Forensic Reasoning and Paleoclimatology: Creating a System That Works

Kenneth M. Anderson KEN.ANDERSON@COLORADO.EDU
Elizabeth Bradley LIZB@COLORADO.EDU
Laura Rassbach de Vesine LAURA.RASSBACH@COLORADO.EDU
Department of Computer Science, University of Colorado, Boulder CO 80309

Marek Zreda MAREK@HWR.ARIZONA.EDU
Chris Zweck CZWECK@HWR.ARIZONA.EDU
Department of Hydrology, University of Arizona, Tucson AZ 85721

Abstract

Human experts in many scientific fields routinely work with data that are heterogeneous, noisy, and/or uncertain, as well as with heuristics that are unproven and with possible conclusions that are contradictory. We present a deployed software system for cosmogenic isotope dating, a domain that is fraught with these difficult issues. This system, which is called ACE ("age calculation engine"), takes as inputs the nuclide densities in a set of rock samples from a landform. It answers the scientific question "What geological processes could have produced this distribution of nuclide concentrations, and over what time scales?" ACE employs an encoded knowledge base of the possible processes that may have acted on that landform in the past, complete with the mathematics of how those processes can affect samples, and it uses a workflow system to encode the computations associated with this scientific analysis. Flexibility and extensibility were critical issues in ACE's design to allow its scientist-users to modify and extend it after its developers were no longer involved. The success of this is evident; the system remains in active use to this day, several years after the development cycle ended, without a single request for help from the geoscientists to the computer science side of the team. The ACE project website has received over 17,000 hits since 2008, including 2500 over the last twelve months. The software (∼20,000 lines of Python code) has been downloaded nearly 600 times as of April 2013, which is a significant number in a research community of a few hundred PI-level scientists.

Keywords: Scientific discovery, discovery informatics, cosmogenic isotope dating, argumentation, scientific workflow, extensibility.

1. Introduction

Science is becoming increasingly challenged by complex reasoning about noisy, heterogeneous data—often too much data, but sometimes not enough. Helping scientists manage those data, and make sense of them, are important challenges for computer science in general and *discovery informatics* in particular. This paper is about the lessons learned during an interdisciplinary collaboration between geoscientists who date landforms and computer scientists who spent five years creating a software tool called ACE that streamlined and enhanced that geological analysis process.

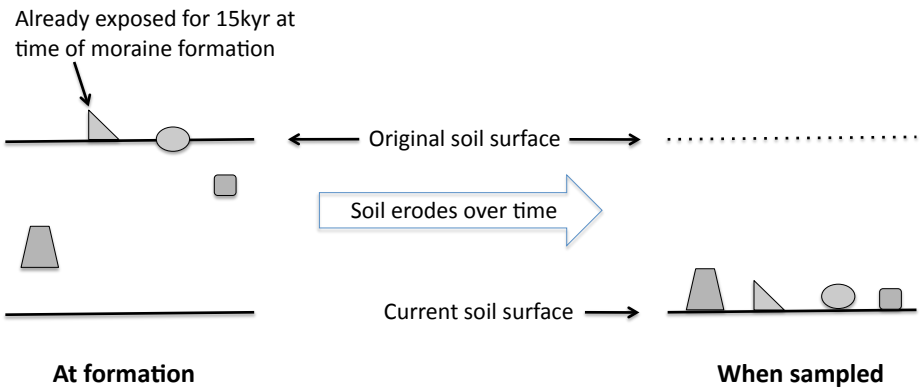

Figure 1. The evolution of a moraine.

Dating landforms is very much like investigating a crime scene: from the information that is available on the surface today, experts must infer what happened in the past. Many landforms are created by a single geological event that happens almost instantaneously in geological time. They then evolve in time in ways that are known, at least in general. Terminal moraines, for example, are formed when a glacier recedes. As these landforms age, subsurface rocks are exposed as the fine matrix around them erodes. A geoscientist sees the situation shown on the right of Figure 1. If she wants to know when that moraine was formed, she needs to reason backwards to determine the initial state of the system—the situation on the left—before she can infer the timeline. This entails figuring out what processes were involved in the evolution of the landform. Geoscientists tackle that problem by making some assumptions about those processes, projecting those assumptions backwards through time and space to the putative formation time of the landform, and iterating the process until the modeling results are consistent with the observations.

A critical challenge here is the measurement of ages and dates. The scientific foundation of the dating technique is the accumulation process of certain rare isotopes produced by secondary cosmic rays. The intensity of cosmic radiation is high enough to penetrate the atmosphere and interact with nuclei of atoms in the top few meters of the solid earth, but not strong enough to penetrate to greater depths. This allows scientists to deduce how long a given rock sample was exposed to the sky. Cosmogenic nuclide dating techniques are ideal for dating surface features such as meteor impact sites, earthquake ruptures, lava flows, alluvial fans, terraces and landforms associated with the retreat of glaciers (Desilets & Zreda, 2003).

ACE is designed to assist expert geoscientists in carrying out this complex scientific reasoning task. The project's development followed a tight feedback loop between the geoscientists on the team (Zweck and Zreda), the AI specialists (Bradley and de Vesine), and the software engineering team headed by Anderson. Working from a collection of exposure ages derived via radioisotope dating from rock samples, its first task is to "calibrate" the isotope dating method. That entails determining a production rate for a particular nuclide from a set of rock samples of known ages—the "calibration set." ACE then uses that production rate to infer the ages of the new, undated

samples. The system incorporates a number of calibration sets for different cosmogenic nuclides and different scientific situations. It uses a workflow engine to make it easy to create, edit, run, and evaluate new cosmogenic dating algorithms. ACE's user creates a new *experiment*—the name for the basic software construct that captures all of the information about a particular run—specifies a nuclide of interest for that experiment, specifies a calibration set for that nuclide, and then runs the calibration workflow in the workflow engine to create the age estimates for the samples. Then, using an encoded knowledge base of rules about mathematical geoscience, ACE's reasoning engine works through scenarios about what processes could have produced that set of sample ages. Finally, ACE reportes the possible scenarios to its scientist user, together with a narration of its reasoning about each scenario.

Our goal in this paper is to offer the insights that we gained, as a result of the ACE project, into the representation and reasoning challenges that arise in forensic science, where timelines are unknown, empirical data are scarce, and controlled experiments are impossible. The challenges that arose during this project ranged from the obvious, such as learning each others' language, to the subtle. The notion that a computer can only do what it is programmed to do, for instance, goes without saying to a computer scientist, but is not at all obvious to those outside the field. That kind of implicit disconnect can easily stymie multidisciplinary research—or even completely derail it. And the traditional artificial-intelligence challenges of representation and reasoning rear their heads here in all their glory. Quality of heuristics and applicability of evidence, for instance, are subtly—but importantly—different from one another in this reasoning task, and data rarely provide absolute proof or refutation of any particular hypothesis. Addressing these challenges required appropriate software-engineering solutions, appropriate artificial-intelligence solutions, and effective integration of the two. This paper focuses on the AI issues: how to design an appropriate reasoning engine and populate its knowledge base with information that accurately captures expert reasoning about paleolandforms. ACE's AI system, which is called Calvin, encapsulates our solutions to these problems. Calvin, described in Section 3 of this paper, is not a standalone system; it operates within ACE's software-engineering framework and inherits many of its fundamental representations from that design. By way of context, Section 2 gives a brief description of that framework and those representations.

2. ACE's Software Engineering Framework

An important goal for this project was to produce a flexible and extensible design environment that would provide geoscientists with software support for cosmogenic nuclide dating. The ACE software architecture, shown in Figure 2, is the basis for this environment. The browser lets geoscientists import, browse, manipulate, and search data sets of rock samples. A suite of tools allows users to create new experiments, apply experiments to imported samples, and visualize results. The workflow engine applies dating algorithms to undated samples to produce calibrated ages. The data model, repository, and workflow engine are described in more detail in the rest of this section; the following section describes how ACE's AI engine, Calvin, reasons about the calibrated ages.

The ACE data model, shown in Figure 3, is a simple conceptual framework that is designed not only to capture the concepts needed to perform cosmogenic nuclide dating, but also to provide a

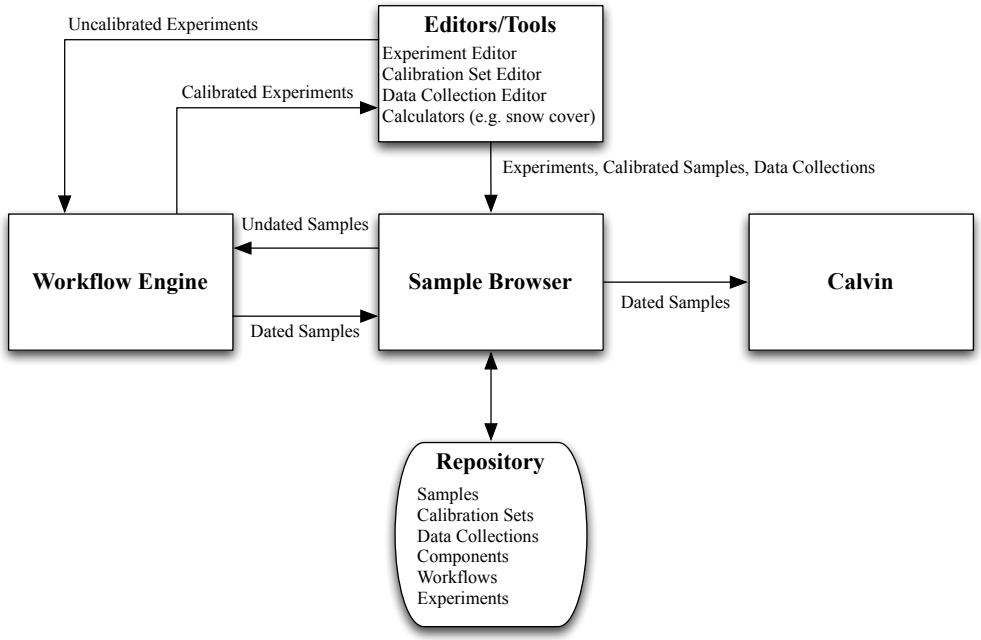

Figure 2. The ACE software architecture.

wide range of customization points for its users to exploit. There are a number of critical elements in this data model that define a large part of ACE's basic knowledge representation and govern the integration of its AI and software-engineering facilities. One or more *sample sets* can be associated with each *landform*. Each sample set contains one or more *samples* (rocks). A sample can contain one or more *nuclides* and has a set of *input attributes* and a set of *output attributes*. Input attributes are either *nuclide-independent* (present for all samples) or *nuclide-dependent* (present only for samples that contain a specific nuclide). Output attributes are calculated by *components*. Samples are processed by *workflows* that consist of one or more *components* and *component placeholders*. Each component consists of a set of *input ports* and a set of *output ports* and performs a specific computation. Components can be connected via their ports to form workflows with sequential, branching, and looping paths.

A component is activated whenever a sample set arrives on one of its input ports. The component iterates over each sample, performs its calculation, and stores the results as one or more output attributes on the sample. Once a component has finished processing a sample, the sample is placed within a set on one of the component's output ports. Each output set is then passed to the components connected to these ports. A component placeholder is a location in a workflow that is associated with a *scaling factor*: a calculation that can influence the results of a workflow. For instance, since the earth's sea level has changed over time, a dating workflow that takes sea-level changes into consideration will produce different exposure ages than a dating workflow that does not. This is handled using a flexible *mode* facility. An *experiment* has a set of input parameters and

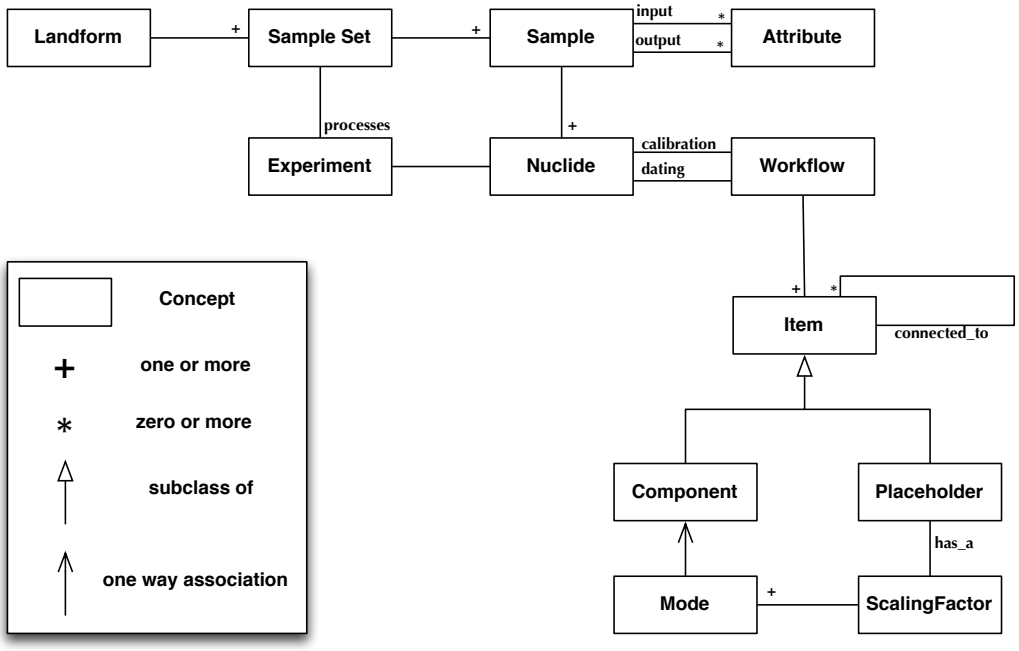

Figure 3. The ACE conceptual framework.

an associated nuclide. Each nuclide has an associated *calibration workflow* and *dating workflow*. These two workflows must both have the same set of scaling factors. Indeed the only difference between these workflows is that a calibration workflow is used to calculate a set of *production rates* that are used by the dating workflow to calculate exposure ages of samples. At that point, the calibration workflow can be applied to a *calibration data set* to produce the production rates needed by the dating workflow.[1] Once the production rates have been calculated, the experiment's dating workflow can then be used to process any sample imported into ACE.

ACE's repository supports a number of data sets—standard scientific constants, sea-level data, sunspot data, metrics related to the periodic table of the elements, and geomagnetic intensity data—as well as information about components, workflows, scaling factors, modes, experiments, nuclides, and samples. A graphical user interface allows users to define the structure of the data sets and import them into the environment from CSV files. To store all of this information, ACE makes use of a simple, extensible repository format that consists of a hierarchical set of folders and text files. This format was chosen because it was human readable, simple to archive, and easy to modify. Scientists nervous about losing data as a result of trying out a new workflow can simply save their repository, quit ACE, and then create copies of the repository by duplicating the directory in the file system for safekeeping. In addition, an input error made while using the graphical user interface can easily be fixed by finding the appropriate file in the repository and editing it with a text editor

1. A calibration data set is simply a sample set whose samples have been dated using another dating technique.

outside of ACE. The flexibility of this repository format lets users adapt the tool to the evolving needs of their research, such as new techniques and new supporting data.

ACE's workflow engine calibrates nuclide production rates and calculates the ages of newly collected samples. A workflow is always executed within the context of a particular experiment, which points at a particular workflow description and a set of specific modes for the scaling factors (such as the influence of sea level or sunspots on the production of a particular nuclide.) The workflow engine reads in the workflow's description—a simple text file that specifies the connections between a set of placeholders or components—and wires up the specified components dynamically at runtime. If the engine encounters a placeholder for a scaling factor (e.g., a token in the workflow description that means "insert sea level calculation here,") it consults the experiment for the selected mode of that factor and inserts the appropriate component, or set of components, into the appropriate spot. At that point, the workflow reads the experiment's selected sample set into memory and passes it to the first component of the workflow, executing the workflow until all samples have been processed.

The flexiblity of this approach makes it extremely powerful. No calculation is hardwired into a dating or calibration workflow. Components can be easily swapped in and out, depending on the needs of the experiment. New workflows can be created by starting from scratch and specifying an all-new set of connections between existing and newly written components—or simply by copying, renaming, and modifying an existing description.

3. Automated Reasoning about Cosmogenic Isotope Dating

The complicated series of calculations involved in dating individual samples just described is only the first step in cosmogenic isotope dating of paleolandforms. The next step is to reason from those results—that is, the exposure ages of the samples—in order to understand the overall history of the landform. This is one of the most challenging and important problems that cosmogenic isotope dating specialists regularly solve. If the exposure ages of all samples overlap, the answer is straightforward: the true age is somewhere in that overlap. This rarely happens, however; rather, the spread of the exposure ages is generally broad and uneven. Faced with this situation, the scientist must construct a geologically meaningful and defensible explanation for the observed spread in order to deduce the true age of the landform.

ACE's reasoning engine, Calvin, automates this complicated, subtle reasoning process (Anderson et al., 2010; Rassbach, 2009; Rassbach, Anderson, & Bradley, 2011; Rassbach & Bradley, 2008; Rassbach et al., 2007; Zweck et al., 2012). This engine is an iterative argumentation system that is based primarily on the Logic of Argumentation of (Krause et al., 1995). Its knowledge base incorporates more than 100 rules, gleaned from an extended knowledge-engineering process involving dozens of geoscientists. Its input is a set of samples that have been dated by the machinery described in the previous section. Its goal is to abduce what process(es), acting over what time periods, could have produced that set of sample ages. Calvin explores this forensic scenario space by enumerating all possible hypotheses about the processes that may have affected the landform, then considers all the evidence for and against each one. Testing of each individual hypothesis involves generating all possible arguments for and against it. To do this, Calvin first finds all of the rules in its knowledge base that apply to that hypothesis, then unifies those rules with the sample ages and

uses that unification to construct a collection of arguments about the associated conclusion (viz., moraine X has been affected by processes Y and Z).

Calvin's design was guided by the nature of the problem at hand, which involves heuristic reasoning with partial support, frequent contradictions, and sparse, noisy data. Most of the explanations that experts find for the apparent age spread of a set of samples come from a short, known list of geologic processes. It is also important to consider the possibility that no process was at work. Despite the relatively small number of candidate processes, constructing these explanations is not a simple matter. Available data are noisy and may not be trustworthy. Different processes may have similar (or cancelling) effects, and multiple processes may be at work. The reasoning involved is heuristic and conclusions do not have absolute confidence; stronger arguments against them may be found and the current best hypothesis overturned. (This is the main distinction between argumentation and classical first-order logic systems or "expert" sytems.) Moreover, these heuristics are often vague or only slightly support their conclusions. For example, several experts informed us that they "prefer the explanation that requires throwing out the least data." This heuristic expresses a preference, not a certainty, but it obviously lends some weight to the discussion. Implementing partial support of this nature has been a traditionally slippery problem for AI (Stenning & van Lambalgen, 2008). Reasoning about sample ages generally involves a fair amount of evidence both for *and against* several processes. Other kinds of contradiction play important roles in this application. Disagreement between experts is to be expected, but the issue does not stop there. During the knowledge-engineering phase of the project, for instance, one geologist said:

> "The thing about inheritance is, it's usually thought about as quantized, not incremental. So in Antarctica, say ice advanced every 100k, so a sample is 20k or 120k, not 21k. So that is one thing that should be commonly true about inheritance, it should reflect events, should be things that date from past advances, should be quantized."

And in the next breath, he said:

> "However, you can convince me you would see a continuum, ... the glacier advanced and quarried exposed material to different depths, so delivering stuff with 100k age but could be at depth and look like only 70k. So if you have stuff with one past event but chop it up and deliver different parts you might not see the quantized aspect."

That is, not only do experts disagree with each other; they sometimes disagree *with themselves*. Handling these different types of contradiction is another challenge for automated reasoning.

Calvin's answers to these challenges begin with the design of its engine. Interviews and joint work sessions made it clear that geoscientists proceed by considering different scenarios for the landform's history and examining all available data for even vague hints to support or refute each scenario. They deal in terms of the overall weight of evidence, rather than in any certainties. The number of solved problems in cosmogenic isotope dating analysis is relatively small, making a machine-learning or case-based approach impractical. The Bayesian approaches that work so well in so many areas of AI were inappropriate here because they are based on forward probabilistic models for reasoning—not forward *mechanistic* models or backward investigatory models, which are the ones used by geoscientists during forensic reasoning. Finally, scientists performing isotope

dating are almost always highly trained experts, so a system that presents answers without detailed support is unlikely to be useful.

For all of these reasons, we clearly needed a defeasible symbolic system with partial support. From the many good options in this area, we settled upon an argumentation-based design for ACE's reasoning engine because geoscientists find it natural to reason via arguments: *for* results with which they agree and *against* those with which they disagree. (One landform dating expert even told us "Well, mostly what we do is argue with each other.") Argumentation, which has a long, albeit almost exclusively theoretical, history in AI, maps naturally onto the method of multiple simultaneous hypotheses (Chamberlain, 1965) that is used in the cosmogenic dating field: scientists construct a list of possible scenarios, then attempt to form complete arguments for and against every hypothesis, and find themselves convinced by the argument with the strongest support. Other traditional AI strategies cannot handle some of the unique ways in which geoscientists reason about cosmogenic isotope data. Experts work with multiple contradictory heuristics at the same time, for instance, and several weak arguments can weaken and/or defeat a strong one, which requires some novel modifications to traditional argumentation strategies—and completely rules out traditional knowledge-based or "expert" systems.

The rules that guide Calvin's operation are Horn clauses $A \Rightarrow C$ ("A implies C"), annotated with associated confidence and quality judgments about the data or the implication. This rule set was the keystone of the AI effort in this project. The design of these rules was critical; noise and partial support, for instance, were addressed by support for hypotheses have variable strength. Noise and partial support, for instance, were addressed by directly annotating rules with an expert judgement of the quality of the knowledge in the rule. Another critical insight in Calvin's rule-design strategy is that not only can specific *knowledge* be more or less certain, but the *evidence used to apply the knowledge in a specific case* may be of variable suitability. For this reason, Calvin uses a rich, multi-level representation to capture experts' confidence in the data and in the conclusions drawn from it, and to propagate that knowledge through the forensic reasoning chain. These design elements are discussed in turn in the remainder of this section; the knowledge engineering process through which these rules were crafted is described in Section 4. More detail on all of this material can be found elsewhere (Rassbach, 2009; Rassbach, Anderson, & Bradley, 2011).

Using a list of the processes that are known to operate upon landforms—erosion, snow cover, and so on—Calvin begins by generating a set of candidate hypotheses about what may have affected the input samples. These hypotheses are constructedn exhaustively from a list of nine typical processes (erosion, inheritance, snow cover, whether a particular sample is an outlier, etc.) that may have affected the landform and the data. It then considers these hypotheses one at a time, building arguments for and against each one using backwards chaining. Because the typical data set in this field is quite small and the list of candidate processes is fairly short, we encountered no need to cull arguments as they are generated, instead allowing Calvin to complete its reasoning chain for every candidate hypothesis. The first step in constructing an argument involves finding all the rules that apply to that hypothesis—i.e., those that refer to the same conclusion. The engine then applies unification to each of these rules, which either produces a new conclusion to consider or generates a comparison to input data. Figure 4 presents a flowchart of Calvin's backwards chaining process. Figure 5 shows an example of the logic flow of Figure 4 as applied to the scenario on page 7. Beside

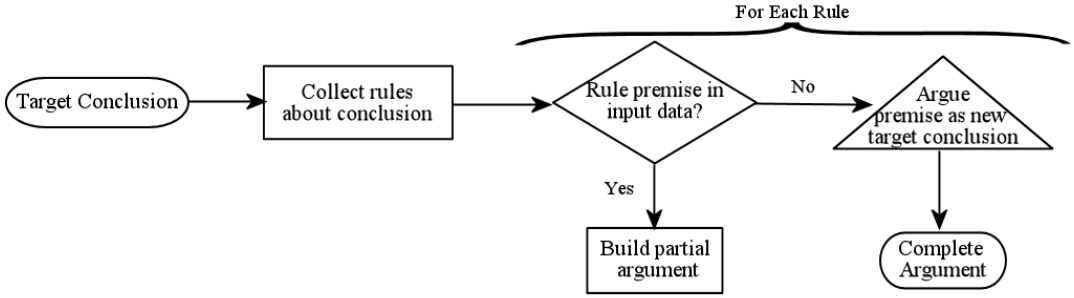

Figure 4. A flowchart of Calvin's backwards-chaining rule-unification process.

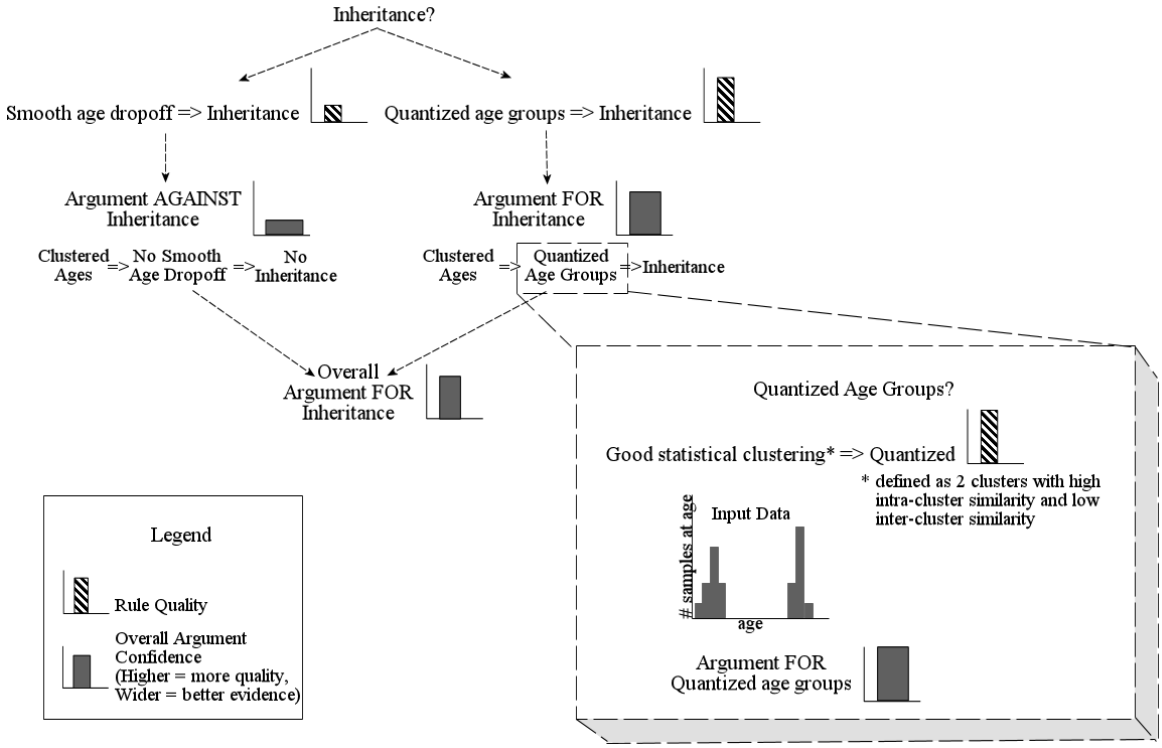

Figure 5. An example of Calvin reasoning about whether or not a particular sample was inherited from an older landform.

each rule is a quality rating recording the expert's expressed confidence in that piece of knowledge. Calvin considers each rule in turn; if its premises cannot be found in the input data, the system then argues about those premises as new conclusions. The engine assigns the resulting arguments a confidence rating that is based on the quality of the rule and the applicability of the evidence to the rule. Finally, Calvin builds an overall argument—in this case in favor of inheritance—from the arguments generated using each rule. The final argument includes any detractors found during the process, so the user can see they have been taken into account. The engine reports an overall confidence in whether inheritance has occurred based on all the contributing arguments, as we describe in more detail shortly.

Every rule in Calvin contains both a conclusion and a template for evidence that supports that conclusion. The primary portion of a rule is an implication of the form $A \Rightarrow C$, where A may be either a single literal or the conjunction (or disjunction) of several literals, and C is the conclusion that A supports. The two contradictory statements on page 7, for example, became two different rules in Calvin's knowledge base: one that looks for a smooth increase in sample ages as evidence for inheritance and one that looks for "quantized" inheritance (defined as highly clustered ages). These rules directly contradict each other; in fact, the example in Figure 5 shows that the same input data leads to both an argument for and an argument against inheritance. Calvin's confidence system lets it sort out this contradiction and thereby reproduce this expert reasoning accurately—i.e., to disagree with itself.

Given a rule $A \Rightarrow C$, Calvin forms an argument—not a proof—for each element in A, then uses those arguments to create an overall argument for C. The representation of an argument contains the rule, the arguments for the antecedents, and a quality rating that expresses expert confidence in the rule. The quality rating of the rule and the strength of the applicable evidence are used to judge the relative and absolute strengths of arguments. Calvin's backwards-chaining engine generally makes no distinction between negative and positive evidence. This is not a valid method in classical logic, where the knowledge that $A \Rightarrow C$ certainly does not imply that $not(A) \Rightarrow not(C)$. However, Calvin's reasoning is intended to mimic that of human experts, who not only apply rules in this negative fashion[2], but even regard it as a sufficiently defensible practice that they discuss it in published reasoning. For example, Jackson et al. (1997) state that, since there is no visual evidence of erosion, it is unlikely in the area under consideration.

Cosmogenic isotope dating experts have firm ideas about confidence in different measurements, processes, and conclusions, and those ideas play important roles in their analyses. Calvin's assignment of quality ratings to rules lets it capture and operationalize this information. When constructing an argument using a rule, the system reasons both about rule quality and about the applicability of the knowledge that is being used. The insight behind this solution is that not only can specific knowledge be well-regarded or more tenuous. The evidence used to apply that knowledge to the conclusion may also be of variable applicability. In the example, the distribution of sample ages might be perfectly linear, somewhat linear, or not at all linear. How closely the actual data matches the template in the rule (in this case, how linearly the samples are actually distributed) determines

2. Consider the following everyday example: A ="natural sunlight" and C ="daytime." Here, both $A \Rightarrow C$ and $not(A) \Rightarrow not(C)$ make sense.

the applicability of the rule in any specific instance. For this reason, Calvin uses a two-dimensional vector with qualitative (not continuous) values to capture both flavors of confidence.

This rich representation of confidence—which is critical to weighing one argument against another—introduces several new research issues: how to set the thresholds and weights, how to weigh the two elements against one another (e.g., a conclusion derived from high-quality evidence and low-quality knowledge versus one where both are of medium quality), and how those levels should be transmuted as the engine maps them through the rules. Thresholds and weights in Calvin's rules were assigned based on data collected in expert interviews, as discussed in Section 4. Assigning a confidence to an overall argument based on the confidence in its parts is particularly problematic when arguments of different strength can be made both for and against a given conclusion. Figure 5 shows a visual representation of how these confidences are combined. Rassbach (2009) provide a full technical discussion and Rassbach et al. (2011) present an extended example. Calvin's final arguments are shown to the user in a way that incorporates all the information used to make the arguments: the top-level conclusions, its overall confidence in each step of the process, and elucidation in cases of controversy (viz., strong evidence both for and against). Figure 6 shows a collection of related screenshots.

4. Knowledge Engineering

One of the biggest challenges in building Calvin was knowledge engineering. This began with the basic design of the reasoning framework. To accomplish this, members of ACE's AI team spent roughly 30 days onsite with the geoscience team members over the course of the first four years of the project. During these joint work sessions, the AI team observed as the geoscience team members worked through a series of cosmogenic isotope dating problems. In the first round of these interactions, the geoscientists presented a somewhat idealized version of the reasoning process. This discussion was limited to (what we later discovered were) extremely unusual projects dealing with many samples and uncommon processes, without explicit discussion that these were unusual cases. This led to an initial AI design that focused on statistical analysis and spatial reasoning about distributions of ages.

As it turned out, however, the average dating project has far too few samples to support this kind of reasoning, so the AI team moved to a qualitative-physics type approach that more naturally encodes expert knowledge like "when you are working with a moraine your first thought is erosion or inheritance." Choosing the basic representations and designing the reasoning system was not straightforward; the reasoning methods used by expert geoscientists were not typical of classic AI systems. Specifically, geoscientists were in general unwilling to commit to a specific explanation for an age distribution without first considering several possibilities. In fact, if possible, they would avoid committing firmly to any explanation. Instead, they favored discussing—in great detail—the evidence surrounding each hypothesis, and then ranking the hypotheses against each other once all the evidence had been considered. Indeed, published papers in this field routinely include a discussion of multiple possible explanations, with evidence for why the selected one was considered the best. The design of Calvin's argumentation engine both reflects and supports this commit-

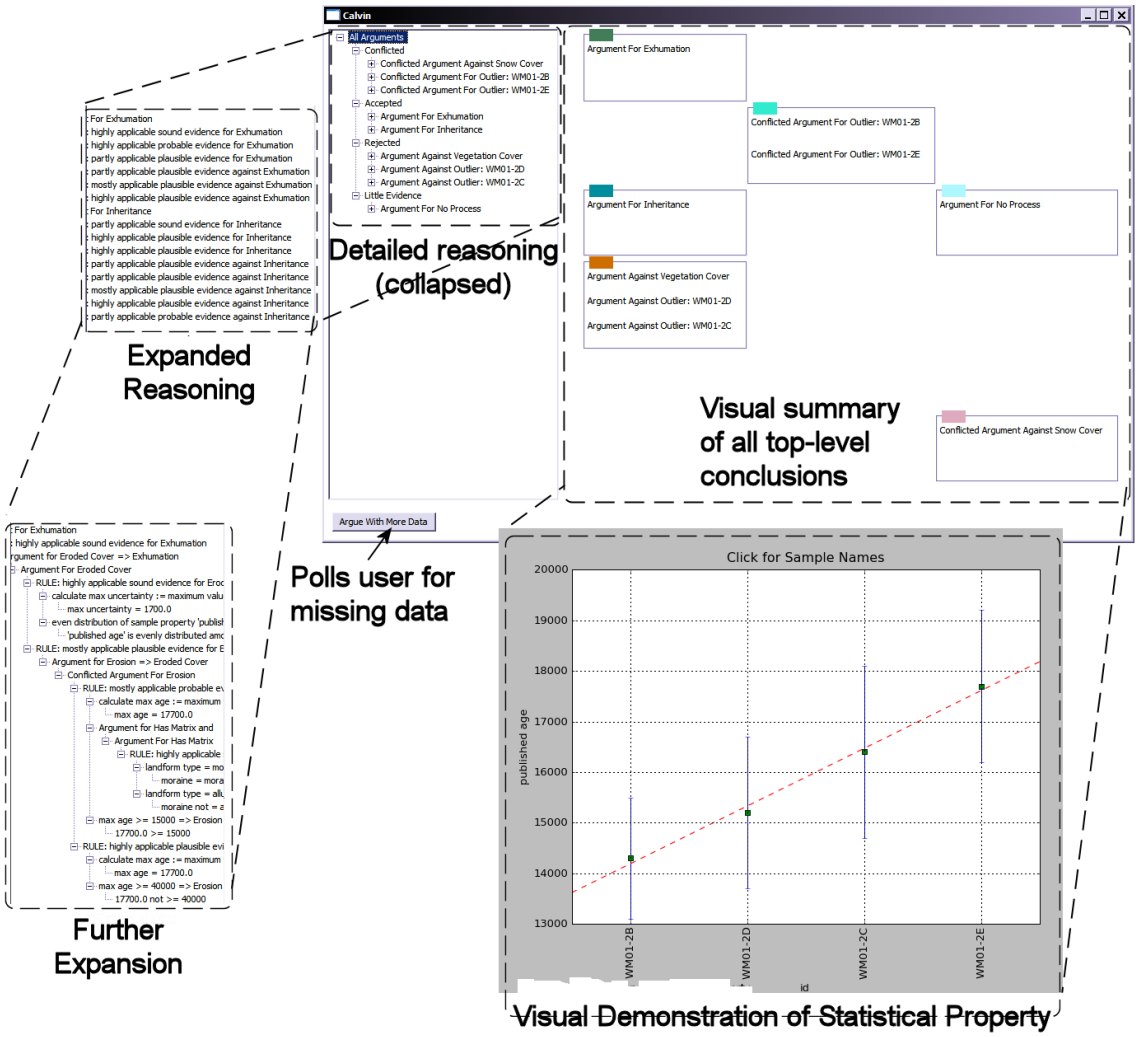

Figure 6. A collection of screenshots from Calvin's interface [after Rassbach (2009)]. Users can drill down to any level of reasoning, letting them explore and understand the system's arguments around any given conclusion.

late style of reasoning: it considers all possiblities, ranking them by strength of evidence, and furthermore lets users engage with the evidence and make their own judgements if they disagree.

Once the design of the engine was complete, the next task was to populate the rule base. The expert knowledge encoded in Calvin's rules was gathered from a total of 22 geoscientists. The joint work sessions described above led to an initial rule base of about 65 rules, including the ones involved in the inheritance example from Figure 5. To broaden and deepen the rule set, the lead AI team member (de Vesine) then conducted two onsite demo/interviews—each spanning almost two days—with geoscientists outside the ACE project team[3]. These interactions fleshed out many of Calvin's existing reasoning chains, such as what percentage of samples must agree to result in a high-confidence argument for 'no process'. They also led to a few surprising discoveries: for instance, that visual observations in the field lend significantly more confidence to conclusions than the AI team had expected. Finally, de Vesine attended the main professional meeting of the geoscience research community—the fall 2008 meeting of the American Geophysical Union—and conducted 11 one-hour demo/interviews with a range of people working actively in cosmogenic isotope dating. Some spoke English as a first language and some did not; some were veterans of the field and others were graduate students or new postdocs. All of the data obtained in these interviews was incorporated into Calvin's knowledge base, which currently includes 108 rules that represent approximately 50 hours (almost 100 transcribed pages) of direct, intensive interviews. Rassbach (2009) presents these transcripts in their entirety. A persistent theme in all of these interviews was how proactively experts acknowledged and emphasized the level of disagreement in the field. Not only were they anxious to point out likely rebuttals that other experts would make to their theories, but they also introduced, without prompting, scenarios where they would need sufficient evidence to overrule a colleague's conclusions about a landform.

Note that although Calvin is a knowledge-based system, it does not share one of the classic drawbacks associated with symbolic reasoning systems: the flexibility of the hypothesis-weighing model means that contradictory rules can be added without breaking the system. In fact, asking Calvin to reason from a knowledge base containing two directly contradictory rules will simply result in a report of controversy about the conclusion. Calvin's design does not, of course, address the other classic problem of the cost of obtaining enough expert knowledge to create a comprehensive rule base.

5. Evaluation

Our evaluation of ACE proceeded in several steps. Preliminary feedback was gleaned from the two on-site demo/interviews mentioned in Section 4, where the outside geoscience experts made many statements that helped to confirm the validity of Calvin's basic design: i.e., that the results were similar in both structure and content to the reasoning of domain experts. This positive feedback was echoed in the 11 AGU interviews as well, as in:

de Vesine: So the paper has a verified explanation?

3. These experts were recommended by the lead ACE geoscientist (Zreda), based on his knowledge of the field.

> **Geologist:** We found some evidence that supported this . . . You look for evidence that supports or falsifies individual hypotheses.

and later in the same interview:

> **de Vesine:** I wanted to ask about how you teach this analysis to new grad students and what they get wrong while learning.
>
> **Geologist:** [long pause], that's not an easy question actually. . . . it's very ad hoc . . . they don't understand [some data] at all so you sit them down and talk about them and give them things they could do to test hypotheses: graphs to make and data to collect and we work through it until we are satisfied we have come up with the most reasonable hypothesis.

Our final assessment of ACE's AI facilities involved using it to reproduce published work: that is, feeding Calvin the data in a published paper and comparing its results against the claims made in that paper. As mentioned above, experts publish a portion of their qualitative reasoning about a landform when they publish a new data set. While this presentation is usually incomplete due to space limitations and the desire to maintain reader interest, it typically includes information about both rejected and accepted conclusions. This matter is useful for determining if Calvin's recall is sufficiently high for the most important arguments. For this comparison, we found 25 randomly selected papers that appeared from their titles to deal with cosmogenic isotope dating. Of these, 18 actually discussed one or more isotope dating problems in any detail. These publications included a large cross section of authors and different isotopes, they and spanned about ten years, providing a broad basis of comparison. For each of these papers, we extracted every statement that made an assertion, such as this example from Jackson et al. (1997):

> Erratic A (sample AE95110101) yielded an age . . . almost four times older than the next oldest age. This age is clearly anomalous . . . [t]he most likely explanation for this anomalous age is exposure to cosmic radiation prior to glacial transportation.

We then converted these statements into a form that more closely matched Calvin's terminology. This involved identifying the conclusion argued for in the statement, estimating a level of confidence from terms such as 'clearly' and 'possibly,' and extracting the evidence used in the statement to support the conclusion. Then we converted the terms in the evidence and conclusion into Calvin's terminology: for example, 'inheritance' instead of 'prior exposure to cosmic radiation,' which is the definition of inheritance. We cross-validated these results by asking two other people (not experts in isotope dating) to perform the same conversions. We then entered all of the data in the paper, ran Calvin, and compared its output to the converted arguments. Calvin produced between seven and 161 arguments for each paper, averaging about 50 arguments per paper. This is almost twice as many arguments as the papers themselves contained, as Calvin produces full arguments even for conclusions that a human expert would consider obviously wrong (and were therefore not addressed in the papers). For arguments that were addressed in the original papers, it closely reproduced the authors' arguments 62.7% of the time and produced similar arguments a further 26.1% of the

time. In many cases, the similarity was striking, especially when the authors of the paper expressed significant doubt about their conclusions. Rassbach (2009) reports detailed results.

Perhaps the most interesting cases in this evaluation process were when Calvin produced an argument that did not appear in the original paper. When examining Ballantyne et al. (1998), for instance, the system argued that exhumation was at work. The main evidence for this was a disagreement with ages determined for this landform via other methods. To judge these results, we asked a domain expert to assess Calvin's new argument. He responded:

> I think I see both sides here. From the results, the fact that the ages are younger than the C14 data means that exhumation should be taken very seriously . . . there is not much in the way of material that could bury them. However the peaks themselves are eroding . . .

In this expert's opinion, then, the lack of explicit discussion of exhumation in Ballantyne et al. was a major oversight. Although Calvin does not give exactly the same argument, it found a major gap in the reasoning published by these authors.

From the software-engineering standpoint, the key design goals of ACE were utility and flexibility: to create a system that was useful to the scientists, that made no assumptions about the types of nuclides, scaling factors, algorithms, etc. used by a research group, and that was easy to modify. To this end, ACE's data model, repository, and workflow engine were designed so as to maximize simplicity, flexibility, and extensibility. We also made design choices that lowered the barriers to adoption of the system. This ensured that the system was not only easy to use, but would remain usable long after it was first developed—and not just by the research group involved in its creation.

As a preliminary evaluation of ACE against these goals, we compared its capabilities to the functionality of the software that it was built to replace. At the beginning of this project, the geoscience side of the ACE team relied on an Excel spreadsheet that implemented a single cosmogenic dating technique for samples containing the nuclide ^{36}Cl. With ACE, they were able to quickly design and run an experiment that exactly mimics the functionality of the original spreadsheet. That configuration, however, is just *one* of an almost unlimited set of experiments that can be specified, executed, and analyzed by the system. The inherent flexibility of ACE's workflow-based design allows any number of calibration and dating algorithms to be constructed, each of which can support multiple scaling factors, parameters, and variables. In addition, it can store multiple sets of samples and can apply a single dating workflow to them simultaneously. Finally, ACE lets users plot, analyze, and export the results. This stands as a solid proof-of-concept that the system has transcended the limitations of the previous state of the art in this field—in particular, that it provides the flexibility that is needed by geoscientists to perform cosmogenic dating research.

ACE's extensive and long-term use by the forensic paleoclimatology research community is a testament to the success of our goals in building it. Indeed, the system remains in active use to this day, years after the development cycle ended, without a single request for help from its geoscientist users. The ACE project website has received over 17,000 hits since 2008, including 2500 over the last 12 months. The software has been downloaded 589 times as of April 2013, which is a significant number in a research community of a few hundred PI-level scientists. Given the state of cosmogenic nuclide dating software prior to the development of ACE, we believe that our

contributions have enabled new science, allowing progress in a domain that previously was hindered by inflexible computational tools that offered no automated reasoning assistance.

6. Related Work

Three other software programs are available to date landforms using cosmogenic nuclides. The CRONUS-Earth series of calculators compute sample ages of ^{10}Be, ^{3}He, ^{36}Cl, and ^{26}Al using an online web server operating as a front end for a set of MATLAB routines[4]. An issue with these tools is that geoscientists must be willing to submit samples relating to unpublished research to a web service managed by an unaffiliated research group. Another tool, Cosmocalc, can be downloaded locally to avoid that privacy issue[5]. In 2012, a new automated reasoning tool for cosmogenic isotope dating was released by a team at Dartmouth (Applegate et al., 2012), but it operates only on moraines. ACE matches and exceeds the functionality provided by these systems; its support for extensibility (Anderson et al., 2007) lets it offer features not found in any other software package for cosmogenic dating.

Our work is closely associated with research on design environments (Winograd, 1995) and domain-specific software architectures (Tracz, 1995). Design environments are characterized not only by functionality to perform a task, but also by the services they provide to reflect on the task itself. For example, ACE not only calculates ages based on measured inventories of cosmogenic nuclides, but it also provides services to analyze and compare cosmogenic dating techniques. Design environments have also been used to help users understand and apply standardized techniques within a domain (Garlan, Allen, & Ockerbloom, 1994). ACE does this, letting experts teach novices how to customize cosmogenic dating techniques. None of the other systems offer this support.

Within the field of argumentation as a reasoning framework, two major branches have been established (Reed & Grasso, 2007). One views argumentation systems from a dialectical viewpoint, largely based on Dung (1995). Systems using this variety of argumentation include Hunter (2004) and Prakken (1996). The other branch approaches argumentation as an extension to and improvement on first-order logic: this is largely based on the Logic of Argumentation of Krause et al. (1995). Systems following this tradition have been reported by Cayrol and Lagasquie-Schiex (2003) and by Morge and Mancarella (2007). Calvin draws on both argumentation paradigms: primarily, its rules and confidence system are drawn from the Logic of Argumentation, but the perspective of treating arguments as trees and the method of performing first a local and then a global argument assessment are more common in the dialectical branch. Our definition and use of confidence was also informed by Farley (1997) and by Amgoud et al. (2005). A distinction is that Calvin is a fully implemented and deployed system—a surprisingly rare thing in the argumentation literature

Although argumentation is the variety of reasoning that best matches the structure of expert reasoning and communication in this domain, there are many kinds of non-monotonic logics that we considered for Calvin's design. These include circumscription (McCarthy, 1980; McCarthy, 1986), default reasoning (Reiter, 1980), and other forms of nonmonotonic reasoning (Gaines, 1996; Pereira, Aparicio, & Alferes, 1991; Pollock, 1994). Although each is useful, none solves all of the

4. `hess.ess.washington.edu/math` and `www.cronuscalculators.nmt.edu`
5. `cosmocalc.googlepages.com`

issues involved in cosmogenic isotope dating analysis. In particular, most varieties of nonmonotonic reasoning—besides argumentation—require explicit definitions of when conclusions may be withdrawn and have problems with the issues of partial support and defeat. Furthermore, many nonmonotonic logics are unsound in some way, making their use problematic (Etherington, Kraus, & Perlis, 1991). These properties made them unsuitable for our needs.

The knowledge engineering"bottleneck" mentioned at various points in this paper is a well-known phenomenon, dating back more than 30 years (Gaines & Shaw, 1993; Self, 1984). Very recently, Mozina et al. (2008) have proposed to combine argumentation and machine learning in order to help experts articulate their knowledge and uncover tacit concepts, and thus aid in the construction of knowledge bases.

One can also view cosmogenic isotope reasoning as a diagnostic process. Model-based diagnosis—e.g., (Lucas, 1997; Santos, 1991; Struss, 2004)—is inappropriate for Calvin because complete models of most geologic processes simply do not exist. Other approaches to diagnosis do exist, including systems that handle contradiction (Doyle, 1983; Gaines, 1996), but their architectures use "absolute" rules, which are not appropriate in our domain.

7. Conclusion

ACE represents a significant advance in cosmogenic dating software. It lets geoscientists apply and evaluate existing dating techniques—and design new ones. It is based on a comprehensive conceptual framework that accurately models all aspects of the the cosmogenic dating process. Its software architecture is flexible, extensible, and designed to lower the barriers for new users. Its argumentation-based reasoning engine captures and operationalizes the knowledge of more than two dozen experts in the field. The system incorporates a rich confidence framework that captures the reasoning of real scientists in a useful way. These features let ACE work through scientific scenarios automatically, thereby saving its user from the time and aggravation of an exhaustive exploration of the hypothesis space—and occasionally finding explanations that they otherwise would have missed. While its design is similar to (and inspired by) previous work, it is more suitable to implementing the multiple-hypothesis reasoning that is employed by experts in cosmogenic isotope dating.

The computer scientists in the ACE team have now moved on to applying this multidisciplinary research approach to a different domain: the analysis of ice and ocean-sediment cores. From these cores—depthwise sequences of information—a geoscientist interested in a past climate event must first infer the timeline for the data: a curve called an *age model* that relates the depth in the core to the age of the material at that point. This is the first critical step in reasoning about the science of the events that produced the core. Like ACE, this new project (entitled CSCIENCE) brings computer scientists and geoscientists together around the goal of producing a software system that enables scientific progress in a challenging application domain.

There are many challenges in the CSCIENCE project, some familiar (tools and data sets are stored in spreadsheets) and some requiring fundamental new work in the areas of 'big data' and automated reasoning. Assumptions about how ice and ocean-sediment cores are created have multiple permutations—leading to a potential explosion in the number of age models to generate and evaluate—and the data involved are very different. ACE's reasoning engine worked with two num-

bers (mean and standard deviation) for each of a few dozen rock samples taken from a single landform that was formed instantaneously in geological time, then influenced by a small list of candidate processes that involved no unknown parameters. CSCIENCE's data sets are thousands of times larger, and their ordered nature allows reasoning about *continuous* events, not just episodic ones, which is a much harder and more general problem. Nevertheless, we believe that the lessons learned in designing, developing, and evaluating ACE will serve us well in tackling this complex new application domain.

Acknowledgements

This material is based upon work sponsored by the NSF under Grants ATM-0325812 and ATM-0325929. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

Amgoud, L., Bonnefon, J., & Prade, H. (2005). An argumentation-based approach to multiple criteria decision. *Proceedings of the Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty* (pp. 269–280). Barcelona, Spain: Springer Lecture Notes in Computer Science.

Anderson, K., Bradley, E., Rassbach, L., Zweck, C., & Zreda, M. (2010). End-to-end support for paleolandform dating. In N. Adams, M. Berthold, & P. Cohen (Eds.), *Advances in intelligent data analysis IX*, Vol. 6065, 171–183. Springer Lecture Notes in Computer Science.

Anderson, K., Bradley, E., Zreda, M., Rassbach, L., Zweck, C., & Sheehan, E. (2007). ACE: Age calculation engine: A design environment for cosmogenic dating techniques. *Proceedings of the First International Conference on Advanced Engineering Computing and Applications in Sciences* (pp. 39–48). Papeete, Tahiti.

Applegate, P., Urbana, N., Kellera, K., Lowell, R., Laabs, B., Kelly, M., & Alley, R. (2012). Improved moraine age interpretations through explicit matching of geomorphic process models to cosmogenic nuclide measurements from single landforms. *Quaternary Research*, *77*, 293–304.

Ballantyne, C., Stone, J., & Fifield, L. (1998). Cosmogenic Cl-36 dating of postglacial landsliding at The Storr, Isle of Skye, Scotland. *The Holocene*, *8*, 347–351.

Cayrol, C., & Lagasquie-Schiex, M.-C. (2003). Gradual acceptability in argumentation systems. *Proceedings of the Third International Workshop on Computational Models of Natural Argument* (pp. 55–58). Acapulco Guerrero, Mexico: University of Liverpool.

Chamberlain, T. (1965). The method of multiple working hypotheses. *Science*, *148*, 754–759. Reprint of 1890 *Science* article.

Desilets, D., & Zreda, M. (2003). Spatial and temporal distribution of secondary cosmic-ray nucleon intensities and applications to in-situ cosmogenic dating. *Earth and Planetary Science Letters*, *206*, 21–42.

Doyle, J. (1983). Methodological simplicity in expert system construction: The case of judgments and reasoned assumptions. *AI Magazine*, *4*, 39–43.

Dung, P. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and N-person games. *Artificial Intelligence*, *77*, 321–357.

Etherington, D., Kraus, S., & Perlis, D. (1991). Nonmonotonicity and the scope of reasoning. *Artificial Intelligence*, *52*, 221–261.

Farley, A. (1997). Qualitative argumentation. *Proceedings of the Eleventh International Workshop on Qualitative Reasoning* (pp. 89–96). Cortona Italy: Istituto di Analisi Numerica, Pavia.

Gaines, B. (1996). Transforming rules and trees into comprehensible knowledge structures. *Advances in Knowledge Discovery and Data Mining* (pp. 205–228). AAAI Press.

Gaines, B., & Shaw, M. (1993). Eliciting knowledge and transferring it effectively to a knowledge-based system. *IEEE Transactions on Knowledge and Data Engineering*, *5*, 4–14.

Garlan, D., Allen, R., & Ockerbloom, J. (1994). Exploiting style in architectural design environments. *Proceedings of the Second ACM SIGSOFT Symposium on Foundations of Software Engineering* (pp. 175–188). New Orleans, LA, USA: ACM.

Hunter, A. (2004). Making argumentation more believable. *Proceedings of the Nineteenth National Conference on Artifical Intelligence (AAAI'04)* (pp. 269–274). San Jose, California, USA: AAAI Press.

Jackson, L., Phillips, F., Shimamura, K., & Little, E. (1997). Cosmogenic 36Cl dating of the Foothills erratics train, Alberta, Canada. *Geology*, *25*, 195–198.

Krause, P., Ambler, S., Elvang-Goransson, M., & Fox, J. (1995). A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, *11*, 113–131.

Lucas, P. (1997). Symbolic diagnosis and its formalisation. *The Knowledge Engineering Review*, *12*, 109–146.

McCarthy, J. (1980). Circumscription—A form of non-monotonic reasoning. *Artificial Intelligence*, *13*, 27–39.

McCarthy, J. (1986). Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, *26*, 89–116.

Morge, M., & Mancarella, P. (2007). The hedgehog and the fox: An argumentation-based decision support system. *Proceedings of the Fourth International Workshop on Argumentation in Multi-Agent Systems* (pp. 114–131). Honolulu, Hawaii, USA: Springer-Verlag.

Mozina, M., Guid, M., Krivec, J., Sadikov, A., & Bratko, I. (2008). Fighting knowledge acquisition bottleneck with argument based machine learning. *Proceedings of the Eighteenth European Conference on Artificial Intelligence* (pp. 234–238). Patras, Greece: IOS Press Amsterdam.

Pereira, L., Aparicio, J., & Alferes, J. (1991). Nonmonotonic reasoning with well founded semantics. *Proceedings of the Eighth International Logic Programming Conference* (pp. 475–489). Paris, France: MIT Press.

Pollock, J. (1994). Justification and defeat. *Artificial Intelligence*, *67*, 377–407.

Prakken, H. (1996). Dialectical proof theory for defeasible argumentation with defeasible priorities. *Formal Models of Agents: ESPRIT Project ModelAge Final Workshop Selected Papers* (pp. 202–215). Springer Lecture Notes in Computer Science.

Rassbach, L. (2009). *Calvin: Producing expert arguments about geological history*. Doctoral dissertation, Department of Computer Science, University of Colorado.

Rassbach, L., Anderson, K., & Bradley, E. (2011). Providing decision support for cosmogenic isotope dating. *AI Magazine*, *32*, 69–78.

Rassbach, L., & Bradley, E. (2008). Challenges in presenting argumentation results. *Proceedings of the Twenty-Second International Workshop on Qualitative Reasoning about Physical Systems* (pp. 123–125). Boulder, Colorado, USA: AAAI Press.

Rassbach, L., Bradley, E., Anderson, K., Zreda, M., & Zweck, C. (2007). Arguing about radioisotope dating. *Proceedings of the Twenty-First International Workshop on Qualitative Reasoning about Physical Systems* (pp. 31–40). Aberystwyth, Wales: AAAI Press.

Reed, C., & Grasso, F. (2007). Recent advances in computational models of argument. *International Journal of Intelligent Systems*, *22*, 1–15.

Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, *13*, 81–132.

Santos, E. (1991). On the generation of alternative explanations with implications for belief revision. *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence* (pp. 339–347). Los Angeles, California, USA.

Self, J. (1984). Building expert systems. *Robotica*, *2*, 119–119.

Stenning, K., & van Lambalgen, M. (2008). *Human reasoning and cognitive science*. Cambridge, MA: MIT Press.

Struss, P. (2004). Deviation models revisited. *Proceedings of the Eighteenth International Workshop on Qualitative Reasoning* (pp. 56–62). Evanston, Illinois, USA: AAAI Press.

Tracz, W. (1995). DSSA (domain-specific software architecture): Pedagogical example. *ACM SIGSOFT Software Engineering Notes*, *20*, 49–62.

Winograd, T. (1995). From programming environments to environments for designing. *Communications of the ACM*, *38*, 65–74.

Zweck, C., Zreda, M., Anderson, K., & Bradley, E. (2012). The theoretical basis for ACE, an age calculation engine for cosmogenic nuclides. *Chemical Geology*, *291*, 199–205.