Solomon: A Next-Generation QA System

* Summary Report *

Selmer Bringsjord (PI)
Micah Clark
Daniel Werner
Andrew Shilliday
Department of Cognitive Science
Department of Computer Science
Rensselaer AI & Reasoning (RAIR) Laboratory:
http://www.cogsci.rpi.edu/research/rair/
Troy NY 12180 USA
selmer@rpi.edu
clarkm5@rpi.edu
werned@rpi.edu
shilla@rpi.edu

DRAFT — December 21, 2007
Contents

1 Introduction & Motivation 3

2 Six Distinguishing Attributes of the Solomon System 4

3 Prototype of the Solomon System 7
   3.1 Knowledge Acquisition via Reading 7
   3.2 Human — Computer Collaboration via Conversation 13
   3.3 Natural Suppositional Reasoning 16
   3.4 Defensible Answers and Rational Justifications 17
   3.5 Unified Reasoning over Visual and Symbolic Knowledge 19
   3.6 Seamless Integration of Existing Databases & QA Systems 23

4 Conclusion 27
1 Introduction & Motivation

No one can dispute the fact that the standard questions used in tests of QA systems are painfully simple, and that this simplicity obviates the need to do sophisticated logical reasoning of the sort that is part and parcel of intelligence analysis (IA). The simplicity of the questions is confirmed by the fact that subjects who are plagued by bias (as shown by the established empirical techniques in psychology of reasoning for exposing bias) can nonetheless for the most part answer such questions correctly, and can provide correct justifications for these answers. Perhaps in response to the simplicity of these questions, few QA systems employ sophisticated logical reasoning, and still fewer can produce rigorous justifications defending the answers they give.

The questions in QA R&D undertaken for the Intelligence Community (IC) heretofore have also been simple because we are developing a radically new knowledge-based QA system, Solomon, that transcends the limitations of existing QA systems, and spans the entire IA process. Solomon is capable of producing rational, justified answers for conceptual, 1

1IA makes use of at least deductive, inductive, and abductive reasoning. This position is expressed, for example, by Moore & Krizan (2001).

2We in no way wish to impugn the value of the kinds of questions typically found today in current QA systems, and in systems tackling “text entailment” challenges like those posed by Pascal RTE. We don’t fully understand why these questions have received so much attention, while “harder” questions of the sort presented in (Bringsjord et al. 2006) are almost completely ignored, but we in no way wish to criticize the work revolving around standard, simple questions.

3Whenever possible, a justification should be an outright proof, though it will presumably be an informal one.

4If a way was discovered to correctly answer questions on a “high-stakes” test without being able to give rigorous justifications at the semantic level, the test would thereby be invalided; e.g., if someone were to produce an algorithm for correctly answering questions on the LSAT on the basis of superficial structure of the text, rather than on the basis of understanding its underlying propositional content, that would be the end of the LSAT. For this reason, it is quite peculiar that QA technology, and testing of that technology, has often steered clear of semantic-level justification.

5The limitations we have in mind are easy to see when one juxtaposes existing QA technology with a conversation one can have with a human child. For example, imagine that the children in a day care center are out for a walk led by their caregivers, and as a precaution are told to use the buddy system, so that each child holds the hand of another. Now imagine asking the children: “Is everyone holding the hand of someone?” In order to understand this question, and to answer it is in the affirmative (as each normal child would), one has to have a knowledge representation and reasoning power beyond extant QA technology, for the simple reason that the query here, formally, involves two nested quantifiers, and thus exceeds the mere atomic formulas that text extraction technology produces. You could also ask the children: “If Johnny lets go of Billy’s hand, will it be true that everyone is holding the hand of someone?” And so on, with the query containing more and more logical complexity. When we shift to the adult case, the gap between current QA systems and modern logicist systems grows much wider.
hypothetical, and even open-ended questions related to disparate knowledge-bases and information derived from reading documents. Herein, we report on the development of a prototype of Solomon that has validated the theoretical approach underlying the full system — which, in short, is to model QA on a more sophisticated form of human-machine interaction: one in which the machine has the power of cutting-edge machine reasoning technology.

The development of Solomon is motivated by the need for IA-focused QA systems to answer, with justification, “harder” questions (as explained in Bringsjord et al. 2006), and by four core needs of intelligence analysts, viz.,

**Avoid Bias:** Intelligence analysts, like most human reasoners, can, with disturbing ease, succumb to bias (e.g., see Heuer Jr. 1999). But because the stakes are high in intelligence, the consequences of succumbing to bias can be dire.

**Dodge Deception:** Intelligence analysts, more than most, must be able to determine when deception is afoot — or, to put it more crudely, when they are being lied to. Our enemies actively try to deceive and mislead the IC, and analysts must be able to detect, unmask, and counter these deceptions. But many of the same reasons why analysts succumb to bias (e.g., failure to systematically consider alternative explanations of the evidence at hand, incorrectly weighing the strength of evidence, etc.) also make counter-deception difficult and prone to error (e.g., see Godson & Wirtz 2002, Stech & Elsaesser 2005).

**Handle Novelty:** Intelligence analysts must be prepared to reason over novel information, in order to thwart novel plans to harm the United States, her allies, her citizens, and others under her protection. Unfortunately, our enemies don’t hand out a nice neat “training set” in advance of their strikes. On the contrary, as far as is possible given various constraints, our enemies explicitly try to behave in ways that cannot be computed from prior cases.

**Justify:** Intelligence analysts must be prepared to justify their recommendations and hypotheses; i.e., an analyst must be able to give, at least in principle, a cogent rationale that supports his/her recommendation. That rationale, if it is to be trusted and is to withstand external scrutiny, must be of a structure that is normatively correct, and certifiably so.

### 2 Six Distinguishing Attributes of the Solomon System

Solomon embodies a number of advances in QA technology. For brevity, we focus on only six of Solomon’s distinguishing attributes, to wit,
Attribute 1: Knowledge Acquisition via Reading

Solomon acquires knowledge through a process akin to how humans learn by reading, not by shallow text extraction technology. The knowledge Solomon acquires by reading far exceeds the knowledge acquired by current QA systems, which cannot extract arbitrarily complex knowledge from text.

Attribute 2: Human — Computer Collaboration via Conversation

Both Solomon and its users are active participants in the question answering process, with each asking and answering questions of the other. Their collaboration is in the form of a dynamic conversation in English wherein the answer to a question depends in part on the prior conversation (the questions previously asked and answered). Solomon’s conversational QA is not reducible to the decomposition of a single complex query.

Attribute 3: Natural Suppositional Reasoning

Solomon is not limited to simply answering questions of fact. Solomon supports conversation-based suppositional reasoning, i.e., “What if...” questions that introduce logical and linguistic contexts wherein further conversation is interpreted and evaluated.
Attribute 4: Defensible Answers and Rational Justifications

Solomon incorporates sophisticated automated reasoning and model finding. Answers and justifications relate to either counter-examples or defensible arguments (proofs/arguments by deductive, inductive, abductive, analogical, or visual means). These answers and justifications are explained in an intuitive fashion, in English, as part of the normal course of conversation.

Attribute 5: Unified Reasoning over Visual and Symbolic Knowledge

Solomon is able to answer questions requiring comprehension of visual as well as symbolic information. It is not that Solomon reduces the visual to the symbolic, for diagrams, pictures, satellite images, movies, maps, etc., all the things that are at the heart of human-level QA, are most certainly not symbolic entities. Solomon uses a new family of visual logics, known simply as VIVID (Arkoudas & Bringsjord 2005), to represent and reason directly over any computable image.

Attribute 6: Seamless Integration of Existing Databases & QA Systems
Solomon can be integrated on top of other databases and QA systems as a meta-QA system. Solomon extends attributes 2–5 across various disparate domains and specialized systems through the sound decomposition of proof-theoretic operations into direct model inspections that are then submitted to subordinate systems as database queries, or as yes/no questions of fact.

3 Prototype of the Solomon System

We have constructed a prototype of Solomon (thanks to the support of IARPA’s AQUAINT program), a first step toward realizing the system’s full potential. The explicit aim of the prototyping effort was to demonstrate attributes 1–5 above (and due to a happy confluence of circumstances, we are able to report significant progress on attribute 6 as well).

The prototype of Solomon builds on our Slate system, an intelligent IA assistant for the visual construction and mechanical certification of arguments. Slate is based on a robust, multi-faceted theory of heterogeneous human and machine reasoning — a theory that affirms the importance of deductive, inductive, abductive, analogical, and visual reasoning; arguments and counter-arguments; proofs and disproofs; models and counter-models; and strength factors in the tradition of Chisholm (1989) and (in the deductive case) Pollock (1995), which force explicit declarations of reliability in source and provenance information. This theory should in no way be confused (let alone be conflated) with limited, prior theories of argumentation and argument mapping, e.g., those based on Toulmin’s (2003) The Uses of Argument. From the standpoint of education and training, Slate is based on a neo-Piagetian view of the development of bias-free human reasoning, according to which, given sufficient training, neurobiologically normal humans can reason in a normatively correct, bias-free mode.

3.1 Knowledge Acquisition via Reading

Like humans, Solomon acquires new knowledge through actually reading text and diagrams. That is to say, the result of Solomon’s reading is new knowledge that

---

6The development of Slate, to this point, has been sponsored by ARDA, DARPA, DTO, IARPA, and by some internal support from RPI itself. While Slate has been engineered as a tool for intelligence analysts, it has also been designed to be of help to any “professional” reasoner, and any student of reasoning. For example, Slate has been part of the curriculum (as courseware) for teaching formal logic, argumentation, and critical analysis at Rensselaer.

7As an immediate corollary, note that Slate is radically different than all software systems based on such prior theories.

8For exposition on this point, see e.g., (Rinella et al. 2001, Bringsjord et al. 1998).
is poised-for the semantically correct generation of output. This semantically rich form of knowledge is fundamentally more expressive than the knowledge produced by the text extraction technology of current QA systems. This can be seen by examining the information in frames used by HITIQA, the Institute for Informatics, Logics and Security Studies’ excellent QA system. Figure 1 shows one of these frames. Notice that all the information in this frame can be expressed as a conjunction of what are called atoms (or atomic formulas) in first-order logic: propositions in which one or more particular objects have a simple property. An analyst can ask such a system various questions, and get back answers — but the range of productive questions is severely limited by the simplistic underlying knowledge. In fact, it seems difficult to ask a question that produces an answer that is non-obvious to a human having command over the knowledge in the frame. By contrast, Solomon will routinely answer questions regarding intelligence analysis case studies, e.g. those used in IARPA/DTO/ARDA programs like NIMD, ARIVA, and CASE, which no human can tackle without hours of reflection and analysis.

Unlike HITIQA, Solomon’s reading of texts directly produces fully quantified logical formulas; formulas expressing the concepts of “for all...” and “there exists...” For example, Solomon reads and understands “TSA searches every foreigner that boards a plane.” as meaning “TSA exists and for all foreigners y and planes z, if y boards z then TSA searches y.” — depicted graphically in Figure 2.

Currently, Solomon can translate text expressed in logically controlled English into multi-sorted logic (MSL), build knowledge expressed in MSL, and reason over that knowledge in proof-theoretic and model-based fashion. It can do this both on its own, and under human guidance. In light of this capability, we say that Solomon, in a fixed and confessedly limited sense, can “read.” A conceptualization of the process by which Solomon reads, shown in Figure 3, is described by three distinct phases:

Phase 1: English texts are rephrased in logically controlled English – i.e., a proper
subset of full English that can be unambiguously translated into a formal logic. At the present time Solomon makes use of Attempto Controlled English (ACE) (Fuchs et al. 1999, Hoefler 2004), a logically controlled English with a fixed, definite clause grammar and a user-defined vocabulary.\footnote{Phase 1 is currently a manual operation, but techniques developed by Mollá & Schwitter (2001) may allow for (at least partial) automation of this phase. Further, advances in wide-coverage semantic parsers (Bos 2005, Bos et al. 2004) may obviate Phase 1 altogether.}

**Phase 2:** Discourse representation structures (DRSs) are automatically generated from the controlled English. DRSs are a syntactic variant of first-order logic for the resolution of unbounded anaphora. Their use in the interpretation of text is a central element of discourse representation theory (Kamp & Reyle 1993, Kamp & Reyle 1996).

**Phase 3:** The DRSs are automatically translated into MSL, the chief native language of Solomon and Slate.\footnote{Slate has built-in translators for going from MSL to straight first-order logic (FOL), using long-established theorems (Manzano 1996).} As a DRS is equivalent to a quantified first-order formula, the translations to first-order logic and MSL are not conceptually difficult. Algorithms for performing such translations are provided by Blackburn & Bos (forthcoming;1999), among others.\footnote{We also point out that Blackburn & Bos — luminaries in the field of computational semantics and NLU — share our view that “state-of-the-art methods in first-order theorem proving and model generation are of direct relevance to inference for natural language processing.” (Blackburn et al. 2001, pp. 11)}

Below are screen-shots from three movies showcasing Solomon’s reading ability. In the first demonstration Solomon reads, and answers, a “harder” RTE-like...
item — one requiring semantic comprehension of quantified sentences, viz.,

T: Everyone likes everyone who is liked.
   Nobody likes himself/herself.

H: Nobody is liked.

Demonstration 1: Solomon comprehends and reasons with English sentences containing generalized quantifiers, as shown in the answering of a “harder” RTE-like item (www.cogsci.rpi.edu/research/rair/solomon/demo/demo1.wmv).

Correct comprehension of quantified sentences is critical for valid reasoning, but doing so is quite difficult for most humans. For example, when given\textsuperscript{12}

T: All the Frenchman in the room are wine-drinkers.
   Some of the wine-drinkers in the room are gourmets.

\textsuperscript{12}The “wine-drinkers” example is from (Johnson-Laird 1997).
humans, unless suitably trained in formal logic and reasoning, very often erroneously conclude:

H: Some of the Frenchmen in the room are gourmets.

The second demonstration shows Solomon immune to such cognitive illusions. Notice that in denying the conclusion, Solomon constructs and visualizes a counterexample showing that it is possible that none of the Frenchmen in the room are gourmets.

Demonstration 2: Solomon answers an illusory RTE-like item; Solomon’s users are immunized against cognitive illusions to which unaided humans regularly fall prey (www.cogsci.rpi.edu/research/rair/solomon/demo/demo2.wmv).
The third demonstration is of Solomon assisting a human in tackling a GRE-like analytical reasoning problem (shown in Figure 4). As shown in the movie, the human adopts a falsification strategy (a.k.a. process of elimination), which is substantially facilitated by Solomon answering questions about the consistency and consequential status of candidate answers.

![Three security force details — ‘Papa’, ‘Quebec’, and ‘Roman’ — are assigned to guard three sections — the high-voltage junction substation, the Kisyoloe turbine generator, and the spent-fuel pool. Each shift begins at 9:00 a.m., 10:00 a.m., and 11:00 a.m. Exactly one of the three details is assigned to each of the sections per shift. No detail is assigned to any given section for more than one shift, and no detail is assigned to more than one section per shift. An Abu Sayaf strike team, intent on stealing spent fuel rods, has tracked the movements of the security forces for the past week. From their observations the strike team has learned:]

- ‘Papa’ always guards the Loading dock beginning at 11:00 p.m.
- ‘Papa’ always guards the Kisyoloe turbine generator, which is the ‘wrong’ answer.
- ‘Quebec’ guards the Pool, which is the ‘right’ answer.

Which of the following does the Abu Sayaf strike team believe can be the assignment of guards at the Loading dock beginning at 9:00 p.m., 10:00 p.m., and 11:00 p.m., respectively?

- (A) ‘Papa’ ‘Roman’ ‘Papa’
- (B) ‘Quebec’ ‘Papa’ ‘Roman’
- (C) ‘Roman’ ‘Papa’ ‘Roman’
- (D) ‘Roman’ ‘Quebec’ ‘Papa’
- (E) ‘Roman’ ‘Roman’ ‘Papa’

Figure 4: A GRE-like analytical reasoning problem.


---

In GRE analytical reasoning problems, the ‘wrong’ answers are necessarily false (i.e., they are logically inconsistent with the information given in the question’s preface) while the ‘right’ answer is possibly, but not necessarily, true (i.e., it is consistent with, but not necessarily a consequence of, the information given in the question’s preface).
3.2 Human — Computer Collaboration via Conversation

Automated theorem proving demonstrates how difficult the task of question decomposition is. Advanced automated theorem provers (ATPs) such as Vampire and SNARK are often able to quickly construct complex proofs of hundreds or even thousands of inferences, yet formal mathematics is rife with novel proofs whose core strategies seem undiscoverable through automated techniques. Probably the two best known examples of such are Cantor’s diagonalization proofs and Gödel’s model-based proofs. If a proof checker is given a proof using one of these forms, it is able to validate all of the inferences required to justify the conclusion, but ATPs are unable discover the proof in the first place.

Solomon overcomes the problem of question decomposition by harnessing both the ingenuity of the human and the raw power of the machine. Solomon and its users collaborate together in the problem answering process. This co-equal interaction maximizes the strengths of each, the guiding of strategy for the user and the execution of tactics for Solomon, so that the resulting process is more efficient and effective than either participant on their own. Collaboration is structured as an open-ended conversation between Solomon and users with each asking and answering questions of the other. The dynamic and non-deterministic nature of this type of conversational QA is not reducible to the query decomposition model prevalent in current state-of-the-art QA systems.

The expressivity of Solomon’s conversational ability is modeled on System S, the graphical argumentation system that is part of our Slate system. In Slate, arguments supporting hypotheses are constructed and diagrammed in a workspace. Various actions can then be performed on the hypotheses and arguments, such as verification, defeating, and alternative hypothesis generation. In Slate’s graphical argumentation system there are a number of visual workspace elements, e.g.: propositions, sets (including ontologies and knowledge-bases), hypotheticals, models & countermodels, arguments & counter-arguments, sub-arguments, inference & evidence links, and databases. An analyst can construct various arguments, make assumptions, assess evidentiary & inferential strength, and formulate hypotheses by manipulating instances of these visual elements in the workspace, and System S provides a formal interpretation of the argument diagram that the analyst constructs.

Solomon’s chat-like conversational interface allows an analyst to linguistically construct and manipulate the elements of System S, i.e., Solomon contains a lin-

\[14\text{Put in terms of some of the reasoning technology in our lab, e.g., Athena, the problem is that ATPs, for challenging problems, can operate only as type-\(\alpha\) systems, not type-\(\omega\) systems. The former only certify pre-existing proofs; the latter create certified proofs (for further exposition on type-\(\alpha\) and type-\(\omega\) systems, see Arkoudas 2000).}\]
guistic counterpart to System S — illocutionary equivalents for creating, exploring, and evaluating arguments in the workspace. To illustrate, Demonstration 4 revisits the “harder” RTE-like item from Demonstration 1 but this time the answer is elicited via Solomon’s chat-like conversational interface.

Demonstration 4: The “harder” RTE-like item (introduced in Demonstration 1) is re-answered, this time using Solomon’s chat-like conversational interface (www.cogsci.rpi.edu/research/rair/solomon/demo/demo4.wmv).

Demonstration 5 also showcases Solomon’s chat-like interface. In this demonstration, Solomon assists a user in solving the “New Order” IA micro-scenario. (In the “New Order” scenario, one must determine the perpetrator of an in-terror-cell killing. The facts relevant of the micro-scenario are shown in Figure 5.)

Assume the following statements are true:
1. John H. was killed by a member of the Al-Qaeda cell ‘The New Order’.
2. The only members of ‘The New Order’ were John H., Majed H., and Essid D.
3. Within-cell killings only occur when the attacker believes the victim is a traitor, and never when the attacker is of lower rank.
4. Essid D. believes that nobody is a traitor who John H. believes is a traitor.
5. John H. believes everyone except Majed H. is a traitor.
6. Majed H. believes that everyone who is not of lower rank than John H. is a traitor.
7. Majed H. believes everyone is a traitor who John H. believes is a traitor.
8. No one believes everyone in ‘The New Order’ is a traitor.

Who killed John H.?

Figure 5: The relevant facts of the “New Order” micro-scenario.

We have embodied Solomon — or, more correctly, some of the technologies behind Solomon — as an autonomous conversational avatar in Second Life (shown in Demonstration 6). In the demonstration, the avatar “BinDistrib Burdeyni” is powered by the same linguistic and reasoning technologies that are behind Solomon, coupled with a small knowledge-base encoding the “memories” of the synthetic character.

3.3 Natural Suppositional Reasoning

The most challenging questions for QA systems, also the most relevant to analysts, are those that simply cannot be succinctly posed as an individual question. Imagine an analyst who wishes to explore a hypothesis through a series of “What if...” questions. It is almost certain that at each step in the exploration, the “What if...” question he/she asks next depends largely on the answers to some, if not all, previous questions. Intuitively, these suppositions introduce nested scopes (contexts) within the dialog in which questions and answers are interpreted. These contexts can themselves be reasoned over, say for an argument by cases. Very few QA systems allow suppositional or hypothetical questions; none, to our knowledge, allow rich open-ended exploration that is on par with Solomon.

In Solomon, phrases such as “suppose...” and “assume...”, introduce suppositional and hypothetical contexts, thus allowing the articulation of generalized conditional answers and arguments. Demonstration 7 is a simple example of natural suppositional reasoning with Solomon. In this demonstration, Solomon is given only part of the necessary information for the “harder” RTE-like item (see demonstrations 1 & 4), the remaining necessary information is suppositionally introduced by the user.

Demonstration 7: The “harder” RTE-like item (see demonstrations 1 & 4) is re-answered again, this time using linguistically-based suppositional reasoning (www.cogsci.rpi.edu/research/rair/solomon/demo/demo7.wmv).
3.4 Defensible Answers and Rational Justifications

If QA systems are to be relied on to answer questions of increasing importance and criticality, then their justifications must be complete, transparent, and available for scrutiny. Solomon provides rational explanations for the answers it produces. Internally, all of Solomon’s answers are grounded in formal reasoning. Solomon can, at any moment, provide the user with a clear and intuitive explanation of a clear argument for the answers it has generated. Only QA systems based on formal reasoning mechanisms can provide such complete justifications.

Solomon generates English explanations from natural style proofs (Solomon’s formal justifications for answers given). These natural style proofs are inherently structured according to human intuitions in reasoning, which results in discourses that are clear, concise, and intuitive. (While our NLG technology is still nascent, in our experience, it is quite useful for the creation of first-draft reports.) Demonstration 8 shows the generation of an explanation for RTE item 570 from a Slate argument affirming that item’s hypothesis is entailed.

Demonstration 8: An English explanation is generated from a workspace argument affirming the positive consequential status of PASCAL RTE item 570 (www.cogsci.rpi.edu/research/rair/solomon/demo/demo8.wmv).

Demonstration 9 is a more involved example of justification generation (again, within Slate); in it, a significant portion of Hughes’s (2003) Case Study #4 (The Sign of the Crescent) — specifically, the evidence and arguments contained in Chart H — is diagrammed in the Slate workspace, and explanations are generated for the various arguments contained therein.
Demonstration 9: Chart H from Hughes’s (2003) Case Study #4 (The Sign of the Crescent) is diagrammed, and English explanations are generated therefrom (www.cogsci.rpi.edu/research/rair/solomon/demo/demo9.wmv).

Lastly, Demonstration 10 shows natural-language explanation generation integrated into Solomon’s conversational interface. In this demonstration, the “New Order” micro-scenario (see Demonstration 5) is re-answered, and then Solomon is asked to generate an English explanation for its answer.

Demonstration 10: Solomon re-answers the “New Order” micro-scenario (from Demonstration 5) and articulates an explanation for the micro-scenario’s solution (www.cogsci.rpi.edu/research/rair/solomon/demo/demo10.wmv).
3.5 Unified Reasoning over Visual and Symbolic Knowledge

Research in QA is typically limited to asking questions about textual knowledge. However, analysts often seek answers to questions pertaining to both visual and textual sources. For example, much effort has gone into adducing Osama bin Laden’s health and possible location from Al-Qaeda propaganda videos.

Now we come to a very important issue, which must be explicitly discussed: Some may say that ARIVA does not need a visual logic, and can thrive with just a traditional symbolic logic for interoperability. Unfortunately, ARIVA systems, as described in the BAA, cannot interoperate on the basis of a symbolic logic. There are at least three reasons why this is so; we cover this trio now.

First, the meaning of a visualization relative to a human user is lost when any attempt is made to reduce it to symbolic logic. The reason is that visual data is often a so-called homomorphic representation; i.e., a representation that refers in virtue, at least in part, of its size, shape, texture, color, etc. In standard logic, I can use ‘car-22’ to refer to your car, but I could also use ‘car22.’ There is no problem here. But if I have a diagram of your car, and I remove a piece of this diagram, that may cause the representation to fail to refer. For example, if I change the color of the representation in the case of the visualization, your car may no longer be denoted. You may own a black Jeep SRT, while Jones may own a red one, and if my diagrammatic representation of your Jeep shows the color as red, reference fails. However, changing the font color of ‘car22’ to ‘car22’ in a logic is guaranteed to have no impact at all on the meaning of this name. (The color may reflect meta-data, but that’s different: the denotation of a constant isn’t affected by meta-data associated with it.) In sum, the meaning of a visualization or diagram is not preserved when one moves to a corresponding purely symbolic representation of it. This is a brute fact (nicely shown at length, e.g., in Barwise & Etchemendy 1995).

The second reason why ARIVA needs visual logic pertains to efficiency, and here we come to mathematical issues. It is easy to prove that reasoning over visualizations is much more efficient, in many cases, than reasoning over symbolic re-expressions of these visualizations. We give now a simple, but nonetheless conclusive, example.

In this simple example, the job of the analyst (or you, as reader) is to see what you can infer about the locations of suspected terrorists a, b, c, and d in the following idealized problem. Suspected terrorists are represented by squares, and cities by triangles. The given information consists of what is shown visually, and also some symbolic information, as indicated (in the four propositions). The symbolic information is given in English; below each sentence is a naive encoding of this English in first-order logic, given the idealization that the grid represents a part of the relevant foreign country. (E.g., a suspect (square) is in a city (triangle) if it adjoins it, and
is not in front of it.) The task is to figure out where on the grid the four suspected terrorists (red squares) must be placed, and to figure out whether or not it can be established that b is dotted. (Dots represent some arbitrary attribute of importance, such as whether the suspect underwent religious training of a certain sort.) So, can a human solve this problem visually? Yes indeed. In fact, an analyst who understands the sentences can soon see that the suspects can each be located in a particular place in the grid. Do you see what those locations are?

![Grid Diagram](image-url)

**Figure 6: TBD.**

- All suspected terrorists are in one of these cities.
  \[ \forall x (\text{Suspected}_T(x) \rightarrow \exists y (\text{City}(y) \land \text{Small}(y) \land \text{Adjoins}(x, y) \land \text{FrontOf}(y, x))) \]
  - a is located in the same corridor city2 is in.
    \[ \text{SameColumn}(a, \text{city2}) \]
  - Suspected, “dotted” terrorists are not located south of city1.
    \[ \forall x ((\text{Suspected}_T(x) \land \text{Dotted}(x)) \rightarrow \neg \text{FrontOf}(x, \text{city1})) \]
  - c is located west of a.
    \[ \text{LeftOf}(c, a) \]

Sufficiently motivated readers will soon enough see that the suspects (squares) can be placed with certainty in particular cites (i.e., in back of and adjoining certain triangles). (It’s generally best to start by realizing that a must be directly north of/back of city2: We know that a must be in the same column as city2, that a must adjoin and be in back of a small city in this column, and that a cannot be located south of/front of city1. The only alternative is that a is located just north of city2.) The picture that shows the right answer is the following one:

This step, that is, the step of inferring this new picture from the first one, can be
done in one step by humans, and, likewise, in one step in Vivid-CL. Please let that sink in. This inference takes literally many pages of laborious inference when cast in some symbolic logic like IKL. The main reason is that all the visual primitives (e.g., FrontOf) need to be axiomatized: that is, represented as a list of standard formulas. And reasoning would need to reference this list. In the case of just FrontOf, we would need an axiom stating that FrontOf(x,x) cannot hold, for all x; that for all x and for all y, if FrontOf(x,y) then not-FrontOf(y,x); that for all x, y, and z, if FrontOf(x,y) and FrontOf(y,z), then FrontOf(x,z); and on and on it goes, ad nauseum. Keep in mind that similar axioms are needed for the other predicates. In short, humans who use visualization to reason outperform humans who don’t; machines that use visual logics to reason outperform machines that don’t. Even in this simple example, when it comes to the length of a proof, Vivid-CL will trounce any symbolic logic, and thus trounce any automated prover using a symbolic logic.

The upshot is that it would be unacceptable, for efficiency reasons alone, to try to reduce visualizations to symbolic representations in IKL, and translate from there, and then "expand" up from the translation to some visualization in another system. Such an approach would be sound only if it were sound to expect mathematicians and logicians to present and publish all their proofs in purely symbolic form, as chains of formulas in first-order logic. But it would take a lifetime to try to carry out such a reduction for even one substantive proof. That’s why nearly all robust proofs created by humans make use of visual representations.

Solomon has begun to address heterogeneous (visual & textual) QA for the limited domain of geometric math questions, viz., those created by the Trends in International Mathematics and Science Study (TIMSS) for 8th grade students (IEA 2003). To “read” these questions, Solomon first employs computer vision techniques to gather declarative knowledge about the visual data. (Note, this pro-
cessing extracts only cognitively plausible visual information so, e.g., Solomon might extract from an image that there are lines $AB$ and $XY$, and that the two intersect at a point $Z$, but Solomon does not, e.g., extract from the image that the angle of intersection between lines $AB$ and $XY$ is 17.07938°. Next, the declarative knowledge extracted from the visual data is combined with the knowledge gained by "reading" the question's text, and with a small knowledge-base of geometric facts.

Demonstration 11 shows Solomon answering a modified "lines and angles" question from the TIMSS problem set. Notice that we have modified the diagram contained in the problem's preface so that the lines $HK$ and $LN$ are not visually parallel. Solomon is able to correctly answer the question because it can (and must) integrate the visual information of lines $HK$ and $LN$, with the textual statement: "$HK$ is parallel to $LN$.”

Demonstration 11: Solomon solves a “lines and angles” problem from the TIMSS problem set by reading, integrating, and reasoning over heterogeneous information (www.cogsci.rpi.edu/research/rair/solomon/demo/demo11.wmv).

---

15 Solomon extracts from images the existence of salient visual elements such as points, lines, and intersections. It also extracts basic relative positioning, e.g., above, below, left-of, etc. Complex relations (e.g., parallel-to, perpendicular-to, acute, obtuse, etc.) and exact measures (e.g., length, angle) are not extracted. Such properties are instead inferred by Solomon based on other knowledge.

16 The knowledge-base in this case, is of the kind which Solomon can automatically build by “reading” elementary geometry textbooks, such as (Collins et al. 1999).
3.6 Seamless Integration of Existing Databases & QA Systems

Thanks, in large part, to the support of the IC through the AQUAINT program, QA technology has achieved a degree of maturity. But for QA systems to be effectual when deployed in the field, they must interoperate with each other and with existing data-stores. While various standards and languages (e.g., XML) can (and have) be used to achieve syntactic interoperability, until now, there as been no general method of achieving semantic interoperability.

The need for semantic interoperability (as opposed to mere syntactic interoperability) is exemplified in the constant and continuing difficulty of database integration (e.g., “the schema mismatch problem”). Even though databases, by and large, use standardized schema and transaction definition languages, achieving semantic interoperability amongst ndatabases has generally required “n choose 2” number of translation functions between systems (shown pictorially in Figure 8).

![Figure 8: Semantic interoperability between n systems often requires \( \frac{n!}{2!(n-2)!} \) translation functions.](Link to Andy’s movie)

Solomon is able to share knowledge with (and reason over knowledge managed by) many other information systems. This communication is achieved through the use of provability-based semantic interoperability (simply called PBSI, see ?, ?), a framework designed to facilitate information exchange between fundamentally different knowledge-management systems provided only that the information stored in each system is somehow meaningfully (i.e., semantically) related. PBSI differs from many other translation frameworks in that it is designed to exchange not only information that is structured in different ways (e.g., a XML file vs a SQL database) but knowledge that differs in meaning (or semantics). That is, using PBSI, multiple systems can share knowledge even when the data that each individual system deals

\[ ^{17} \text{For further information on PBSI and “the schema mismatch problem,” see our presentation} \]
with is drawn from separate ontologies.

PBSI provides a language for formalizing the relationships between ontologies via bridging axioms, and our extension, PBSI+, associates each information exchange with a proof certifying the conservation of semantic meaning. The basic construct of PBSI+ is the signature, a collection of (meta-theoretic) definitions which, coupled with a set of axioms, captures a given ontology. A translation graph provides the framework for bridging signatures (and so, ontologies) while preserving semantics. A translation graph, like the one in Figure 9, is a directed graph where the vertices are each unique signatures, and each edge describes the application of a primitive operation relating two signatures. The framework and corresponding implementation of PBSI significantly reduces the number of translation functions required for semantic interoperability, and partially automates their construction. The semantic interoperability of multiple systems is achieved by interconnecting each system (by way of a translation graph) to an intertheory. This approach requires only $2n$ translation functions for $n$ systems (shown in Figure 10).

![Translation Graph](image)

**Figure 9:** A sample translation graph enabling semantic interoperability between four related ontologies.

![Intertheory Graph](image)

**Figure 10:** With PBSI, semantic interoperability between $n$ systems only requires $2n$ translation functions.

To exemplify the relevance of PBSI to QA, imagine that $\Sigma_1$ is an INS database of US persons and companies sponsoring foreign nationals, $\Sigma_2$ is an IRS database linking, among other things, persons to employers, and $\Sigma_3$ is an ATF database of

\[18\] At present, PBSI is semi-automated. Work is underway on a more complete automation of the approach.
registered owners of firearms; using PBSI and the supposition that foreign nationals have access to his/her sponsor’s firearms (or possibly to the firearms of their sponsor’s other employees), Solomon is able to answer an analyst’s question as to whether some particular, suspicious foreign national has access to firearms (i.e., a question requiring reasoning over $\Sigma_1$, $\Sigma_2$, and $\Sigma_3$).

We have exercised PBSI in a series of demonstrations of interoperability between Slate and SQL databases, Oculus’ GeoTime, and PNNL’s AKEA (two of which are shown below in Demonstration 12 and Demonstration 13).

![Demonstration 12](image-url)

Demonstration 12: PBSI enables semantic interoperability between Slate, Oculus’ GeoTime, and SQL databases; together, they tackle Chart E from Hughes’s (2003) *Case Study #4 (The Sign of the Crescent)* ([www.cogsci.rpi.edu/research/rair/solomon/demo/demo12.wmv](http://www.cogsci.rpi.edu/research/rair/solomon/demo/demo12.wmv)).
With this additional information, the analyst entertains the hypothesis that Palar is the coordinator of the attack, and seeks information to support or refute it.

Demonstration 13: Andy, can you provide a caption (www.cogsci.rpi.edu/research/rair/solomon/demo/demo13.wmv).
4 Conclusion

To do.

References


IEA (2003), ‘TIMSS 2003 Mathematics Items Released Set Eight Grade’.


