

RESEARCH

Open Access



# A possible future for next generation adaptive learning systems

Alfred Essa

Correspondence:  
alfred.essa@mheducation.com  
McGraw-Hill Education, 281  
Summer St., Boston, USA

## Abstract

Recent advances in big data, learning analytics, and scalable architectures present new opportunities to redesign adaptive learning systems. This paper is part directional and part speculative. We sketch a possible future for designing next generation adaptive learning systems based on new developments in learning science and data science.

**Keywords:** Adaptive learning, Intelligent tutoring systems, Learning analytics, Lambda architecture, Learner model, Knowledge space theory, Item response theory, Learning science, Data science

## Introduction

Research studies consistently show that students achieve significant learning gains when using adaptive systems, which includes intelligent tutoring systems (ITS) (Dodds and Fletcher 2004; Kulik and Kulik 1991; Durlach and Ray 2011; Ritter et al. 2007; VanLehn 2011). However, “despite their demonstrated value over a thirty-year history, the use of ITS remains restricted to research projects and a few commercial applications” (Robson and Barr 2013). The vantage point of this paper is perched between applied research and product development. The goal is to distill key design principles for creating next generation adaptive learning systems.

A next generation adaptive learning system should minimally have seven characteristics. The system should be:

1. *cost-effective* to build, maintain, and support;
2. *accurate* in its assessment of learner characteristics and learner knowledge state;
3. *efficient* in carrying out decisions and recommendations, such as identifying optimal instructional resources and activities for each learner at each moment in time;
4. able to *scale* to support hundreds of thousands, if not millions, of simultaneous users;
5. *flexible* in being able to integrate with enterprise systems based on open standards;
6. *generalizable* to domains beyond Science, Technology, Engineering, and Mathematics (STEM) disciplines;
7. able to support *transparent* open learner models to encourage learners to take greater control and responsibility of their own learning.

‘Teaching machines’ section sketches an *informal framework* for understanding the structural characteristics of adaptive learning systems. The framework is based on Richard D. Smallwood’s pioneering study of “teaching machines” (Smallwood 1962).

‘Formal framework’ section specifies a *formal framework* for capturing and representing this structure. Despite the diversity of adaptive systems most utilize the same set of foundational design principles.

‘Intelligent tutoring systems’ section leverages a key distinction (“inner loop” vs “outer loop”) in ITS in order to discuss two major senses of adaptivity in learning systems.

‘Deep learner models’ section introduces the concept of a deep learner model, which is a 360° perspective of each learner based on multiple dimensions including cognition, affect, motivation, and meta-cognition. It is suggested that deep learner models will become the vehicle for incorporating theoretical and practical advances in learning science into adaptive learning systems.

‘Learning objects’ section defines the concept of modular learning objects as flexible building blocks for adaptive systems. As we will make clear, learning objects are “pedagogical” atoms and should not be confused with mere learning assets or learning resources. Modular learning objects, within the context of scalable systems, have the potential for unlocking the yet unrealized benefits of Open Educational Resources (OER).

‘Advanced learning analytics’ section discusses the role of data and advanced learning analytics. Advanced analytics can strengthen two functions in adaptive systems: a) the ability to surface actionable insights and b) to establish a feedback loop for iteratively improving the quality of adaptive models.

Finally, ‘Big data architecture for adaptive learning systems’ section describes a scalable, cloud-based learning analytics platform which runs generalized adaptive and analytical models on educational data in parallel. The architecture also allows distributed systems to exchange data based on open standards.

## Teaching machines

In his monograph “A Decision Structure for Teaching Machines” Smallwood (1962) gives one of the earliest and clearest statements of the advantages of automated instruction based on adaptive principles.<sup>1</sup> Smallwood first states some primary properties of a “teaching machine”:

1. *Each student proceeds at his own individual pace.*
2. *By answering questions at the end of each block, a student masters the information in a block before going to the next block.*
3. *The student finds out immediately whether or not he has answered a question correctly and so is able to correct any false impressions at once.*
4. *Complete records of student’s performance on the teaching machine program are available, so that improvements can be made in the program itself* (Smallwood 1962, p. 2-3).

The first three properties embody the concept of mastery learning, which Bloom (1968) and others (Carroll 1963; Keller 1968) have shown through research to be a superior mode of instruction compared to traditional classroom lecture.

“The core idea (of mastery learning) is that virtually all students can achieve expertise in a domain if two conditions are met: (1) the domain knowledge is appropriately analyzed into a hierarchy of component skills and (2) learning experiences are structured to ensure that students master prerequisite skills before tackling higher level skills in the hierarchy” (Corbett and Anderson 1995, p.253).

The fourth property anticipates embedded analytics, or the continuous collection of data to update and improve the quality of learning models.

The fundamental “desirable” property of a teaching machine, however, is that it be able to *vary* its presentation of learning material based on the individual characteristics and capacities of each learner.

“This adaptability requires that the device be capable of branching — in fact, one would expect the *potential* adaptability of a teaching machine be proportional to its branching capability. In order to accommodate a high branching capability in the class of teaching machines discussed here, a large network of information blocks is assumed to exist *for each concept* to be taught” (Smallwood 1962, p. 11).

After describing some basic properties of a teaching machine Smallwood states a set of structural principles. In the next section we will map each principle below to a set of formal models (for each principle the corresponding model is stated in parentheses):

1. *The decomposition of the subject matter into a set of concepts that the educator would like to teach the student. (domain model)*
2. *A set of questions, for each concept, that adequately tests the students understanding of the concept. (assessment model)*
3. *An array of information blocks, for each concept that can be presented to the student in some order (to be decided by the teaching machine) — and thus provide a course of instruction to the student on the concept. (pedagogical model)*
4. *A model that can be used to estimate the probability that a given student with a particular past history will respond to a given block or test question with a particular answer. (learner model)*
5. *A decision criterion upon which to base the decisions mentioned in (3). (transition model)*(Smallwood 1962, p.27)

### Formal framework

In this section we take Smallwood’s characterization of a thinking machine as a starting point and present a formal framework that incorporates modern developments. Our goal in this section is to derive a *formal, structural* understanding of adaptive learning systems.

At a high level of generality all adaptive educational systems rely on five interacting models.<sup>2</sup> The *domain model* specifies what the learner needs to know. We will also refer to the domain model as the knowledge space. The *learner model* represents what the learner currently knows in the knowledge space. We will also refer to the learner model as the learner’s knowledge state. The *assessment model* is how we infer a learner’s knowledge state, typically through assessment probes. The *pedagogical model* specifies the activities to be performed by the learner to attain the next knowledge state. The pedagogical model can encompass a wide range of activities, ranging from watching a video to engaging in

a collaborative exercise with peer learners.<sup>3</sup> The *transition model* determines what the learner is ready to learn next.

$\Delta$  = Domain model (knowledge space)

$\Lambda$  = Learner model (knowledge state)

$A$  = Assessment model

$\Pi$  = Pedagogical model

$\Theta$  = Transition model

### Domain model

We begin with the domain model or what the learner needs to know. The domain model can be represented initially as a set of concepts, knowledge components, or knowledge units (KU). Each KU can be seen as an “elementary fragment of knowledge for the given domain” (Brusilovsky 2012).

“Every educator, whether he is a teacher or a writer of teaching machine programs, must necessarily have a set of goals — a list of things that he is trying to teach his students. We shall call these ‘concepts’, although a very broad definition of the word ‘concept’ is intended” (Smallwood 1962, p. 10).

We will represent the knowledge domain  $\Delta$  and its corresponding set of knowledge units  $\delta$  (or concepts) as:

$$\Delta = \{\delta_1, \delta_2, \delta_3, \dots, \delta_n\} \quad (1)$$

In the simplest case the domain model consists only of the set of knowledge units. Practically, however, most adaptive systems impose a set of implication relations or links among the knowledge units. The intuition behind the implication relation is that learning occurs in some sequence.

“This structure of implications among KU is determined by the order in which we learn concepts, or acquire competencies, and it constitutes one of the most important characteristics of the general learning process” (Desmarais et al. 1995).

The work of Doignon and Falmagne in knowledge space theory (KST) represents one of the earliest and most rigorous attempts to formalize the structure of a knowledge domain for the purposes of adaptivity (Doignon and Falmagne 1985). It will serve as the paradigm for what we call the domain model.

In KST a knowledge space defines the knowledge that a learner needs to master. The knowledge space is decomposed first into a set of KUs.<sup>4</sup> Then we impose a structure of interdependencies among the KUs. KST stipulates formally that the items in a knowledge domain are mastered in a constrained order as defined by implication relations.

In KST the interdependencies, also known as surmise relations, among KUs is represented by what in the field of Artificial Intelligence is known as an AND/OR graph. In an AND/OR graph the two inference rules are:

1. If  $A$  is known, then  $B$ , and  $C$ , ... and  $N$  are also known.
2. If  $A$  is known, then *either*  $B$ , or  $C$ , ... or  $N$ , is known.

The inference in (1) captures the *basic prerequisite* relationship. “A prerequisite link represents the fact that one of the related KU has to be learned before the other” (Brusilovsky

2012). If a student can solve a problem represented by  $A$ , for example, then we can “surmise” or infer that they can solve problems represented by  $B$ ,  $C$ , ..and  $N$ , which are all  $A$ ’s prerequisites.

The inference in (2) captures the *alternate prerequisite* relationship. It expresses cases where the same knowledge state can be reached through alternate routes. For example, knowing how to create a loop in a programming language such as **C** requires mastery of the ‘for,’ ‘while,’ or ‘do’ constructs. To reach a knowledge state that contains the ability to create loops requires one or more, but not necessarily all, of the alternate predecessor states.

There exist a variety of potential architectures for building knowledge structures. Whatever the choice, the important factors are:

1. *KUs represent meaningful and significant units in the domain of knowledge.*
2. *The user’s mastery of each KU can be reliably assessed.*
3. *There is some order in the way users learn KUs* (Desmarais et al. 1995).

One of the original motivations of KST was to overcome the shortcomings of the psychometric approach to the assessment of knowledge. The psychometric model, as used in standardized tests and implemented in the form of Item Response Theory (IRT), places an individual at best in one of a few dozen ordered categories. IRT lacks the ability to make granular cognitive assessments of knowledge mastery.

This leads to a statement of our first design principle:

**Principle 1** *The ability to make granular, dynamic cognitive assessments should be a yardstick for evaluating baseline effectiveness of any learning system, including adaptive systems. Baseline effectiveness can be further strengthened and extended through techniques such as randomized control trials (RCT) or observational studies that utilize causal inference and modeling.*

As we will see in our discussion of learner models, with KST’s overlay approach, we can uncover the individual’s “knowledge state” in terms of the exact set of concepts mastered, not just after the fact but throughout the learning process. The granularity is as fine as the number of KUs in the domain model and the potential adaptability is proportional to the knowledge structure’s branching capability, which in the case of KST can be in the trillions.

Although AND/OR graphs are a powerful formalism for building knowledge structures, we will give preference to a somewhat simpler variant called a partial order knowledge structure (POKS). Desmarais and his colleagues proposed POKS as a more efficient and cost-effective way of building and representing knowledge structures. It draws from KST for the representation of knowledge and on naive Bayesian networks for the inference of knowledge” (Desmarais et al. 1995; Desmarais et al. 2007). In POKS the ordering constraint involves only the first type of inference rule, the basic prerequisite relation. POKS knowledge structures, therefore, do not have the ability to represent alternate means of reaching the same knowledge state.<sup>5</sup>

POKS has the formal properties of directed acyclic graphs (DAG), more commonly known as network graphs. Although DAG lacks the expressive power of AND/OR graphs, it has several advantages. First, because DAG is ubiquitous in many domains for modeling,

their formal properties and methods of application are likely to be better understood by practitioners. Second, POKS makes it easier to automate the discovery, using data rather than human engineering, of knowledge structures and user models. Desmarais, Maluf, and Liu (1995) have argued, for example, that the POKS technique allows the induction of knowledge structures from empirical data. The induction technique is based, in part, on statistical hypothesis testing over conditional probabilities that are determined by the KUs' learning order.

Formally we can represent the domain model, therefore, as a DAG consisting of nodes and edges. Nodes represent KUs and edges represent prerequisite relationships among nodes. We will refer to the graph representation of a domain model as a structured domain model.

The branching structure can be complex as in Assessment and Learning in Knowledge Spaces (ALEKS)<sup>6</sup> implementations of KST or very simple, reflecting the tree like structure of a textbook's organization into chapters and sections. We conclude this section on domain models by stating our next principle:

**Principle 2** *Partial Order Knowledge Structures (POKS) or Directed Acyclic Graphs (DAG) are optimal for formally representing knowledge domains.*

#### Learner model

A learner model encodes what the learner knows and what they don't know *against* some pre-defined knowledge domain. Thus, an important class of adaptive systems take the approach of "overlaying" the learner model against the domain model.

We have already seen, for example, that a knowledge domain denoted by  $\Delta$  and composed of KUs can be represented as:

$$\Delta = \{\delta_1, \delta_2, \delta_3, \dots, \delta_n\} \quad (2)$$

An individual's knowledge state  $\Lambda$ , of items mastered in some knowledge space, can then be represented as a subset of  $\Delta$ :

$$\Lambda = \{\delta_i \text{ mastered} \mid \delta_i \in \Delta\} \quad (3)$$

For example, in the ALEKS adaptive system, which is based on KST, the knowledge space for Preparation for Calculus consists of 181 KUs (Harper and Reddy 2013). We can be represent its domain model (the unstructured portion) as:

$$\Delta = \{\delta_1, \delta_2, \dots, \delta_{181}\} \quad (4)$$

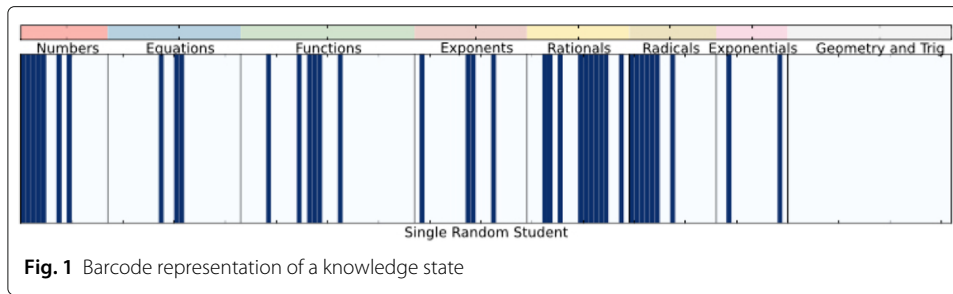
Accordingly, we can encode the knowledge state of each learner as an array of ones and zeros corresponding to whether or not the knowledge state includes that item.

"Enumerating the items in the knowledge space from 0 to 181, we can visualize a single knowledge space as a barcode, with the bar filled in if the student has the item and empty otherwise" (Harper and Reddy 2013).

Figure 1 represents a barcode visualization of a knowledge state.

If  $\Lambda_1$  is a specific learner, then the array  $\Lambda_1^t$  represents her knowledge state at time  $t$ .

$$\Lambda_1^t = [1, 1, 1, 0, 0, 1, \dots, 0, 0] \quad (5)$$



**Principle 3** *A simple barcode visualization, and corresponding one-dimensional array, can serve as a canonical representation of a learner's knowledge state at any moment in time  $t$ .*

The learner's *evolving* knowledge state can be represented as a time-series matrix where rows represent KUs and columns represent discrete times.

$$\Lambda_1 = \begin{matrix} & t_0 & t_1 & t_2 & \dots & t_n \\ \begin{matrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \vdots \\ \delta_n \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & \dots & 1 \\ 0 & 1 & 1 & \dots & 1 \\ 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} \end{matrix} \quad (6)$$

Since a learner's knowledge state is always inferred we can also attach probabilities to each item. Let  $\iota$  represent the inferred knowledge state of some learner  $\Lambda_1$ :

$$\iota(\Lambda_1) = \{P(\delta_1), P(\delta_2), \dots, P(\delta_n)\} \quad (7)$$

where  $P(\delta_i)$  is probability associated with knowledge unit  $\delta_i$  and  $n$  is the number of KUs in  $\Delta$ .

Global mastery of the domain  $\Delta$  can be computed, for example, as:

$$\frac{\sum P(\delta_i)}{n} \quad (8)$$

The global probability can be interpreted as the probability than an arbitrarily chosen KU would be mastered by the particular learner (Desmarais et al. 1995).

We conclude this section by stating our next design principle:

**Principle 4** *We can achieve granular, dynamic cognitive assessments of knowledge mastery by using an overlay approach for learner modeling.*

#### Assessment model

The assessment model is *how* we know what a learner knows. What a learner knows is inferred through assessment probes. Thus far we have used KUs, items, and concepts synonymously. But this masks an important distinction. In systems such as ALEKS, and in many intelligent tutoring systems, items are problems or problem sets. Learners exhibit mastery of an item by successfully solving problems. In KST 'problems' serve a dual purpose: on the one hand, they serve as KUs in the domain model and, on the other hand, they serve as assessments in the assessment model. But this is not generalizable to other

use cases or other domains where solving problems is not the only means for exhibiting mastery.

This raises the question of how we can include and assess concepts within POKS structures. The simplest way of including concepts into POKS is to mimic what teachers do by breaking up learning mastery into a set of learning goals or learning objectives.

The relationship of a KU or concept (in the domain model) to a learning objective can be regarded as one-to-many. Accordingly, we map each knowledge unit to a *set* of learning objectives. We say that a learner has *mastered* or learned a concept if they have mastered the corresponding set (or some subset) of weighted learning objectives.

When KUs are concepts or concept-like entities rather than problem-types, we introduce learning objectives as intermediate entities or bridges from concepts to assessments. Corresponding to each KU then is a set of learning objectives. The set of learning objectives corresponding to a knowledge unit  $\delta_1$  can be represented as:

$$\delta_1 = \{lo_1^1, lo_1^2, lo_1^3, \dots\} \quad (9)$$

Once we have defined the set of learning objectives and mapped them to the knowledge units, we perform a similar mapping of learning objectives to assessments.

$$lo_{22} = \{a_{22}^1, a_{22}^2, a_{22}^3, \dots\} \quad (10)$$

The relationship of learning objectives to assessments is also one-to-many.

In the most general case the assessment model consists, therefore, of a mapping from the space of knowledge units (as defined in the domain model), their corresponding learning objectives (also in the domain model) to a set of assessments in the assessment model.

**Principle 5** *To generalize adaptive systems beyond STEM, knowledge units (KUs) can be thought of as concepts or topics. But then KUs need to be mapped to learning objectives, which in turn need to be mapped to assessments.*

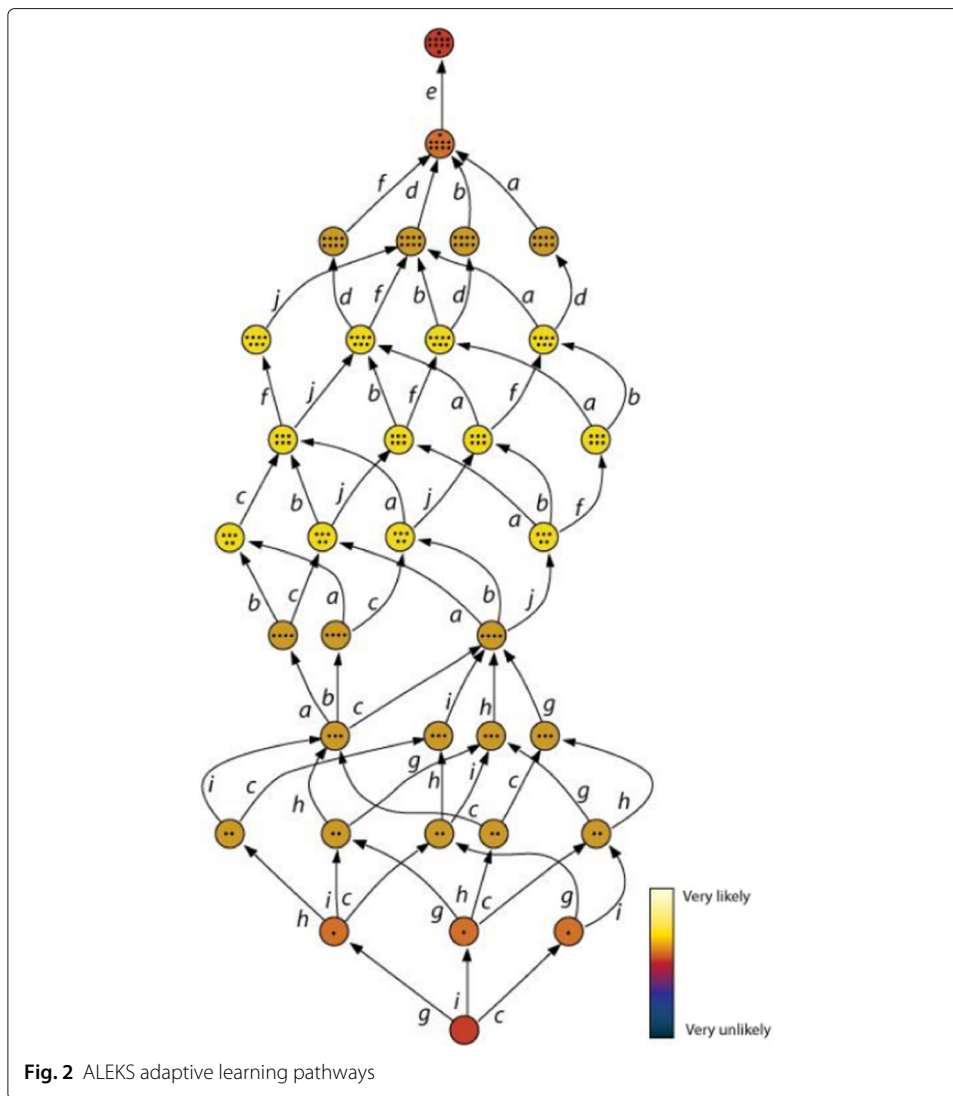
### Transition model

The assessment model determines a learner's current knowledge state. The transition model then determines the next logical knowledge state, or what the learner is ready to learn next. In systems such as ALEKS (Falmagne et al. 2013), for example, the number of possible pathways can be in the trillions. By contrast, most ITS follow the relatively simple hierarchical scope and sequence of the structure of chapters in a book.

For purposes of visualization, the graph in Fig. 2 represents a learning space of only 10 topics and 34 knowledge states. In reality, an actual learning space in an ALEKS implementation contains roughly 200 - 400 topics and over 1 trillion knowledge states. The adaptive algorithm for ALEKS calculates dynamically the unique state transition for each learner among the billions of possible paths. In KST transitions are calculated based on "inner" and "outer" fringes. The intuitive idea is that based on the precedence relationship of the knowledge structure, learning can take place step by step, one problem type at a time.

The "outer fringe" of some knowledge state  $K$  is the set of all problems (or items)  $p$  such that adding  $p$  to  $K$  forms another knowledge state. Learning progresses by mastering a new problem in the outer fringe, which in turn creates a new knowledge state with its





own outer fringe. If a learner experiences difficulties mastering a problem in the outer fringe, they are likely to be sent back to reviewing material in the inner fringe which are predecessors to state  $K$ .

KST's concepts of inner and outer fringe represent a particular implementation of the transition model. In general, each adaptive system will have a transition model or transition function  $\Theta$  that maps an individual learner  $\Lambda_i$  and their current knowledge state  $\sigma$  to their next logical knowledge state  $\sigma + 1$ .

$$\Theta(\Lambda_i, \sigma) = \sigma + 1 \quad (11)$$

We can also state another design principle towards transparency of adaptive systems:

**Principle 6** *An adaptive system's transition model, or how it determines what the learner is ready to learn next, should be made transparent to the learner in order to support open learner models. An open learner model makes a particular student's learner model explicit in order to support self-awareness and self-regulation.*

### Pedagogical model

In an adaptive learning system *how* a learner is to progress from their current knowledge state to the next is specified in the pedagogical model. Formally we define the pedagogy associated with a knowledge transition as a set of activities.

The pedagogy  $\Pi$  that allows a learner to go from knowledge state  $\sigma$  to  $\sigma + 1$  can be formally represented as a set of activities:

$$\Pi_{\sigma}^{\sigma+1} = \{\alpha_1, \alpha_2, \alpha_3, \dots\} \quad (12)$$

Some examples of activities:

$\alpha_1$ : learner reads chapter 2

$\alpha_2$ : learner watches video 5

$\alpha_3$ : learner participates in group exercise 7

The pedagogical and assessment models lie at the heart of adaptive systems. The strength of the assessment model depends on the availability of a *range* of assessment types and how they are implemented concurrently with the pedagogical model. The strength of the pedagogical model in turn depends on the breadth and depth of activities specified for each knowledge transition.

**Principle 7** *The pedagogy of an adaptive system should be stated explicitly as learner activities, not reified as learning resources or learning assets. Moreover, not all learning activities, to accomplish a learner's transition from one knowledge state to another, will or should take place within the adaptive system proper.*

### Intelligent tutoring systems: inner loop vs outer loop

Intelligent Tutoring Systems (ITS) form an important class of adaptive systems. ITS emerged in the late 1960s and have become more sophisticated through the years. ITS are based functionally on a distinction between an *inner loop* and an *outer loop* (VanLehn 2006).

ITS guide learners through a sequence of instructional activities in an outer loop and monitor step-by-step progress on particular activities within an inner loop. The outer loop determines *which* items the learner is ready to learn next or the next set of tasks to be performed. The outer loop is about the sequence and selection of knowledge units or tasks. By contrast, the inner loop is about the navigation of steps *within* a task.

Corresponding to the outer loop and inner loop there are two associated sets of models. We may call these the inner and outer cognitive models. The inner loop model contains for each problem type or task an explicit understanding or model of not only the correct answer but the *correct steps* required to arrive at the correct answer. Based on the inner cognitive model, adaptive monitoring within the inner loop provides the learner with error-specific feedback on incorrect steps, hints for the next step, and solution review. By contrast, the outer loop model primarily determines the scope and sequence of items or knowledge units to be learned. Thus far, our discussion of models in 'Formal framework' section have focused only on the outer loop.

Corresponding to the outer and inner loop distinction there are also two major senses of adaptivity. Macro-adaptation refers to the variation among tasks, activities and instructional materials presented to the learner in the outer loop. Micro-adaptation refers to the variation in feedback and error correction as a learner works within a particular task or activity.

Much of the research and advances in ITS have focused on learner modeling within the inner loop. Examples include knowledge tracing in Cognitive Tutor, Constraint-based modeling, and the Expectation-Misconception approach used in AutoTutor.<sup>7</sup> “In all of these models, the emphasis is on supporting micro-adaptivity. Attempts to handle macro-adaptivity (in Intelligent Tutoring Systems) have been modest” (Rus et al. 2013). By contrast, KST and systems such as ALEKS have focused on macroadaptivity in the outer loop.

Model design of inner loop in ITS has been highly domain specific and also task specific. Therefore, ITS have been costly to implement, difficult to calibrate, and difficult to scale. If we look at the history of ITS, implementations have also been restricted mostly to STEM domains.

“A primary weakness of this type of approach for domains with many skills or misconceptions is that the skills and misconceptions must be reasonably enumerated in order to provide feedback (Brown and VanLehn, 1980). The creation of this domain model can be time consuming. Although development time is not widely reported, general ITS system design is estimated at 200-300 hours of development time per 1 hour of instructional content (Pavlik et al. 2013)”

The focus on microadaptivity in ITS has come at a cost. Inner cognitive models focus narrowly on very specific skills such as the steps required in solving a quadratic equation. Cognitive models for each skill has to be crafted independently and remains largely manual. And as we have noted, microadaptivity also remains restricted to STEM disciplines and even then only areas which lend themselves to a high degree of procedural codification.

It can also be argued that microadaptivity is best left to the instructor. When it comes to providing feedback and error correction in the inner loop, an instructor's knowledge and experience trumps machine intelligence and, we can speculate, will continue to do so in the foreseeable future.

**Principle 8** *For modeling the outer loop, next generation adaptive learning systems should draw on KST as a best practice. Domain specific microadaptivity, except in niche cases, should be regarded as the primary realm of the instructor. For modeling the inner loop more emphasis should be given on developing domain independent models to accelerate scalability.*<sup>8</sup>

Are we suggesting then that domain specific microadaptivity should not be a core component of adaptive systems? No. As we suggested earlier, the vantage point of this paper is between applied research and product development. While microadaptivity will and should remain an active part of research in adaptive systems, we are not likely to see it scale in the near term by becoming incorporated into mainstream products. In the section on learning objects we will suggest an approach that could inject microadaptivity into the mainstream of product development.

### Deep learner models

By a Deep Learner Model (DLM) we mean extending the core learner model to track not only what a learner knows, but also what the learner knows about their knowing, how they feel about their learning, their desires and motivations, the learning strategies they employ, and how they interact socially with others. A DLM also incorporates learner preferences and broader learner characteristics such as accessibility and the need for assistive technologies.

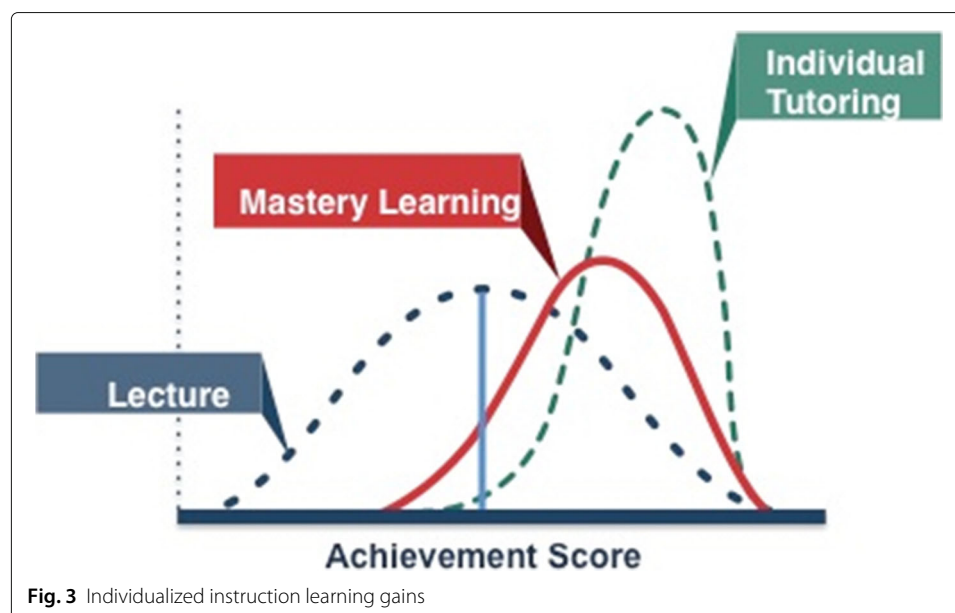
In order to set the stage for our discussion of Deep Learner Models, let's review an important implication of Bloom's findings in "The 2-Sigma Problem" (Bloom 1984). Bloom's central claim is well known: individualized instruction, when compared to the standard mode of instruction, substantially increases the mean of performance.<sup>9</sup> Please see Fig. 3.

What is less well known about Bloom's findings is a subtler and deeper point: Individualized instruction can also *decrease* the standard deviation. This means that with the right instructional strategy, students at the low end of the distribution begin to catch up with learners at the high end.

According to Bloom, a good tutor should exhibit an aptitude-treatment interaction: both groups should learn, and yet the learning gains of the low students should be so much greater than those of the high ones that their performance in the post-test ties with that of the high ones. That is, one benefit of tutoring is to narrow or even eliminate the gap between high and low (Chi and VanLehn 2010).

If Bloom's hypothesis is correct, then an important benefit of adaptive learning systems is that they can also help to narrow the achievement gap between high and low learners. We can state this desired effect as an evaluative design principle for adaptive learning systems:

**Principle 9** *A highly effective adaptive learning system should not only help to raise the mean of performance among learners but also decrease the standard deviation. We will refer to this phenomenon as closing the achievement gap.*



**Fig. 3** Individualized instruction learning gains

If closing the achievement gap between high and low performers is also a desired outcome, then we need to consider what accounts for the gap in the first place. We should not assume that learner success is due *only* to domain level understanding and skill. Human teachers impart not just knowledge but include among their goals: “first, to sustain and enhance their students’ motivation and interest in learning, ... and second, to maintain their pupil’s feelings of self-esteem and self-efficacy, even in the face of difficult or impossible problems” (Lepper et al. 1990, p.219).

Great teachers motivate students to learn and equip them with a variety of cognitive and meta-cognitive strategies to succeed. In that light,

“one of many hypotheses is that low learners lack specific skills about how to think, including general problem-solving strategies and meta-cognitive skills” (Chi and VanLehn 2010).

In his commentary of VanLehn’s review (2006) of ITS du Boulay (2006) noted that the affective, motivational and metacognitive state of the student has only “fleetingly” been addressed in most learning systems. Traditional educational systems “have operated largely at the cognitive level and have assumed that the learner is *already* able to manage her own learning, is *already* in an appropriate affective state and also is *already* motivated to learn” (du Boulay et al. 2010, p.197).

In recent years, recognizing the limitations of “cognition-only” approaches, researchers have begun to model key aspects of students’ motivation, affect, and meta-cognition with the aim of providing adaptive scaffolding for addressing differences in these areas (Desmarais and Baker 2012). Drawing on the work developed by du Boulay and colleagues (2010) we can classify deep learner models in terms of the taxonomy in Table 1.

We now provide some extensions to the core learner model as examples of non-cognitive models. The aim is not to be comprehensive but to illustrate modeling approaches and lines of thought that take us beyond purely cognitive models.<sup>10</sup>

### Learning strategies

Learning strategy is a broad concept. It encompasses the plans, steps, tools, and methods a learner employs during the learning process. In this section we consider how adaptive learning systems can support one such strategy, namely improving knowledge acquisition and recall.

Acquisition and recall is a fundamental characteristics of all learning. Generic learner models are based on a simplification. They assume that once someone knows what they know, they know it forever. Moreover, the same models tend to assume that knowledge

**Table 1** Deep learner models pedagogical taxonomy

Type	Deep learner models Pedagogical Focus
Cognition	Increase the learner’s knowledge and skill
Affect	Improve the learner’s emotional state and overall sense of well-being in their approach to learning
Motivation	Increase the learner’s desire to learn, including her confidence and willingness to expend effort in learning
Learning Strategy	Increase the learner’s awareness and skill employing various learning strategies

acquisition is a one time event rather than the result of sustained practice and reinforcement. Both assumptions, of course, are false. Knowledge is acquired, recalled, and strengthened through various modes practice and repetition.

Studies of learning have shown a relationship, for example, between practice amount and performance (Ericsson 2006). But learning is also enhanced by practice spacing. If practice is separated rather than massed, or if the spacing between practice sessions is larger rather than smaller, retention tends to improve.

The distributed-practice effect is surely one of the most solid findings in learning and memory research. It holds for both motor skill and declarative learning (Tom and Dewar 2014, p.512).

Experimental psychology has revealed a number of strategies for improving acquisition and recall. Despite the demonstrable benefits of these strategies they are not adopted by many learners. Some plausible reasons for this are:<sup>11</sup>

- *Ignorance*: learners don't know of these strategies (corollary: if learners knew of the strategies they would use them and improve their learning.)
- *Metacognitive failure*: a persistent illusion whereby familiarity with material-to-be-learned induces overconfidence that it can be successfully recalled (corollary: if the learners could be shown the ineffectiveness of naive learning practices they would change how they study.)
- *Motivation*: use of these strategies is less fun than naive study techniques or they require more effort to deploy (corollary: to change how learners learn we need to make it easier or more fun for them to study in different ways.)
- *Scaffolding*: learning systems do not provide proper "guard rails" for learners to use the strategies properly (corollary: if learning systems could detect and provide personalized support for optimal learning strategies learners will be able to improve their learning.)

An important extension to the base learner model for adaptive systems, therefore, is to provide adaptive scaffolding for improving learning strategies, including enhancing knowledge acquisition and recall.

**Principle 10** *Learner models should provide adaptive scaffolding for optimizing knowledge acquisition and recall. The models are likely to be domain independent and, therefore, scalable across different knowledge domains.*

### Motivation

It has long been acknowledged that motivation plays a central role in learning (del Solato and du Boulay 1995). It affects how learners approach their education, how they relate to others, the amount of time and effort they devote to their learning, how much support they seek when they are struggling, how they engage or disengage from learning activities, and how they perform on informal and formal assessments (Usher and Kober 2012).

It is difficult to address knowledge gaps, no matter how sophisticated the learning system, if a learner is fundamentally unmotivated:

Even with the best administrators, faculty, curriculum, and materials in place, if students are not motivated to learn and excel, achievement gains will be difficult, if not

impossible. Higher motivation to learn has been linked not only to better academic performance, but to greater conceptual understanding, satisfaction with school, self-esteem, social adjustment, and to lower dropout rates (Usher and Kober 2012, p.3).

Although there are a number of frameworks in learning science that characterize motivation, they tend to agree on the major components as defined in Table 2 (for Education Policy).

For designing adaptive systems that take motivation into account, some of the central questions are:

- What are the principal characteristics of motivation in learning?
- How can we measure and detect motivation and its loss?
- How does motivation effect learning and vice versa?
- How do motivational states change during learning and what are its causes?
- How can motivation be changed in a learner?

We need not assume that motivation, in all its dimensions, subtleties and manifestations, can be influenced by machines alone. The teacher will always have a central place in motivating and inspiring learners. However, we might be able to use machines to detect and monitor fluctuations in motivation. du Boulay and his colleagues have coined the term “Motivationally Intelligent Educational Systems” to describe adaptive systems that aim “to maintain or even increase the learner’s *desire to learn* and *her willingness to expend effort* in undertaking the, sometimes hard, activities that lead to learning” (du Boulay et al. 2010).

A number of efforts are underway to model motivation in learning systems. It is beyond the scope of this paper to survey the various approaches. In this section we sketch how we might *operationalize* one such framework for implementation within an adaptive system.

Rebolledo-Mendez, du Boulay, and Luckin (2006), for example, have modeled motivation within a Vygotskian intelligent tutor. A somewhat different approach was taken by del Solato (1995) where the detection of motivation was operationalized<sup>12</sup> with three variables: effort, confidence, and independence. Effort can be defined as the degree of persistence and participation a learner shows in their learning. Confidence is the learner’s self-reported degree of confidence in solving a problem. Independence is the learner’s use of help and other available forms of assistance. Each characteristic can be modeled and a learner’s motivational state and fluctuations can be monitored.

Once the variables are operationalized in the learning system, the adaptive engine can respond or intervene with comments, encouragement, provision of help, or choice of

**Table 2** Four possible dimensions of motivation

Dimension	Motivation	
	Attitude	Indicators
Competence	Am I Capable?	The learner believes he or she has the ability to complete the task.
Autonomy	Can I control or manage my learning?	The student feels in control by seeing a direct link between her actions and an outcome.
Value	Does it interest me? Is it worth the effort?	The student has some interest in the task or sees the value of completing it.
Relatedness	What do others think?	Completing the task brings the student social reward, recognition or sense of belonging.

activity. These reactions are determined on the basis of a set of production rules (see Table 3) that fire in response to the values of the three variables (del Solato and du Boulay 1995).

**Principle 11** *The design of a deep learner model should begin with an explicit hypothesis (e.g. components of motivation). The hypothesis should then be operationalized in order to confirm model validity with data and experiments.*

### Self regulation

In recent years there has been increasing research interest in understanding and incorporating support for metacognitive skills in intelligent learning systems.

Flavell (1979) coined the term metacognition to indicate “one’s stored knowledge or beliefs about oneself and others as cognitive agents, about tasks, about actions or strategies, about how all these interact to affect the outcomes of any sort of intellectual enterprise” (Flavell 1979, p.906). Metacognition is more than simple self awareness. It is deeply connected to agency and control. A sophisticated learner constantly makes purposeful adjustments in themselves and in their environment during the learning process.

We can describe this purposeful monitoring and adaptation by the learner as self-regulated learning, which is a form of metacognition. Pintrich described self-regulated learning as: “an active, constructive process whereby learners set goals for their learning and then attempt to monitor, regulate, and control their cognition, motivation, and behavior, guided and constrained by their goals and the contextual features in their environment” (Pintrich 2000, p. 453). Students’ use of such strategies has been shown to correlate positively with learning outcomes (Pintrich 2000).

**Principle 12** *We can extend the deep learner model taxonomy by incorporating a “meta” layer corresponding to cognition, affect, motivation, and learning strategy. See Table 4.*

### Advanced learning analytics

Until recently adaptive learning and learning analytics systems have developed independently of each other. In this section we consider how their convergence is likely to

**Table 3** Motivation diagnosis and intervention

Likely state	Production rules	
	Cognitive Intervention	Metacognitive intervention
Mastery, Low Effort	Increase problem difficulty	Display learning progress
Mastery, High Effort	Maintain problem difficulty	Praise effort and persistence
Hint Abuse, Low Effort	Reduce problem difficulty	Deemphasize importance of immediate success
Mastery, Medium Effort	Maintain problem difficulty	Praise importance
Quick Guesses, Low Effort	Reduce problem difficulty	Deemphasize importance of immediate success
Hint Avoidance, High Effort	Reduce problem difficulty	Offer hints upon incorrect answer in the next problem
Quick Guesses, Hint Abuse	Reduce problem difficulty	Deemphasize importance of immediate success
Low Mastery, High effort	Reduce problem difficulty	Emphasize importance of effort and perseverance



**Table 4** Extended deep learner model taxonomy

Type	Deep learner models Pedagogic Focus
Cognition	Increase the learner's knowledge and skill
Meta-Cognition	Increase the learner's insight into what she understands and can do, and her ability to regulate the learning process
Affect	Improve the learner's emotional state and overall sense of well-being in their approach to learning
Meta-Affect	Increase the learner's insight and ability to regulate her feelings as a learner
Motivation	Increase the learner's desire to learn, including her confidence and willingness to expend effort in learning
Meta-Motivation	Increase the learner's insight and ability to regulate her motivation
Learning Strategy	Increase the learner's awareness and skill employing various learning strategies
Meta-Learning Strategy	Increase the learner's insight and regulation of learning strategies

shape the future of adaptive systems. Advanced analytics can strengthen two functions in adaptive systems: the ability to surface just-in-time actionable insights and to establish a feedback loop for iteratively improving the quality of adaptive models.

#### **Analytics as actionable insights**

The current generation of adaptive systems are closed systems. Because all data is local to the system, learner models are restricted to what the underlying data can support. As adaptive systems incorporate deep learner models, they will need to evolve architecturally from closed data islands to open systems capable of exchanging data and services residing externally to the adaptive system proper. We will consider the architecture of such systems in more detail in 'Big data architecture for adaptive learning systems' section. In this section we consider some baseline characteristics for generating learner insights and feedback using advanced analytics.

In general analytics capability or maturity can be seen as occurring at three levels or stages. Analytics is first and foremost about posing and answering questions. *Analytics Level I* is the realm of traditional business intelligence dashboards and reports. At a company that sells widgets, for example, some questions necessary to operate the business might be: How many widgets were sold? How did sales break down by region? How did actual sales compare to targeted sales? Analytics I poses and answers questions about the *past*, or, at best, about the *present*.

With *Analytics Level II* we forecast the future. Using techniques such as predictive modeling we can pose and answer questions about what will happen. How many widgets will be sold next year? If web advertising were to increase by amount  $x$ , will there be a corresponding increase in revenue by amount  $y$ ? At the next level of maturity Analytics II poses and answers questions about the *future*.

*Analytics III* is optimization and is the most advanced level of analytics capability. In a decision situation multiple options are available. Among these which is the best option? Or, if an individual is trying to navigate from point A to point B, what are the different available routes and which particular route is the best route for that individual? Analytics III is highly personalized. The best course of action for a particular individual or company is not the same for another. What are the different ways of increasing widget sales next

year? Among these which is optimal given constraints such as budget, resources, and competition? In Analytics III the data is about the desired or *optimal future*.

Among the most important attributes of an effective learning systems is its ability to provide continuous quality personalized feedback to the learner. Given the three levels of analytics the feedback to the learner must provide data and insights not just about the past, but include data about the future and the desired future:

**Principle 13** *An effective adaptive system provides learner insights, feedback, error correction, and enrichment at all three analytics levels. See Table 5*

Exemplary large scale implementations of predictive modeling in learning analytics include the Course Signals Project at Purdue University (Arnold and Pistilli 2012) and the Student Success System by Desire2Learn.<sup>13</sup>

**Data analytics and experimental learning science**

Models are scientific hypotheses subject to confirmation, refinement and refutation. In adaptive systems each model is provisional and takes root through ongoing experiments and data collection. At a higher level of generality the set of models in turn form a nascent theory. As the theory develops, what were once independent models begin to coalesce into a fabric of connected hypotheses. In this section we illustrate, with a concrete example, the use of analytics as part of a workflow where models are implemented initially as hypotheses and data analytics is used to validate and refine the model.

In the section on learning strategies we discussed how experimental psychology has revealed a number of “practice” strategies for strengthening acquisition and recall. The analysis of practice in experimental studies of learning and memory spacing confirms that distributed practice is better than massed practice. Research in perceptual learning has confirmed that outcome can be affected by both the length and distribution of practice. Thus, there are several parameters that can be varied: the number of practice sessions, the amount of practice in each session, different types of practice in each session, and the length of breaks between sessions. Initial results also indicate an important difference in early and later stages of training. Larger gains occur in early stages (due to latent learning) than in later stages.

Given the importance of practice in many adaptive learning system it would be natural to implement a model that takes advantage of the distributed space effect for enhancing acquisition and recall. But the experimental literature only provides cursory guidance in terms of the choice of initial parameters.

Accordingly, we can roughly map five stages in model design and implementation as part of an analytics or data driven implementation of adaptive learning systems. During the initial phase a model is designed (e.g. distributed practice) and implemented given

**Table 5** Analytics levels

Level	Analytics levels	
	Learner	Instructor
Analytics I	How am I doing?	How are my students doing?
Analytics II	How will I do?	How will my students do?
Analytics III	How can I do better?	How can my students do better?

the best knowledge in learning science. During the second stage explicit experiments are devised within the system to control and isolate relevant variables. During the third stage data is collected and analyzed to confirm and validate model assumptions and parameters. During the fourth stage model parameters are adjusted and tuned based on the prior analysis. Finally, more sophisticated automatic models are implemented that can self-learn a subset or all of the model parameters.

In 'Big data architecture for adaptive learning systems' section we will see how these two analytics capabilities, of providing just-in-time insights and embedding ongoing data collection and experimentation to improve the quality of adaptive models, can be supported with an open, modular architecture.

### Learning objects

In this section we introduce the idea of modular, reusable learning objects as possible building blocks for a next generation adaptive learning system. Learning objects are often confused with learning assets. A learning object, unlike a learning asset, is a *pedagogical* atom and, as such, must conform to a certain formal structure. Although the concept of a learning object has been around for at least two decades, how they might be incorporated in adaptive systems is not well understood. In this section we outline some formal properties of learning objects so that they might be incorporated in future adaptive systems.

The current generation of adaptive systems, including Intelligent Tutoring Systems, suffer from two major limitations. First, they operate as *closed environments* on *fixed domains*. Learners and instructors, therefore, have little or no ability to control or vary the course of instruction themselves. This violates an important principle of learning, namely that learners should be able to control their own learning and be able to self-regulate the course of their learning journey. Second, advanced learning is often serendipitous and unstructured. In the course of studying the topic of logistic regression, for example, a student might be inspired to take a short detour and learn more deeply about mathematical functions or specific concepts in probability theory such as a probability distribution function. Can adaptive systems be designed to support such investigative and opportunistic learning?

Modularity in the form of learning objects potentially addresses both limitations. We assume a world, therefore, in which learning objects are available to be combined and re-combined to form personalized learning pathways. In the pre-iTunes world in music, for example, consumers had to purchase the entire album even if they were interested in listening to only one song. It was also not possible, except in professional studios, to combine songs from multiple albums. Today the standard music experience is that each listener is able to create, modify, and access their personalized playlists of songs.

Similarly, a next-generation adaptive system functions in an *open environment* operating on *flexible domains*. Open means that the learner or instructor is able to create their own learning pathway based on a mixture of proprietary and Open Educational Resources (OER). Flexible means that the learning pathway is not fixed ahead of time but can be dynamically generated as needed. For example, as part of a biology course an instructor might wish to intersperse various topics in statistics at different points in the course. Or, a learner might create a "refresher" playlist on logarithms and exponential functions as part of their study of physics.

If learning objects serve as the building blocks of a next-generation adaptive learning system, the challenge is to determine which individual learning objects are *effective* and which combination or sequence of learning objects is *optimal* for each learner.

The starting point for a modularized adaptive system are curated learning objects and playlists. By curation we mean that subject matter experts (SMEs) design individual learning objects and recommend the initial sequence. But then we use embedded analytics to update our models using a design loop that mimics the scientific process. SMEs, through their work of curation, are in the best position to frame initial *hypotheses* about what works best in learning. But then using data and experiments we continuously update the quality of our models.

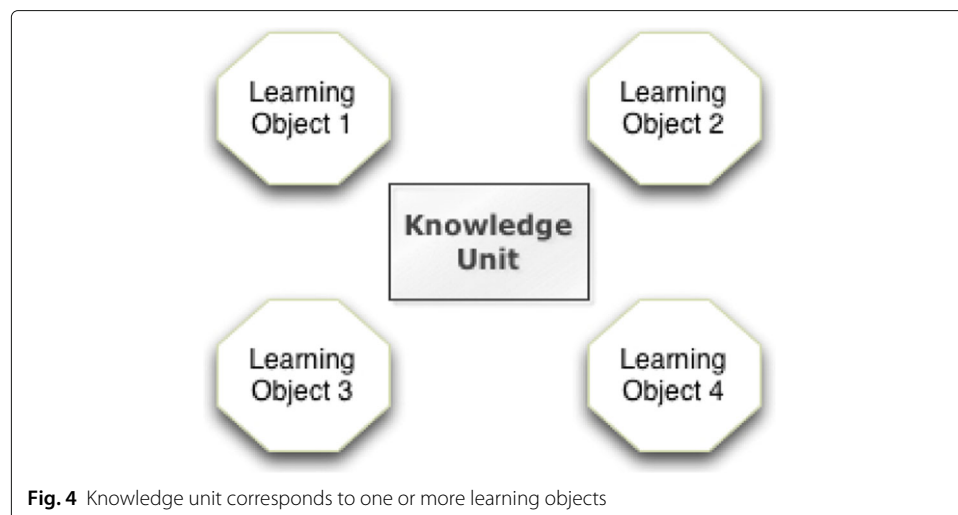
We are not precluded, of course, from also crowd-sourcing the generation and updating of learning objects. In fact, the most valuable contribution for crowd-sourcing might come in designing cognitive models in the inner loop. An open source authoring tool that allows user creation of learning objects, including embedding the inner cognitive model, could significantly drive innovation in adaptive learning systems.

Let us now turn to the formal structure of a learning object. The nucleus of a learning object is a *learning objective*. The relationship of learning object to learning objective is 1 – 1. In our scheme each knowledge unit corresponds to one or more learning objects or learning objectives. The relationship of knowledge unit to learning objects is one-to-many. See Fig. 4.

Orbiting the nucleus of a learning objective is the corresponding set of learning activities and assessments. The learning activities define the pedagogy of the learning object(ive) while the assessments define how we know that the learner has mastered the learning objective. See Fig. 5

A learning object, therefore, forms a tri-partite structure consisting of a learning objective and its corresponding learning activities and assessments. And built-in to the assessment set is some notion of what it means to have mastered a particular learning objective. A learning object contains in microcosm the domain, pedagogical, and assessment models discussed in ‘Formal framework’ section.

Once we have the base class of learning objects, adaptivity can be designed in a variety of ways. Microadaptivity, in the form of the inner loop, would reside within the learning



**Fig. 4** Knowledge unit corresponds to one or more learning objects

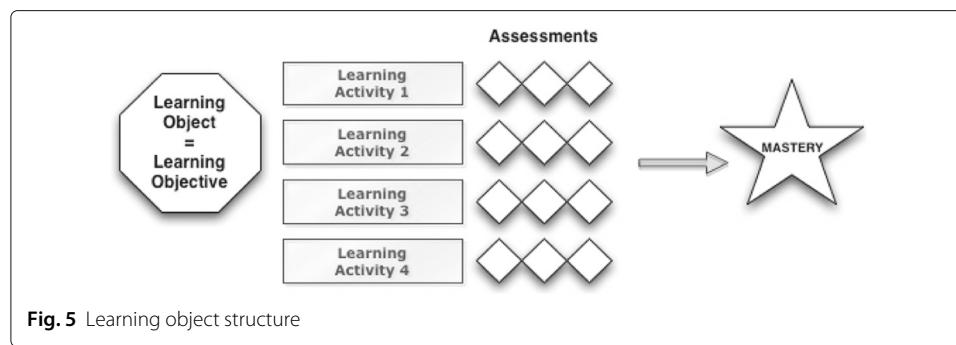


Fig. 5 Learning object structure

object itself. Macroadaptivity can be bootstrapped through a combination of human intelligence and machine intelligence. Subject Matter Experts (SMEs) or crowdsource contributors could create initial playlists corresponding to learning pathways. The data generated by the use of the learning pathways then becomes the basis for personalized recommendations.

### Big data architecture for adaptive learning systems

In this section we describe a scalable cloud-based architecture to support a next-generation adaptive learning system. A principal requirement of the architecture is that we are able to provide near *real-time* adaptive and analytic feedback to hundreds of thousands, if not millions, of users. We also assume that the data to generate the feedback can originate from an arbitrary set of distributed learning systems or tools. We also assume that learner feedback is dynamically generated from multiple deep learner models.

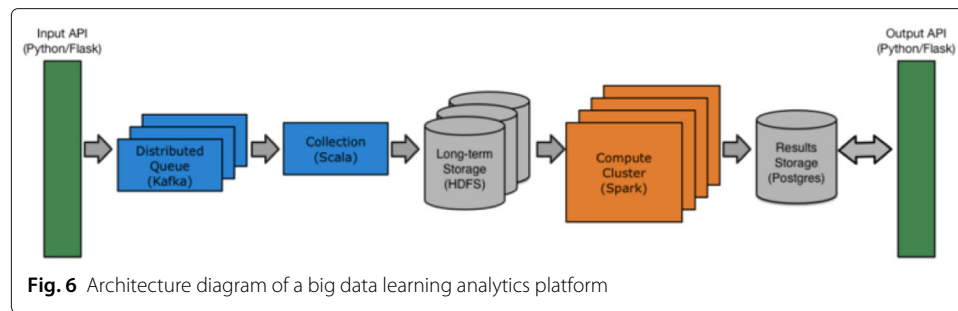
Accordingly, the principal components of a big data learning architecture include:

- a *real-time or streaming layer* (in addition to traditional batch mode) to support data ingestion from distributed learning tools and systems;
- a standards-based protocol for *describing, capturing and transmitting* learning activities and events;
- *short-term and long-term databases* for persistent and transformed data;
- a *parallelized computation layer* to support multiple scalable models, each with its data pipeline and transformation set;
- an *output layer* to support end-user visualizations or data APIs.

These requirements can be met in terms of a  $\lambda$  *Lambda Architecture*, a useful framework for designing scalable big data applications (Marz and Warren 2015). The interoperability requirements can be met in terms of the IMS Caliper Analytics standard in conjunction with other IMS standards such as LTI/LIS/QTI.<sup>14</sup>

In the remaining part of this section we describe a reference implementation called Open Analytics Collaboration Research Environment (Open-ACRE), built by researchers at McGraw-Hill Education and Athabasca University (Lewkow et al. 2016). The Open-ACRE platform consists of input and output APIs, long- and short-term databases, and a parallel computation cluster. A high-level diagram is shown in Fig. 6.

The platform is designed to handle the challenges of scalability, resiliency against data loss, and fault tolerance. Open-ACRE is extensible in that future models can be added without drastic modifications to the base system. In Open-ACRE deep learner models can be based on very simple aggregations to sophisticated machine learning algorithms.



In Open-ACRE learning event data, represented as IMS Caliper events, is ingested by the input API and placed into a distributed queuing system which is implemented using Kafka. The input layer is implemented using RESTful APIs because REST is stateless, easily extensible for future functionality, and agnostic to programming languages and technology stacks.

A collection service, implemented in Scala, pulls data from the queue and stores it in long term storage, which is implemented using Hadoop Distributed File System (HDFS). The Apache Spark compute cluster runs models in parallel on the data in long-term storage and persists output views to the results store, implemented in PostgreSQL. Output views can then be accessed through the output API. Both the input and output APIs are RESTful and implemented in Python using Flask.

The computation engine is capable of simultaneously running multiple deep learner models by taking data from the long term store and performing transformations/aggregations to create output views and results. Apache Spark is used as the model's computation engine as it allows for massively parallel computation, horizontal scalability on commodity hardware, and supports a rich set of APIs based on map-reduce functions, machine learning algorithms, and commonly used statistical routines. Currently, Apache Spark implements APIs in Java, Python, and Scala.

What's important about the  $\lambda$ -architecture approach is not the specific implementation technologies used, which will vary from one organization to the next, but how we can modularize and standardize a scalable fault-tolerant system based on distributed systems while allowing low-latency reads and updates. Most importantly, the architecture provides the ability to run multiple learner models to be run in parallel and their output displayed back to users in near real-time.

## Conclusion

We have sketched a possible future of adaptive learning systems where the next leap of innovation will come from *real-time* dynamic analytic systems that provide *just-in-time* feedback in the *learning moment* to learners and instructors. But to support this new future the architecture of learning systems will have to evolve from what are essentially closed systems today to a modular approach with embedded analytics running on standards-based open platforms.

## Endnotes

<sup>1</sup> Pavlik, Brawner, Gire, Olney and Mitrovic (2013) have been among the first to recognize the groundbreaking nature of Smallwood's contributions (Pavlik et al. 2013).

<sup>2</sup> There is considerable variability in adaptive systems, including the terminology used to describe them. The high-level description of adaptive systems in this paper should be thought of as the analogue of “central tendency” in statistics. Our description targets the mean, not the variability across implementations.

<sup>3</sup> The set of activities specified in the pedagogical model need not take place within the adaptive system proper.

<sup>4</sup> In KST knowledge units are called items.

<sup>5</sup> POKS provides mechanisms for overcoming the more expressive power of KST. See Desmarais (1995)

<sup>6</sup> ALEKS adaptive learning software was developed at UC Irvine starting in 1994 with support from a major National Science Foundation grant. The company ALEKS was founded in 1996 and later acquired by McGraw-Hill Education in 2013

<sup>7</sup> Cognitive Tutor is an ITS developed at Carnegie Mellon University. AutoTutor was developed by researchers at the Institute for Intelligent Systems at the University of Memphis.

<sup>8</sup> See discussion of deep learner models for examples of domain independent models in the inner loop.

<sup>9</sup> By individualized instruction we mean both mastery learning and mastery learning delivered as a 1-1 tutorial.

<sup>10</sup> It should be noted that Bloom’s theory of mastery learning anticipates the idea of deep learner models by considering two principal student characteristics: cognitive entry behaviors and affective entry characteristics. See Bloom (1974).

<sup>11</sup> Derived from a discussion and correspondence with Tom Stafford, University of Sheffield.

<sup>12</sup> By ‘operationalize’ we mean the process of defining a theoretical concept precisely in ways that it can be measured. The concept of operationalization was popularized by the physicist P.W. Bridgman in (Bridgman 1927)

<sup>13</sup> The Student Success System also embeds sophisticated interactive visualizations as part of a workflow where Analytics I and II corresponds to the diagnostic phase and Analytics III to the intervention phase for treating at risk students (Essa and Ayad 2012).

<sup>14</sup> At the core of IMS Caliper Analytics standard are metric profiles, which can be thought of as a common language, common format, and semantics for representing learning activity data gathered from activities across multiple learning systems.

#### Competing interests

The author is an employee of McGraw-Hill Education which owns ALEKS, an adaptive learning software described in this paper.

Received: 13 January 2016 Accepted: 30 August 2016

Published online: 14 November 2016

#### References

- KE Arnold, MD Pistilli, in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, Course signals at purdue: using learning analytics to increase student success (ACM, 2012), pp. 267–270
- BS Bloom, in *Evaluation Comment, 1*, UCLA Center for the Study of Evaluation of Instructional Programs, Learning for mastery, (Los Angeles, 1968)
- BS Bloom, The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educ. Res.* **13**, 4–16 (1984)
- P Bridgman, *The Logic of Modern Physics*. (MacMillan, New York, 1927)
- P Brusilovsky, Adaptive hypermedia for education and training. *Adaptive technologies for training and education*. **46** (2012)

- M Chi, K VanLehn, Meta-cognitive strategy instruction in intelligent tutoring systems: how, when, and why. *Educ. Technol. Soc.* **13**(1), 25–39 (2010)
- JB Carroll, A model of school learning. *Teachers Coll. Rec.* (1963)
- AT Corbett, JR Anderson, Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adap. Inter.* **4**, 253–278 (1995)
- T del Solato, B du Boulay, Implementation of motivational tactics in tutoring systems. *J. Interact. Learn. Res.* **6**(4), 337 (1995)
- MC Desmarais, RS Baker, A review of recent advances in learner and skill modeling in intelligent learning environments. *User Model. User-Adap. Inter.* **22**(1–2), 9–38 (2012). doi:10.1007/s11257-011-9106-8
- A Desmarais, MC Maluf, J Liu, User-expertise modeling with empirically derived probabilistic implication networks. *User Model. User-Adap. Inter.* **5**(3–4), 283–315 (1995)
- M Desmarais, X Pu, J Blais, in *Proceedings of the 2007 GMAC Conference on Computerized Adaptive Testing*, Partial order knowledge structures for cat applications, (2007). Retrieved [date] from [Www.Psych.Umn.edu/psylabs/CATCentral](http://www.Psych.Umn.edu/psylabs/CATCentral)
- P Dodds, JD Fletcher, Opportunities for new ‘smart’ learning environments enabled by next-generation web capabilities. *J. Educ. Multimedia Hypermedia.* **13**(4), 391–404 (2004)
- J-P Doignon, J-C Falmagne, Spaces for the assessment of knowledge. *Int. J. Man-Machine Stud.* 175–196 (1985)
- B du Boulay, K Avramides, R Luckin, E Martínez-Mirón, A Méndez, GR Carr, Towards systems that care: A conceptual framework based on motivation, metacognition and affect. *Int. J. Artif. Intell. Ed.* **20**(3), 197–229 (2010)
- PJ Durlach, JM Ray, Designing adaptive instructional environments: Insights from empirical evidence. *Army. Res. Inst. Rep.* (2011)
- KA Ericsson, in *The Cambridge Handbook of Expertise and Expert Performance*. ed. by PJ Charness, N Feltovich, and RR Hoffman, The influence of experience and deliberate practice on the development of superior expert performance (Cambridge University Press, Cambridge, 2006), pp. 683–703
- A Essa, H Ayad, in *Proceedings of the 2Nd International Conference on Learning Analytics and Knowledge. LAK '12*, Student success system: Risk analytics and data visualization using ensembles of predictive models (ACM, New York, 2012), pp. 158–161. doi:10.1145/2330601.2330641. <http://doi.acm.org/10.1145/2330601.2330641>
- (D Falmagne, J-C Albert, C Doble, D Eppstein, X Hu, eds.), *Knowledge Spaces: Applications in Education*. (Springer, Heidelberg, 2013)
- JH Flavell, Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *Am. Psychol.* **34**(10), 906–911 (1979)
- C for Education Policy, What is motivation and why does it matter? Graduate School of Education and Human Development
- M Harper, AA Reddy, Detecting concepts crucial for success in mathematics courses from knowledge state-based placement data. *arXiv* (2013)
- FS Keller, “good-bye, teacher...” 1. *J. Appl. Behav. Anal.* **1**(1), 79–89 (1968)
- C-L Kulik, JA Kulik, Effectiveness of computer-based instruction: an updated analysis. *Comput. Hum. Behav.* **7**(1), 75–94 (1991)
- MR Lepper, LG Aspinwall, DL Mumme, RW Chabay, in *Self-Inference Processes: The Ontario Symposium*. ed. by JM Olson, MP Zanna, Self perception and social perception processes in tutoring: Subtle social control strategies of expert tutors, vol. 6 (Lawrence Erlbaum Associates, Hillsdale, 1990), pp. 217–237
- N Lewkow, J Feild, N Zimmerman, M Riedesel, A Essa, D Boulanger, J Seanosky, V Kumar, S Kinshuk, in *Proceedings of the Third 2016 ACM Conference on Learning @ Scale. L@S '16*, A scalable learning analytics platform for automated writing feedback (ACM, New York, 2016), pp. 109–112. doi:10.1145/2876034.2893380. <http://doi.acm.org/10.1145/2876034.2893380>
- N Marz, J Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. (Manning Publications Co., 2015)
- P Pavlik, K Brawner, A Olney, A Mitrovic, in *Design Recommendations for Intelligent Tutoring Systems*. ed. by GAHXR Sottolare, H Holden, A review of student models used in intelligent tutoring systems (U.S. Army Research Laboratory, Orlando, 2013)
- PR Pintrich, *The Role of Goal Orientation in Self-regulated Learning*. (Academic Press, 2000)
- S Ritter, JR Anderson, KR Koedinger, A Corbett, Cognitive tutor: applied research in mathematics education. *Psychon. Bull. Rev.* **14**(2), 249–254 (2007)
- R Robson, A Barr, in *Design Recommendations for Intelligent Tutoring Systems*. ed. by GAHXR Sottolare, H Holden, Lowering the barrier to adoption of intelligent tutoring systems through standardization (U.S. Army Research Laboratory, Orlando, 2013)
- V Rus, G Baggett, D Elizabeth, D Franceschetti, M Conley, A Graesser, in *Design Recommendations for Intelligent Tutoring Systems*. ed. by GAHXR Sottolare, H Holden, Towards learner models based on learning progressions (lps) in deeptutor (U.S. Army Research Laboratory, Orlando, 2013)
- R Smallwood, *A Decision Structure for Teaching Machines*. (MIT Press, Cambridge, 1962)
- S Tom, M Dewar, Tracing the trajectory of skill learning with a very large sample of online game players. *Psychological Science.* **25**(2), 511–519 (2014)
- A Usher, N Kober, Student motivation: An overlooked piece of school reform. summary. Center on Education Policy (2012)
- K VanLehn, The behavior of tutoring systems. *Int. J. Artif. Intell. Educ.* **16**(3), 227–265 (2006)
- K VanLehn, The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ. Psychol.* **46**(4), 197–221 (2011)