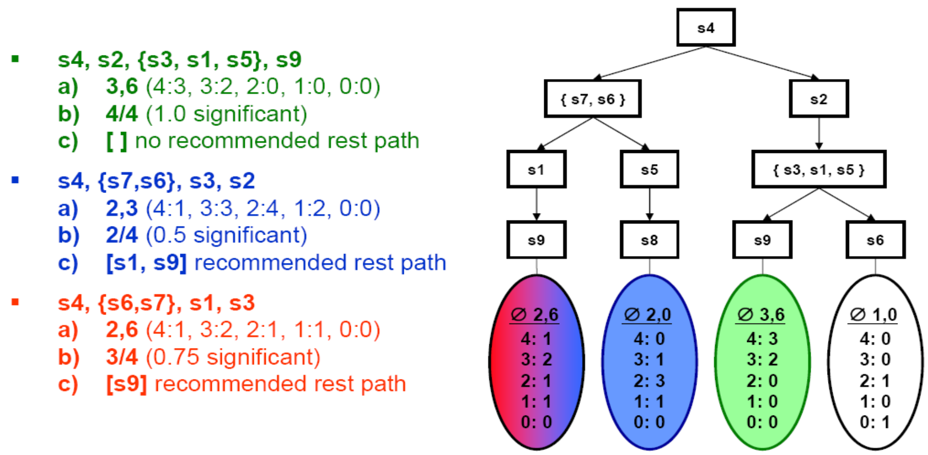

Fig. 4: Examples of (a) success estimation, (b) its significance, and (c) recommended rest paths

Consequently, we need to include meta-knowledge such as individual learning needs, learning desires, preferences and talents.
Such meta-knowledge may additionally be used for maintaining the university's educational resources in accordance to typical user profiles of current and future students as well as for a related organizational issues such as class schedules represented by storyboards according to the students' desires.

In fact, the above mentioned list of objectives and visions starts with items that are well done so far, but ends up with items that are hard to achieve and subject of much research left. In particular, the last two items are not touched at all so far, but they are our dream and ultimate goal.

References

[Boeck 2007] Boeck, R.: *Ein Data Mining Verfahren zur Pfadbewertung in Storyboards.* (German), Diploma Thesis, Technical University Ilmenau, Faculty of Computer Science and Automation, 2007.

[Chiang and Chung 2006] Chiang, F.-K. and Chung, S.: A Case Study on An Experienced Elementary Teacher's Teaching Strategies and Teaching Reflection in Mathematical Classroom. C. Crawford et al. (Eds.): *Proceedings of Society for Information Technology and Teacher Education International Conference 2006* Chesapeake, VA: AACE, pp. 3697-3703, 2006.

[Dohi et al. 2006a] Dohi, S., Nakamura, S., Sakurai, Y., Tsuruta, S., and Knauf, R.: Dynamic Learning Need Reflection System for academic education and its applicability to Intelligent Agents. *Proc. of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006)*, Kerkrade, the Netherlands, Los Alamitos, CA: IEEE Computer Society, ISBN 0-7695-2632-2, pp. 459-463, 2006.

[Duesel 2007] Duesel, H.: *Konzeption und Realisierung von Methoden der formalen Verifikation von Storyboards* (German), Diploma Thesis, Ilmenau University of Technology, Faculty of Economic Sciences, 2007.

[Kasperski 2007] Kasperski, S.: *Entwicklung eine unabhaengigen Storyboardrepraesentation* (German), Studienarbeit, Ilmenau University of Technology, Faculty of Computer Science and Automation, Chair of Artificial Intelligence, 2007.

[This paper's author] This paper's author: a recent self reference, 2008.

[Sauerstein 2006] Sauerstein, G.: *Formale Modellierung und daduktive Interpretation von Storyboards. Modellierung eines komplexen Prozessmodells* (German), Studienarbeit, Ilmenau University of Technology, Faculty of Computer Science and Automation, Chair of Artificial Intelligence, 2006.

[Xu 2006] Xu, G.: *Ein Vererbungskonzept fuer Storyboards* (German), Studienarbeit, University of Ilmenau, Faculty of Computer Science and Automation, Chair of Artificial Intelligence, 2006.