

Reflections on the KVL Tutoring Framework: Past, Present, and Future

James C. Lester, *North Carolina State University, Raleigh, NC 27695-7535, USA*
lester@csc.ncsu.edu

INTRODUCTION

One of the great pleasures of working in a rapidly changing field is watching it grow and mature. With an auspicious beginning in the late 1960s, the field of AI in Education has evolved from a gleam in its founders' eyes to a well established discipline with an active scientific community that has produced an imposing literature. Now with more than three decades of experience under its belt, it is a good time for us to pause, take stock, and reflect on what works and what does not. We are most fortunate that Kurt VanLehn has done precisely this. He is particularly well suited to the task. Having led many of the key AI in Education projects spanning much of the history of the field, he has an excellent vantage point from which to observe the evolution of intelligent tutoring systems (ITSs) first-hand.

In "The Behavior of Tutoring Systems," VanLehn has moved beyond literature reviews and architectural descriptions to present a unified procedural view of ITSs. Generalizing over the myriad implementation details of the most successful projects, he has distilled more than three decades of research into a generic tutoring procedure defined by two nested loops that iterate over tasks and steps. The two loops also serve as an organizing device for systematically exploring the core design decisions faced by tutoring system developers as they devise task selection, feedback, hinting, assessment, and review functionalities. To ground the discussion, examples are drawn from six landmark tutoring systems: Steve, the Algebra Cognitive Tutor, Andes, Sherlock, AutoTutor, and SQL-Tutor.

The KVL Framework¹ concisely articulates the collective wisdom of the AI in Education literature as it now stands. Considering the breadth of the research it summarizes, its appearance invites speculation about how it would have been perceived when the field was founded, how it can serve as a guide for scientists and practitioners in the present, and how it will be seen by future researchers. For purposes of discussion, we shall take the year of the framework's creation (2005) as the "bisecting year" between the founding of the field 35 years ago (the year in which Scholar was introduced) and the year 2040, 35 years from now. We first take an historical perspective and consider how the founders of the field might have reacted to this synthesis. We then take a contemporary view and argue that the framework should be employed as a blueprint for today's researchers and developers. We conclude with an exploration of how it may be seen by future scientists with another 35 years' experience designing, building, and deploying tutoring systems.

¹ "KVL" is suggested by its designer's initials.

AN HISTORICAL PERSPECTIVE

Had a team of ITS researchers in the late 1960s and early 1970s somehow been given the opportunity to study the KVL tutoring system framework, how would they react? Assuming (1) that they accepted the framework was authored by a recognized authority some thirty years in the future, (2) that they were able to understand the framework, which features a multitude of unfamiliar technologies, and (3) that we can dispense with all of the niggling counterfactual issues, what would they be pleased with? What would they be skeptical of and what disappointments would it hold? What surprises would it offer?

One can imagine the team would be thrilled they had participated in founding a field that would be flourishing in the new millennium. All of the design recommendations offered by the KVL framework have scholarly underpinnings that are substantial, evidencing an active research community. Our 1970s crew would also be impressed with the significant advances in our understanding of learning mechanisms. Based on a sizable literature characterized by a solid theoretical framework and serious empirical studies, the KVL framework would present itself as clearly based on hard science. Featuring very specific recommendations, e.g., under what conditions should hints be given, how should hints be given, what types of hint sequencing is most effective, etc., the framework is in effect a "manual" for how one should design tutoring systems. Critically, the manual's recommendations are principally based on cognitively informed computational approaches, e.g., macroadaptation for controlling the outer loop's task selection activities by explicitly modeling the student's knowledge components to be exercised.

With respect to the technologies dominating the framework, one can only imagine that early ITS researchers would be extraordinarily impressed with two families of developments. First, they would likely marvel at the sophistication of the foundational inference approaches employed. They would probably be very comfortable with the central role played by rule-based problem solving in many aspects of tutoring behaviors, but they would perhaps be surprised at plan-based approaches to the outer loop. They would likely be amazed at the predominance of probabilistic and decision-theoretic approaches, particularly in diagnosis. Second, they would surely marvel at the UI developments. Merely seeing the GUIs (presented in a wonderful appendix) of the Algebra Cognitive Tutor, Andes, AutoTutor, and SQL-Tutor would have to be quite an experience. Seeing Steve inhabiting a VR world in which he physically demonstrates the operation of an air compressor would have to be nearly unfathomable.

Some of the concerns raised by discussion of the KVL framework might seem foreign. For example, while gaming issues (e.g., help abuse) were probably known to our 1970s team, it is not clear that they would have been thought to pose a particularly serious problem. Some issues might seem perplexing. For example, caveats about becoming overzealous about student modeling might be difficult to appreciate without having had the last two decades of research experience. Perhaps some issues that seem apparent to us with the benefit of hindsight, such as distinguishing between slips and conceptual errors, would be curious for early ITS researchers.

Despite the sense of wonder with which they would surely greet the state of knowledge coalesced in the framework, two aspects might be concerning to our team: the lack of ill-defined task domains and the dearth of fielded systems. First, they might be a bit taken aback by the narrowness of the task domains. Without exception, the subject matter of the tutors used to illustrate the framework is uniformly well defined. Focusing on mathematics (algebra), science (physics), programming (SQL), and machine operation and diagnosis (electronics, air compressors), the framework is based on

tutoring system research that has emphasized task domains that are relatively easy to formalize. While it is doubtful they would be troubled by this work – one expects they would embrace it – they might nevertheless be surprised that a field that got its start with tutoring South American geography would have to such a great extent de-emphasized tutoring in the social sciences, language learning, and what we would today refer to as "soft skills." In the end, they might wonder if the framework had simply neglected large bodies of work on tutoring in these other areas, but they would by and large be wrong about this. There are of course exceptions, but in fact the framework's definition and the focus of its discussion are generally representative of how the field has evolved.

Our team might also be perplexed at the lack of fielded systems. With the notable exceptions of the Algebra Cognitive Tutor and SQL-Tutor, the KVL framework is based on a collection of tutoring systems that are for the most part research systems. While they are more than prototypes and most have been empirically studied (many of them extensively so, e.g., AutoTutor and Andes), the tutoring systems that serve as the basis for the discussion are not in widespread use. Some of our 1970s team might have been more pessimistic, but a number of them probably expected that after three decades of work, tutoring systems would be widely used if not ubiquitous. We shall revisit this issue when we move to the future (2040) and find that the problem has happily been resolved.

A CONTEMPORARY PERSPECTIVE

Fast forwarding 35 years to the present, the appearance of the KVL framework marks the arrival of AI in Education as a mature field. Because the framework articulates a generic model that is both accurately descriptive and effectively prescriptive, it is evident that we have achieved a degree of clarity that was certainly not present a decade ago and perhaps not even as recent as a few years ago. Rather than providing a comprehensive top-to-bottom architectural description, it presents a clearly defined double-loop procedure that specifies a canonical set of tutoring functionalities. It is a great resource for the AI in Education community and will undoubtedly be useful for many years to come.

Because the framework is fundamentally an algorithm, it is appropriate to analyze it as such. First, we must ask if it is correct. Verification in this case is very much an empirical matter, but because the framework is based on a large literature characterized by serious formal studies, by and large the answer will be in the affirmative. Of course, correctness is to a large degree determined not just by the framework itself but also by how the services associated with its black boxes are defined. One can distinguish two versions of the question: the engineering question (Does it tutor effectively?) and the cognitive fidelity question (Does it tutor like humans tutor?). Services can be chosen to optimize for either, and no doubt they will sometimes overlap. Second, we must ask if the framework is efficient. Tutoring has been considered an "AI-Complete" task for many years, so it comes as no surprise that approximation techniques must by necessity be adopted. That said, the question of efficiency is very much a practical one because tutoring is inherently interactive. The answer to the efficiency question is perhaps best answered by looking at the response times of the example tutors from which the framework has been abstracted, and again the answer is for the most part in the affirmative.

The framework serves as an excellent development guide for the novice tutoring system designer. Its discussion is at a level of abstraction that manages to walk the fine line between ethereal desiderata that would be too general to be of use and nuts-and-bolts implementation details that would overwhelm the reader. Focusing on core issues in learning and teaching rather than on more

peripheral concerns such as interface design or infrastructure (e.g., web delivery, databases of student records), it hides unnecessary details by presenting key services as black boxes. It supplies just enough theoretical background by introducing notions of transfer, learning curves, and meta-cognitive skills as needed, and briefly touches on learning styles and preferences. It is also careful to make important distinctions, e.g., distinguishing between UI actions and problem-solving steps, that might otherwise be obscured by a breezier discussion. For those new to ITSs but familiar with CBT, it provides a gentle introduction to ITSs by deftly casting ITSs as tutoring systems that have both an outer and an inner loop and CBT as tutoring systems that have only an outer loop. This is a helpful distinction that will smooth the transition for those less familiar with the AI in Education literature.

Replete with specific operational advice about how to deal with the major issues faced by tutoring system designers, the framework mindfully attends to the needs of the practitioner. For example, it cautions that despite the lure of mastery learning, adopting such an approach to task selection is sure to conflict with the practical demands of class-paced courses. It provides clearly specified criteria for which problem-solving step to hint: "The step must be correct. The hinted step should not have been done already by the student. Instructors' preferences should be honored. If the student has a plan for solving the problem or even just for the next step, then the tutor should hint the step that the student is trying to complete." With respect to fine-grained assessment, it recommends IRT if not all learning events are equally difficult because "success on easy learning events provides less evidence of competence than success on difficult learning events." To deal with help abuse, it advises developers to "insert a delay between the time that a bottom-out hint is requested and the time that it is displayed." It frequently notes trade-offs to be considered, such as the one "between the freedom the tutoring system gives to the student and its ability to diagnose the student's failures."

The framework so clearly spells out its recommendations that it should serve as the foundation for standardization efforts. First, to the extent possible, the framework clearly defines its terminology. Introducing the notions of *task domain*, *task*, *step*, *knowledge component*, and *learning event*, each accompanied by many examples from the example tutoring systems, it even includes a lengthy, politely apologetic notion of what it means to be *incorrect*. Second, the tutoring algorithm is a kind of "procedural reference architecture," which is also well specified. For each procedure, e.g., determining when to give error-specific feedback, the framework provides specific advice about the issues bearing on the decision, again accompanied by detailed examples drawn from the literature. While it is sometimes argued that it is too early in the history of tutoring systems to move towards standardization, the framework offers an existence proof that we may now be ready for such a move, and it could be used to combat the scalability issues that have long plagued the field.

The best surveys are not simply summaries: they establish a research agenda. In this light, the framework should serve as a roadmap for AI in Education researchers. The call to arms is explicit in some cases. Along the way it offers signposts, such as, "Research is needed in order to determine the pedagogical benefits of using the student's plans and preferences during step generation," and, "The field would benefit from having a greater variety of hinting techniques to choose from, and from empirical studies comparing their effectiveness." One of the framework's greatest services is its implicit establishment of a research agenda. In effect, it defines a multidimensional space of tutoring systems (one dimension per feature functionality), and the community is now well positioned to explore it. The key question it raises is how to most effectively conduct this investigation.

It will be particularly fruitful to conduct a *tutoring system sensitivity analysis* by systematically considering each of the functionalities in the framework and determining its pay-off in learning effectiveness relative to the amount of effort required to provide that functionality. Cautions about

overbuilding the student model are given, and one of the lessons of the pseudo-tutor efforts is that simplicity can fare well. One therefore wonders what other functionalities are the subject of too much attention and which ones deserve intense, focused investigation. Similar analyses need to be conducted with respect to runtime efficiency. For example, it is noted that, "Although hinting is more computationally complex than minimal feedback, it is probably more pedagogically important. Hints are given when students get stuck, and being stuck may be a powerful motivation for learning. Thus, getting hinting right may significantly increase student learning."

A FUTURE PERSPECTIVE

The physicist Neils Bohr is reputed to have remarked that, "It is hard to predict anything, especially the future," a cautionary note to bear in mind as we journey 35 years into the future to 2040. How will the AI in Education community react to the KVL framework then? Will it seem quaint, or worse yet, incorrect? Will it be incomprehensible? Will they be surprised at how accurately it is able to portray tutoring? The smart money is on the latter. Not that there won't be significant conceptual and technological shifts between now and then, but the fundamental procedural structure and its functionalities will perhaps be for the most part unchanged. Because it crystallized the lessons of so many projects into a single, coherent view of how effective tutoring systems can and should behave, it does not seem at all unlikely that the appearance of the framework will be viewed as a watershed moment in the history of the field.

Consider an ITS research team in 2040. To them, the appearance of the framework signified the maturation of the field because it was the first time that the "standard" consensus view of what it means to be a tutoring system was given a clear procedural articulation. Because its algorithms are based on findings that have now been replicated many times and in many settings (laboratory, classroom, and beyond), it has become the classic guidebook on tutoring system construction. While it was devised primarily for well-defined task domains, with a few relatively minor adaptations, it transferred directly to tutoring in the ill-defined task domains that turn out to be much more common in mid-century learning environments. Instead of having been replaced by more recent work, it has become the backdrop against which new exploratory research is undertaken and evaluated. Thus, the most surprising differences are not to be found in what the framework recommends compared to what is now believed but rather in the elimination of obstacles that hampered its delivery in earlier days and the gradual shedding of restrictive assumptions about what tutoring encompasses.

Tutoring systems in 2040 have become part of the fabric of everyday life. The three primary technological problems impeding their widespread use at the beginning of the twenty-first century have long since disappeared. First, the authoring problem was addressed with machine learning techniques. It is now a trivial matter to induce a custom knowledge base for a new task domain and incorporate it into a "KVL shell" whose parameters are quickly tuned to the needs of the learner population and the demands of the context in which it will be delivered. Second, the computational substrate (formerly known as the Internet) on which all communication and computation are delivered has for decades provided the connectivity and bandwidth to support true "anytime, anyplace learning." Third, the natural language problem that eluded solution for many years has been addressed. Tutorial dialogue, which turned out to be a particularly vexing form of the problem rather than a simple special case, is now well understood. Multimodal interfaces utilizing modes once thought exotic (e.g.,

gesture, gaze, body posture) are just as pervasive in tutoring systems as they are in other applications, and they support rich problem-solving activities and their accompanying conversational interactions.

Building on the framework in the intervening 35 years, our team now knows how to design tutoring systems that are both effective and efficient. The latter property, it turns out, was particularly important in the sweeping changes in corporate training in which tutoring systems were first deployed on a broad scale. In addition to effectiveness and efficiency, however, a third property has since become equally important: motivation. While many early investigations considered the relative merits of extrinsic and intrinsic motivation, widespread acceptance of tutoring systems first occurred when intrinsic motivational objectives figured prominently in the design of users' experience. In this same vein, affective modeling, which was once viewed as a peripheral issue of only secondary interest, is now known to be a key factor in crafting successful learning events. Moreover, the social and physical contexts in which learning plays out is known to call for variations on the nested loops. The standard form of the framework is used to realize interactions emulating human one-on-one tutoring, while collaboration-specific techniques for task selection, feedback, hint giving, and assessment are substituted for their single-learner counterparts to support the dynamics of cooperative group interactions that typify many real-world problem-solving situations. Fortunately, the dilemma of how to incorporate tutoring systems seamlessly into classroom activity has long since been resolved, although our crystal ball is alas too cloudy to reveal the precise details of this resolution.

CONCLUDING REMARKS

In the KVL framework we have an unusual design artifact that effectively informs scientists and practitioners alike. One can imagine that it is something akin to what our 1970s predecessors would have envisioned, while at the same time serving the needs of the AI in Education community today and tomorrow. We look forward to the next 35 years and the designer's 2040 follow-up article that is a retrospective on this subject.