CHAPTER 7 – Design and Construction of Domain Models

Andrew M. Olney and Arthur Graesser
University of Memphis

Introduction

The problem of domain modeling is to represent domain content so that it can be efficiently authored, optimally delivered to students, and precisely tracked with respect to student mastery. While the previous section focused primarily on the representation aspect of domain modeling, the present section focuses on various methods and concerns related to authoring, delivery, and mastery.

An overarching method that unites most of the chapters in the present section is a data-driven approach to domain modeling that uses machine learning. This is consistent with the current zeitgeist of Big Data, enabled by Internet-scale data sets and cloud-computing resources. With these data, researchers are able to parameterize increasingly complex models from data, perform model selection on alternatives of such models, and even author content using crowdsourcing or semi-supervised machine learning.

Data-driven approaches are just one part of the story, and if we look behind the curtain, we see that the following chapters are deeply nuanced in their treatment of domain models. Each chapter raises issues and concerns not only for what domain models currently are but also what they could be. This section provides an excellent overview of domain modeling and a guide to future research. For the purposes of this introduction, we focus on three major themes arising from this work: bringing external information into domain models, mapping items to skills, and creating domain models.

Theme 1: Bringing External Information into Domain Models

The chapter by Brawner, Goodwin, and Regan discusses the problem of hybrid systems where training is not bounded by a computer based environment but instead is also distributed across human based training. Ideally, the Generalized Intelligent Framework for Tutoring (GIFT) should be able to know about training delivered outside of itself in order to best adapt the training it delivers. This external training information could be longitudinal (in the sense of slowly acquired mastery) or immediate (in the sense of a lecture delivered by an instructor just prior to interaction with GIFT). Sharing information can be accomplished in three ways. First, sharing can use a high-level competency framework that is loosely coupled with actual skills. This essentially is a shared domain model at a high level, analogous to an upper level ontology. Second, sharing can use a low level approach where the set of problems and student responses is shared, and each system is responsible for inferring the competencies in which it is interested from this interaction data. Thirdly, machine learning could be used to unify these two levels to model what actions best predict competencies at the organization level. That is, the learning records of various systems are forwarded to the organization level system (e.g., Army-wide) for inference and matching to high level competencies.

The chapter by Sinatra shares this concern in the sense of wanting to use system-external information to adapt to the student, but the emphasis is on interest- and motivational-based enhancement of memory rather than adaptive problem selection or mastery. In other words, the focus is less about what to teach and more about how to teach. One technique is the Self-Reference Effect (SRE), a way of enhancing recall, retention, and motivation across many domains and age groups. The SRE can be created simply by changing the pronouns used in the delivery of instruction to reference the student (“your respiratory system”) rather than the typical description in which the student is not a participant (“the respiratory system”). Similar to SRE, context personalization, in which the interests of the students are incorporated into the material, can increase problem-solving performance, knowledge transfer to new problems, and positive attitudes about the material. These effects could be realized in GIFT by (1) authoring guidelines to enhance SRE, (2) adding user interest surveys to populate templated GIFT materials, such that interests could be substituted for topics as needed, or (3) supporting a diversity of interest-aligned content such that once interests are established, GIFT could preferentially select content matching those interests.

Theme 2: Mapping Items to Skills

The chapter by Goldin, Pavlik, and Ritter is concerned with mastery of particular knowledge components (KCs) and also defines domain models primarily in Q-matrix terms. Like the Desmarais and Xu chapter in the previous section, the focus is on refining the Q-matrix, but unlike the previous chapter, the objective is not to re-parameterize an initial Q-matrix (e.g., by changing the value in a particular cell) but to instead refine the structure of the Q-matrix itself by collapsing or splitting existing KCs and their corresponding matrix columns. The approach is based on learning curves, which are models of the error rate for a population of students as they attempt to learn a KC. Given a set of alternative domain models and associated learning curves, one can rank and select the domain model that best represents the learning curve data. The major focus of this chapter is on the qualitative depiction and analysis of learning curves for the refinement of Q-matrices.

Theme 3: Creating Domain Models

The chapter by Williams, Kim, Glassman, Rafferty, and Lasecki represents an emerging area of domain model authoring that relies on crowdsourcing. The goal of crowdsourcing is to shift the burden of authoring domain models from a handful of experts to capable students or novices. One of the principal problems in this approach is maintaining quality. This chapter focuses not on the creation of intelligent tutoring systems (ITSs) wholesale but instead on the addition of new content to an existing system. After solving math problems, students contribute explanations of their solutions that are rated by other students, and then machine learning is used to select the best explanations. Results show that students both learn more and that explanations (as ranked by the machine learning algorithm) improve over time. By focusing on self-explanations, this technique appears to be fairly generalizable and therefore useful for systems implemented in GIFT.

The chapter by Barnes, Mostafavi, and Eagle presents interaction networks, which are data-driven, problem-specific domain models based on student actions during problem solving. Interaction networks can be used to mark the correctness of student actions, model student ability, and adapt delivery of instruction. Interaction networks are particularly suited to state-space domains (like algebra and logic) where student actions can be viewed as edges between nodes representing states. Given even a small amount of data to characterize the solution space, interaction networks can be used to generate next-step hints. By clustering student action sequences in the interaction network, the problem-solving behaviors of groups of students can be grouped not only into successful versus unsuccessful approaches but also different successful approaches to the same problem. These methods can be extended to implement data-driven knowledge tracing and associated problem selection. This approach could be used in GIFT for state-space domains where a complete system exists (necessary to collect the student interaction data) but has no associated pedagogy.

The chapter by Dargue, Pokorny, and Biddle presents a specific variant of cognitive task analysis, called Precursor, Action, Results, Interpretation (PARI), for the purpose of building domain models based on expert mental models. PARI proceeds with a problem-solving dyad of subject-matter experts (SMEs), one of whom simulates outcomes (e.g., equipment responses) to the SME engaged in problem solving. Using a set of pre-selected tasks to structure these interviews, PARI records the problem-solving process, step by step, in terms of four pieces of data: action precursor, action, result, and result interpretation. While the first two pieces of data may be considered as part of a production rule, the last two pieces correspond to the revision of a mental model given a result. In several respects, this can be viewed as the manual approach to the problem that the previous chapter tries to automate. However, this manual process could be used to author a domain model directly from experts rather than needing a system to collect data for data-driven methods to be applied. Because cognitive task analysis approaches are the most comprehensive, they continue to represent the gold standard for domain model authoring.

Implications for GIFT

These chapters provide guidance for the continued development of GIFT by illustrating what domain modeling methods are valued in the community and what impacts these methods have in past and ongoing research. Briefly stated these are as follow:

- What methods can be used for sharing domain modeling data across systems such that mastery can be tracked across systems and instructional adaptation can occur across systems?
- How can GIFT represent item-to-knowledge mappings (e.g., Q-matrices) in a flexible way that allows the Q-matrices to be restructured/re-parameterized as needed?
- How can GIFT support domain model authoring across the lifecycle, stretching from initial expert interviews, to data-driven refinement from use, to crowdsourced authoring of additional content?