Graph-based Modeling for Adaptive Control in Assistance Systems

Alexander Streicher[1], Rainer Schönbein[1], Stefan Pickl[2]

[1] Fraunhofer IOSB, Karlsruhe, Germany
{alexander.streicher, rainer.schoenbein}@iosb.fraunhofer.de
[2] Universität der Bundeswehr, München, Germany
stefan.pickl@unibw.de

Abstract. The topic of this contribution is characterization and analysis of assistance systems in order to enable adaptivity, i.e., as personalized adaptive systems. The research question of this article is how to facilitate the modeling efforts in adaptive e-learning assistance systems. Adaptivity here means to personalize the usage experience to the individual needs of the users and their current working context. For that, adaptive systems need usage models and user models. The problem statement is that expert knowledge and recurrent efforts are needed to create and update these types of models. Data driven and graph analytics approaches can help here, in particular when looking at standardized interaction data and models which encode sequences such as interaction paths or learning paths. This article studies how to make use of interaction usage data to create sequence-typed domain and user models for their use in adaptive assistance systems. The main contribution of this work is an innovative concept and implementation framework to dynamically create Ideal Paths Models (IPM) as reference models for adaptive control in adaptive assistance systems.

Keywords: Adaptivity · Adaptive Control · Graph Analytics · Modeling

1 Introduction

Assistance systems can support users to achieve their tasks [1, 2]. An intelligent assistance system observes the users' interactions and automatically adapts to the users' needs and their working context [2]. This is, it could change the way the users can interact with the system, or by providing context-sensitive support, e.g., context-related recommendations such as learning help in intelligent tutoring systems [1, 3].

This paper presents how to characterize and analyze assistance systems to enable adaptivity, hence forming personalized adaptive systems. The research question of this work is how to facilitate the modeling efforts in adaptive e-learning assistance systems: adaptivity in this article means to personalize the usage experience to the individual needs of a user and his current working context. For example, in adaptive e-learning the systems can provide the users with adaptive guidance or make dynamic difficulty adjustments, i.e., making it easier or more challenging. Adaptive guidance could be recommendations on the next best activity following (individual) learning paths [2]. However, adaptivity components in personalized assistance systems need user and domain models to determine the next best course of action. User models typically contain information on how, when and what a user has interacted with, or – for a cognitive user model – information on the cognitive state or load (e.g., stress level). The domain models can be interaction paths or, in e-learning, assessment questions or learning paths [2].

The problem statement is that expert knowledge and recurrent efforts are needed to create and update such kind of models. Data driven and graph analytics approaches may help here, especially when looking at interaction data and models which encode sequences such as usage paths. This article studies how to make use of interaction usage data to create sequence-typed domain and user models for their use in adaptive assistance systems. These models contain information on the general usage of the attached systems (domain model) as well as the individual interactions of single users (user model). The concrete research questions studies to create dynamically adjustable and flexible user and usage path models, and how to analyze the models for their application in adaptive control. This contributes to one of the main questions in adaptive assistance: when to actually adapt.

The contribution of this work is a concept and implementation framework to dynamically create Ideal Paths Models (Fig. 1) [4] as reference models for adaptive control in adaptive assistance systems. Our graph-based modeling approach links the IPM to domain and user models. The concept shows how to define graph-based IPMs, and how to apply them to quantify user performance. The framework presents a software architecture which uses standard-based activity stream tracking data to generate IPMs.

Our field of application is assistance systems for education and training, in particular e-learning for image interpretation [4]. Our adaptivity approach tries to keep the users immersed in gamified interactive learning environment (so called serious games [5]) by keeping them in the so-called Flow channel [6], balanced between the perceived skills and challenges.

This article concentrates on the modeling aspects of usage pathways which encode which sequence of actions users undertook [7]. An additional challenge is to align the modeling with established standard models to provide a solid basis for the usage paths, learning paths [7], learning goals [2], or learning performance [8]. For example, it must be possible to model and compute learning progress. The literature review points to established domain and user modeling approaches from the field of Intelligent Tutoring Systems (ITS), as described by Woolf [2] or with focus on user modeling by Kurup et al. [9]. All in common is the obvious separation of the usage pathways into atomic or logical coherent elements. In the context of this article we see modeling approaches such as the Knowledge Tracing (KT) model with Competence-based Knowledge Space Theory (CbKST) [10]. In comparison, for KT the usage pathways need to be predefined into states or knowledge components and the transitions between them, plus additional models on learning and the alignment to competencies. We adopt this modeling approach but our concept uses the observed interactions to construct the models' base usage pathways layer. To quantify performance and determine the estimated level of needed assistance we adopt the Performance Factors Analysis (PFA) logistic regression model [11]. The PFA builds upon the KT modeling approach and uses observed success and failure states to compute performance scores.

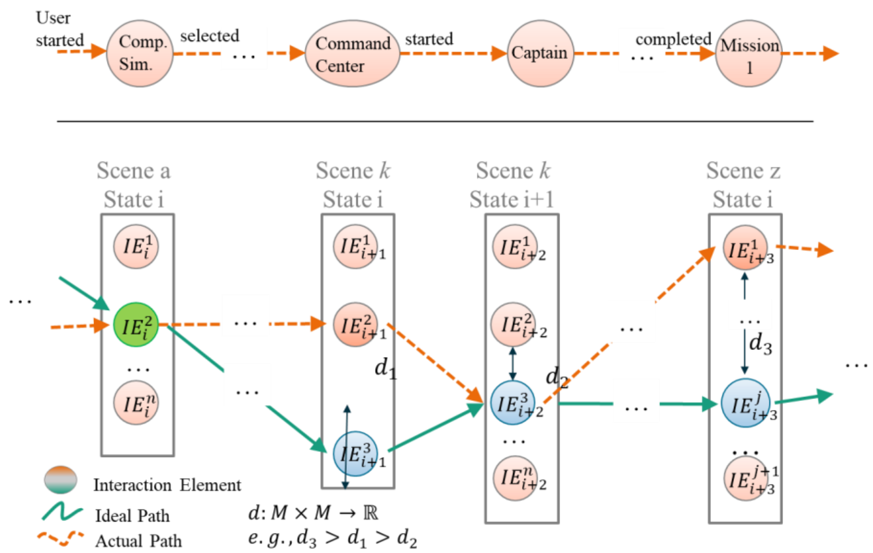

Fig. 1. Example for usage pathways (top) and general Ideal Paths Model concept [4].

2 Adaptive Control and Pathways Modeling

Various adaptive assistance systems exist [1, 3, 8], and their underlying principle follows the concept of control systems theory, here adaptive control [12]. General (linear) closed-loop feedback control systems react to observations or measurements to modify the controlled system or plant. With human behavior in the loop, we typically deal with complex, nonlinear processes which motivates the use of adaptive control theory [12]. Central to this – and central to this work – is the concept of Model Reference Adaptive Control (MSAR) [12, 13], i.e., using reference models to more broadly and informed react (or adapt) to parameter changes. For instance, these reference models can provide information on the corrects usage pathways which the users should follow, or to quantify their performance by computing a metric as deviation from a targeted pathway. The quantification aspect is of high importance for adaptive systems since they need to know when to actually adapt and in which direction. However, building such reference models is typically done manually and expert or domain knowledge is needed [2].

Our view on adaptive systems follows the 4-phased adaptivity cycle by Shute et al. [3]. It structures an adaptivity process in four consecutive phases or stages where each new run depends upon the previous run, hence forming a cycle. Its main components are the four phases (1) capture, (2) analysis, (3) select and (4) present, plus an additional user or learner model after the analysis phase. However, we incorporate the user models into the analysis phase (2) [14]. The argumentation is that the select phase (3) not only builds upon and uses the user models but also incorporates additional analysis results, such as usage pathways models.

We define a usage pathway (Fig. 1) as the sequence of user interactions with a system. For e-learning, a special form is a learning pathways model. Typically, these models are pre-defined sequences of usage patterns within an e-learning system [7]. Learning pathways are a crucial element for adaptive e-learning systems since they provide information how learning courses are structured, how to determine if the users are on track, and to estimate the learning progress (*cf.* previous section). Without loss of generality this is also valid for assistance systems in general. Modeling of these pathways typically follows a standard directed graph model $G = (V, E)$ with vertices or nodes V and edges E. Since our modeling is based on observations our graph is directed with a linear ordering in the sequence of user interactions. We can differentiate between predefined or offline usage pathways and effective usage or online pathways. The former is typically defined in the design and implementation phase of an assistance system [1, 2]; the latter, effective pathways, are dynamically build at runtime. Two cases of usage pathways adaption can be distinguished: macro and micro pathways adaption.

Macro adaption looks at the whole pathways. An adaptive system would offer the users recommendations on suitable learning paths, or it would modify the navigation in such a way that for the continuing system usage suitable pathways are selected.

Micro adaptation works on individual elements inside of pathways, i.e., it works on the nodes. Therefore, the recommendations or adaptations are more immediate. In this work the adaptation model uses the micro level.

For our modeling approach there is no restriction on the granularity. Better said, it follows the level of detail of the observations. If the observations are at a very high, abstract level, then the resulting graph contains only a few nodes, and vice versa. Typically, assistance systems do not record all possible smallest events in a fine granular way (e.g., all mouse movements), but follow the systems' logical structure (e.g., windows, scenes) and the events on the user-interaction elements (e.g., buttons). Basically, the filtering of which events to observe reflects the decision of what type of adaptivity should occur. For example, if adaptivity is to guide the user only at the macro level, then only the beginning and end of a user session might be required.

In our e-learning application context the important aspect of didactic modeling is not yet made explicit. While the data-driven usage pathways detection hint at how users navigate through the content it does not directly reveal the didactic model of the e-learning system. Model knowledge of learning pathway levels and sequencing of content would help in recommending next learning objects. Learning path levels could be: (1) sequence of (learning) courses; (2) sequence of chapters or missions; (3) sequence of subchapters; (4) sequence of knowledge units (individual scenes, web pages, etc.); (5) interaction sequence within a knowledge unit, e.g., factual knowledge before action knowledge before source knowledge.

3 Data-Driven Modeling for Flexible Adaption

The idea is to use the observations from standardized tracking data to create graph-based common paths models. These common paths are needed for our concept of the Ideal Paths Models (IPM) (Fig. 1) [4] as reference model for adaptivity. The IPM describes all necessary steps to achieve the objective without unnecessary detours. Essentially, it is a sequence of episodes and interactions that leads most directly to the next goal. For instance, in a computer simulation for reconnaissance training users' first select those interaction elements which lead them to a virtual command center where they are briefed on their first mission (Fig. 1). A scene can have multiple manifestations for each possible user interaction. These interactions are observed or tracked in an assistance system, typically in the form of activity streams, e.g., "John has completed reconnaissance mission 1". To allow general applicability we propose to use the W3C Activity Streams standard which encodes usage interaction events in triple form with <actor, verb, object>. In the e-learning domain this has been adopted to the Experience API (xAPI) standard following the same triple principle. This observation data can be recorded in a graph-database to make use of graph or social network algorithms [14, 15]. For instance, to find most or least active users or learning objects, or to compute the shortest path between nodes or subset of nodes. Graph pattern analysis [15] can find individual learning paths or, for several users, common learning path models. The results or outputs of these analysis processes are, for example, learning path models or ideal path models, which again can serve as inputs in further downstream processes in the formation of learning path models. Thus, the observation data (capture phase) can also be used for data-driven model building, which allows a much more flexible application, because the application-specific data models can "grow naturally", schema-free.

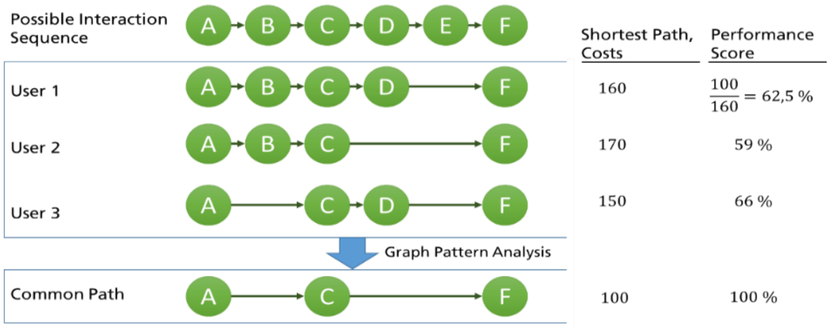

Fig. 2. Concept to find common path and compute performance scores based on possible interaction sequences, individual user paths and application of shortest path graph algorithm.

One example using graph pattern analysis and a shortest path algorithm is depicted in Fig. 1 3. Given is a possible interaction sequence $P = \{A, B, C, D, E, F\}$ with the individual interaction elements $A...F$. Different interaction paths are observed for different users 1-3. As stated, before the observations are captured using xAPI whereas the xAPI objects reflect the interaction elements (Fig. 1), and they are stored as nodes in the graph. Using graph pattern matching, e.g., string similarity, a common subset of nodes $C \subseteq P$ between all user paths can be determined. The result is a common path, e.g., $C = \{A, C, F\}$ (Fig. 2). One approach to pattern analysis and to find the common path is to encode the users' pathway graphs P_j (for user j) as binary adjacency matrices. Multiplication of these matrices yields the nodes which are common in all graphs, i.e., nodes on the common path. The next step is to quantify the individual user paths to determine the individual user performance. In our graph-based usage pathway model the transitions (or relationships) are assigned non-negative weights forming a weighted graph. This makes it suitable for applying path finding algorithms such as Yen's k-shortest path [15]. The concept is depicted in Fig. 2 on the right: for each user compute the shortest path and select that path or sub-graph P_j^* with the lowest cost c_{min}. These costs are then normalized using the costs $\dot{c}$ of the common path as reference (the common path must always have the lowest cost), i.e., $p_j = \frac{\dot{c}}{c_{min}}$ (Fig. 2). In the context of the IPM the found common path is the basis for the "ideal paths". Since ideality is subjective to the individual user and his personal usage (or learning) goal, the common path is only the base for the user's own other pathways. This performance score tells how near a user is on an ideal path. In an adaptive system this score could be used to determine the point in time when adaptivity should be enabled, e.g., by thresholding on window-based aggregation of scores.

4 Application Example

We have conducted experiments with xAPI and graphs, implemented for adaptive assistance in serious games. The results indicate that storing usage data as graph structures indeed brings advantages for modeling (flexibility) and analysis (existing algorithms). This is, multiple assistance systems have been equipped easily with xAPI trackers, and for the modeling process a schema-free, NoSQL graph database (e.g., Neo4j) helped to stay flexible, e.g., for new application domains. These kinds of databases also allow to directly apply graph algorithms such as Social Network Analysis, e.g., to determine frequently occurring (learning) paths, shortest paths, or to determine frequent or rare activities. A real-life example of such graphs is depicted in Fig. 3.

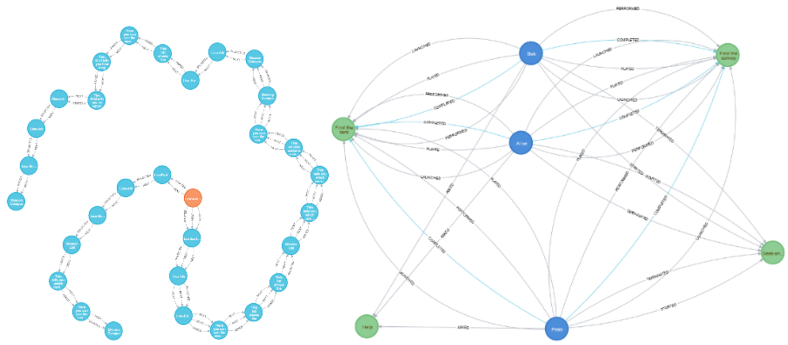

Fig. 3. Real usage graphs, from xAPI observations. Nodes are actors and activities; edges are the verbs. Left: interaction sequence for one user. Right: interactions graph for multiple users.

The selection of tracked events directly corresponds to the level of the targeted adaptivity level, i.e., macro- or micro-adaptivity (*cf.* section 2). For micro-adaptivity in our serious game application scenario we observe each interactive element in a scene. The outcome is that the adaptivity system can pinpoint the current state of a user within a session by comparing that to the common path reference model. Because of the cold-start problem the adaptive assistance system needs data from multiple user sessions to build valid pathway models.

Neo4j's graph algorithms can be applied directly to the xAPI based graphs. In our current implementation the graphs have identical edge weights (cost=1); varying weights and stochastic graphs are planned for the future. After collecting the xAPI data from multiple user sessions we get all nodes and transitions on the possible interaction sequence (PIS) sub-graph (Fig 4, top). The minimal PIS from start to end can be found by using path finding algorithms such as Yen's k-shortest path [15] ($k = 1$). The final step is to find those nodes which are on the common path but not the users' currently observed usage path (e.g., string edit distances). The adaptive system can use the information from the next estimated node (e.g., metadata such as the activity name or the name of the transition) to issue a hint to the user or to modify the system's navigational path, i.e., allowing only to interact with the next estimated activity. In our application scenario we choose the adaptivity strategy based on the performance score and an additional assistance level (based on other features, e.g., cognitive load [16]).

5 Conclusion

Adaptive systems need reference models to determine the correct timing and the direction how the automatic adaptation should happen. The underlying principle follows adaptive control from systems theory, i.e., in closed-loop feedback systems [12].

A key aspect in technical control systems is to measure the current state and derive some feedback. Assistance systems that follow that principle can quantify the deviations from ideal paths by computing a distance metric between the current interaction sequence and (pre-defined) usage pathways. However, the construction of these usage pathway models requires domain and expert knowledge.

This contribution addresses the data-driven, graph-based generation of such kind of models. The presented approach makes use of standardized triple-structured tracking data which is the input to the model generation process.

In our application of e-learning assistance systems this is Experience API (xAPI) data. However, as xAPI is related to the more general W3C Activity Streams standard the approach is not restricted to the e-learning domain. By applying graph algorithms, we extract common paths and Ideal Paths Models which can act as reference models for adaptivity systems. Future work will further deepen the transfer to the specific e-learning domain as well as to assistance systems in general. In the case of the former, the modeling must take didactic models and user model more into account.

Acknowledgments. The underlying project to this article is funded by the Federal Office of Bundeswehr Equipment, Information Technology and In-Service Support under promotional references.

References

1. Gorecky, D., Schmitt, M., Loskyll, M., Zühlke, D.: Human-machine-interaction in the industry 4.0 era. In: 2014 12th IEEE International Conference on Industrial Informatics (INDIN). pp. 289–294 (2014). https://doi.org/10/ggz7vg.
2. Woolf, B.P.: Building Intelligent Interactive Tutors. Morgan Kaufmann (2009).

3. Shute, V., Zapata-Rivera, D.: Adaptive educational systems. Adaptive technologies for training and education. 7, 1–35 (2012). https://doi.org/10.1017/CBO9781139049580.004.
4. Streicher, A., Leidig, S., Roller, W.: Eye-Tracking for User Attention Evaluation in Adaptive Serious Games. In: 13th European Conference on Technology Enhanced Learning, EC-TEL 2018. pp. 583–586. Springer, Leeds, UK (2018).
5. Dörner, R., Göbel, S., Effelsberg, W., Wiemeyer, J. eds: Serious Games - Foundations, Concepts and Practice. Springer International Publishing, Cham (2016).
6. Nakamura, J., Csikszentmihalyi, M.: The Concept of Flow. In: Flow and the Foundations of Positive Psychology: The Collected Works of Mihaly Csikszentmihalyi. pp. 239–263. Springer (2014). https://doi.org/10.1007/978-94-017-9088-8_16.
7. Streicher, A., Heberle, F.: Learning Progress and Learning Pathways. In: Fuchs, K. and Henning, P.A. (eds.) Computer-Driven Instructional Design with INTUITEL. pp. 37–55. River Publishers Series (2017).
8. Conati, C., Manske, M.: Evaluating adaptive feedback in an educational computer game. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 146–158 (2009).
9. Kurup, L.D., Joshi, A., Shekhokar, N.: A review on student modeling approaches in ITS. In: Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016 (2016).
10. Melero, J., El-Kechaï, N., Labat, J.-M.: Comparing Two CbKST Approaches for Adapting Learning Paths in Serious Games. In: Conole, G., Klobučar, T., Rensing, C., Konert, J., and Lavoué, E. (eds.) Design for Teaching and Learning in a Networked World. pp. 211–224. Springer International Publishing, Cham (2015)..
11. Pavlik, P.I., Cen, H., Kenneth R. Koedinger: Performance factors analysis – a new alternative to knowledge tracing. In: 14th Intl. Conf. on Artificial Intelligence in Education 2009 (2009).
12. Åström, K.J., Wittenmark, B.: Adaptive control: second edition. Dover Publications (2013).
13. Nguyen, N.T.: Model-Reference Adaptive Control: A Primer. Springer (2018).
14. Streicher, A., Pickl, S.W.: Characterization and Analysis with xAPI based Graphs for Adaptive Interactive Learning Environments. In: WMSCI. , Orlando, FL, USA (2020).
15. Needham, M., Hodler, A.E.: Graph Algorithms: Practical Examples in Apache Spark and Neo4j. O'Reilly Media (2019).
16. Aydinbas, M.: Realizing Cognitive User Models for Adaptive Serious Games, (2019).