Research Statement

Paulo Pinheiro da Silva

My primary research interests focus on trusting software systems and on trusting answers from software systems in distributed environments such as the web. Specifically, I am interested in explaining answers from a broad number of software systems including database systems, information integration and information extraction systems, information retrieval systems (e.g., search engines), and theorem provers. To explain answers, I am also interested in how these systems represent and reason with knowledge including their potential use of ontologies. My long term research goal is to develop a universal distributed infrastructure over the web enabling software systems to explain their answers. Explanations may include descriptions of how answers are produced, provenance information about sources, engines and inference rules used during the production of answers, and the computation of trust values for answers. Demonstrating the usefulness of explanations for users to understand answers is also part of the research goal.

Current and Past Research

Explaining Answers from Software Systems

During my Postdoc, I have being a technical co-leader along with Deborah McGuinness of a research team developing the foundations of Inference Web (IW), a distributed infrastructure for web explanations. IW enables a number of types of software systems to generate portable and distributed justifications for their answers. IW addresses the need for trust that arises when software systems perform diversified kinds of information manipulations, i.e, deduction, extraction and retrieval tasks, in heterogeneous environments such as the web. IW aims to support users (humans and computer agents) as they decide whether to trust answers before they use those answers with confidence. IW addresses knowledge provenance issues on the web in an equivalent way that these issues are discussed in the database community. IW provides tools and metadata for checking and browsing proofs (justifications). IW addresses the issues of explanations (proofs transformed by rewrite rules for understandability) with its language axioms and rewrite rules. The Proof Markup Language (PML) is a major IW component addressing the needs for combining and sharing proofs. PML is a key component of the DARPA PAL and ARDA NIMD projects. PML is a key component of IBM’s UIMA effort intended to be a standard framework for text analytics.
Specifying and Verifying Interactive Systems

During my Ph.D., I studied how models of user interface functionality are used to represent different features of an interface, and indicated how such functionality can be supported using both standard UML and the Unified Modeling Language for Interactive Applications (UMLi), a conservative extension of UML with explicit support for interface modeling. Indeed, the user interface is an essential part of most software systems, and often represents a significant proportion of the code developed during implementation. However, despite this there has been little specific attention given to user interface issues in many object modeling languages, including UML. Later in my Ph.D. work, I developed a semantics for UML by mapping UML constructs into the LOTOS specification language. Thus, by using available tools for LOTOS verification, I could perform typical model checking over generated specifications. Semantic problems related to complex relationships between class and activity diagrams can be identified in addition to the identification of deadlocks, livelocks and unreachable states in behavioral diagrams.

Future Directions

As web usage grows, a broader and more distributed array of information services becomes available for use and the needs for explanations that are portable, sharable, and reusable grows. Despite the actual impact of the Inference Web, we are just starting to address some relevant limitations of the current infrastructure:

- **Support for automated trust value computation of answers from the web.** There are many approaches in the literature for computing trust values between resources such as information sources and users. However, there are few approaches for computing trust values for answers. In fact, such computations require access to data structures providing full support for tracking provenance. From trust values for information sources, engines can vet the use of information sources during answer derivation. Even more interesting is the ability to order question answers by trust values.

- **Flexible capabilities for matching rule specifications against proof steps.** Matching is a key operation for both checking proofs and abstracting proofs into explanations, e.g., by applying rewriting rules. IW has limited support for matching and one of the Inference Web goals is to increase matching capabilities as much as possible. Research issues remain in the face of different syntaxes for sentence presentations and varying orders of proof step antecedents.

- **Support for a flexible and diverse proof interlingua.** The existing Proof Markup Language addresses many styles of information manipulations, however it requires extensions in order to meet the needs of a true semantic web proof markup interlingua. Examples of new uses of PML that may require extensions to the language are proofs based on inductive and probabilistic reasoning.

In terms of trusting software systems, I am interested in the development of a semantics for models of interactive systems aimed at identifying specification problems. In fact, verified specifications
are likely to have fewer problems than unverified specifications. Also, implementations of software systems derived from verified specifications may have fewer problems.

- *Integrated approach to verify and reason with behavioral and structural properties of software specifications.* There are many proposals for the semantics of UML constructs. However, few proposals, if any, describe the meaning of a subset of UML constructs that is comprehensive enough to support the construction of structural diagrams, e.g., class diagrams and dynamic diagrams, e.g., activity diagrams. I propose to develop a temporal logic based semantics for a subset of constructs of selected structural and dynamic UML models. The proposed semantics would enable a combination of reasoners and model checkers to verify UML models in an integrated manner preventing developers from introducing new problems in one diagram by correcting problems identified in another diagram.

In terms of funding, I plan to pursue a multi-pronged approach. I expect to obtain support from Stanford, funded by ARDA’s NIMD and DARPA’s PAL projects, to continue some of the Inference Web work in collaboration with Stanford. I also expect to apply for new faculty grants as well as collaborating on new proposals with colleagues from my new department and current collaborators from organizations such as Stanford University, Manchester University, SRI, and IBM.

While I am dedicated to research, I would also like to have my research tested in commercial solutions. To that end, when research ideas appear commercially promising, I would like to facilitate technology transfer of the ideas to startup companies. In particular, the use of trust values for ranking answers and UML model checking hold promise for such experiments.

**Summary**

It is desirable that software systems in general are able to explain their answers. Thus, there is a broad scope of possible usage of Inference Web. Software systems can vary from centralized database systems to distributed hybrid reasoners over the grid. Application domains can also vary from everyday contexts, i.e., web applications selling airplane tickets, to complex applications, i.e., genomic database systems and national security systems. I am very motivated to contribute to research efforts in such application areas when I can perceive explanations as a topic of interest for application users.

There is growing demand for services such as those provided by Inference Web. This is evidenced by the increased usage of Inference Web in the U.S., Canada, Europe, and South America along with the increasing dependency of major U.S. Government projects on our explanation strategies. Information extraction engines, information integration systems and theorem provers are examples of software systems using the IW for explaining their answers. IW achievements, however, are still modest and we do expect many other kinds of systems to benefit from our research work. I do expect users and researchers to identify limitations and areas for growth which is a good reason motivating me to keep doing research.