

Candidate's Statement

Premkumar Devanbu

<http://www.cs.ucdavis.edu/~devanbu>

My core area is software engineering. I also have a secondary interest in information security. In software engineering, I have two areas of focus: middleware, and safety of meta-programs. In the information security area, the two concerns are authenticity and trustworthiness of information.

1 Middleware

Middleware is a layer of abstraction over the operating system, which simplifies application software development. Distribution middleware, my area of interest, makes it easier to build systems whose components must be distributed. Distributed systems are common in commerce, healthcare, transportation, *etc.* Distribution middleware handles details such as communication and addressing, thus simplifying distributed systems development. This is achieved through a high-level interface definition language (IDL) for modeling components, generation of glue code, and a powerful run-time environment. My research (with Eric Wohlstadter) began with the observation that the implementations of requirements such as security, fault-tolerance, and billing (collectively known in the literature as *non-functional requirements*) present special difficulties. They affect many different components, in ways that cannot be modeled in current IDLs; they require specialized skills (not always present in application developers); they must be re-implemented from scratch in each application, because they affect different applications in different ways; and finally, must often be determined at run-time, for each given pair of interacting components. We have made two contributions in this area. The first, published in *ICSE 2003* conference [6, 4, 5] introduced a new type of middleware, that greatly simplifies the construction of software components for non-functional requirements such as security, through modeling and code generation; in addition DADO provides a separate stage of modeling and code-generation to actually bind these non-functional components to the application software. This allows, for example, third-party development of security components, which can then be later integrated into different applications. A key aspect of DADO is the possibility of binding non-functional requirements *late*. The first DADO paper [6] was nominated for an ACM Distinguished paper award. We have also been invited by OMG (the cognizant body for the popular CORBA standard for distribution middleware) to present our work at their next meeting. As another indication of the recognition of our work, I was invited to serve on the program committee of *ACM Middleware 2004*, the leading specialized conference in the area. In follow-on work, we developed GlueQoS [7], which is a new type of middleware (again comprising modeling, code generation and runtime) that allows the negotiation and selection of non-functional features, between a pair of interacting components, *at runtime*. This work is funded by an NSF grant, and an IBM Faculty Partnership award.

2 Metaprogramming correctness

The standard notion of correctness applies to programs which compute over *values*. Static type-checking, for instance, can ensure that a program will never store a number into a string variable. Metaprograms are programs which may compute over *programs*. For example, a program that accesses databases may construct programs in a query language such as SQL, and execute them. The popular JDBC standard interface allows Java programs to construct SQL queries and execute them. As another example of metaprogramming, a program *p1* may dynamically access another program, *p2*, study its structure, and then dynamically construct “glue” code that allows *p1* to run *p2*. This occurs, for example, in reflective programming, as supported by Java and middleware such as CORBA. In both these settings, the notion of correctness is quite different from the usual notion of type-safety...one would like to guarantee that all programs constructed by a metaprogram are correct. Currently, the only way would be extensively test the metaprogram under various conditions to see that the constructed programs do not go wrong. However, testing is costly, time-consuming, and typically cannot rule out the possibility of residual errors. Along with Zhendong Su, we have been exploring the possibility of statically checking the metaprograms, (without the need of testing) in several settings. Our first foray was a JDBC checker, which statically checks Java/JDBC programs, and *guarantees* that the queries produced by these programs will be free of syntax and type errors. Papers [8, 9] on this work were published at ICSE 2004, and won a distinguished paper award. Currently, along with Su, and Derrick Pallas, we are extending this approach to statically check reflective programs written in Java.

3 Authenticity & Trust in Information

With the wide variety of information sources now available, It’s difficult for users to know whether information is trustworthy. Given the scale of the internet, managing trust relationships between information users and providers is a challenge. This work has taken two directions. The Truthsayer project, with Martel, Gertz and Stubblebine, has developed a notion of authentic publication, where untrusted providers can efficiently provide unforgeable “proof” that their answers to queries are the same as what would be provided by the (trusted) information authors [3]. Most recently, we have developed a generalized technique that can be used to add authentic publication algorithms at a modest overhead to existing information systems. A paper has been accepted to Algorithmica [1]. We also developed a flexible signature model that allows selected portions of XML documents to be authenticated [2]. This project, supported by a medium ITR grant, has gained good visibility. A search of citeseer reveals over 40 citations collectively to the various papers we have published in recent past.

The other direction of this work is on managing trust relationships with providers, using ratings of these providers provided *a priori* by reliable rating services. In this context, we consider the problem of responding to a demand from a user to provide a response to a query at a prescribed trust level. Is it even possible to provide such a response? It would be desirable to provide a static answer to this question, before evaluating the user’s query. We have developed the notion of static trust typing [10, 11] for doing just that, along with

desirable properties (correctness, precision, completeness) of trust typing algorithms; we have also developed a trust typing framework for the relational algebra, along with proofs that it satisfies the above mentioned properties.

4 Impact & Recognition

Most notably, my paper on GENOA, in ICSE '92 [12] was nominated for the 10-year most influential paper award at ICSE 2002. A letter I received from Prof. Michal Young, the program chair of ICSE 2002 states:

Your ICSE 92 paper on GENOA was a strong contender and first runner-up for the award. Although the award will go to another paper, we thought you would like to know that your 1992 paper is held in high esteem by leaders in the research community, and is recognized by them as having had significant impact on software engineering research.

This award is given retrospectively to the paper (from the conference held 10 years prior) judged to be the most influential by the 2002 conference committee; it's nice to even just have come so close. Previous winners of this award include Dave Parnas, Mark Weiser, David Harel, Leon Osterweil, Walter Tichy, Dewayne Perry, and Bob Balzer. In ICSE 2003, (the first year the distinguished paper awards began) the paper on DADO [6] was one of six nominees (out of 437 submitted papers) for a distinguished paper award, although it did not get the award. In ICSE 2004, the paper checking JDBC programs [9] was one of 5 papers (out of 450 submissions) that won the distinguished paper award. I have published 14 full-length papers, during my career, at ICSE: by comparison, a search of the DBLP on-line database reveals that just one colleague (Prof. Vic Basili) has published more full-length papers at ICSE. Considering all types of ICSE papers, I drop down to fifth, behind four distinguished colleagues: Basili, Barry Boehm, Leon Osterweil, and Dewayne Perry. Considering all papers published in ICSE and FSE (the two top conferences in software engineering), I drop to sixth, behind Basili, Boehm, Lori Clarke, Jeff Kramer and Perry.

It's also a pleasure to report that my first three Ph.D students will be out this year, and all have obtained faculty positions! Eric Wohlstadter, had an outstanding record (including 3 ICSE papers) and interviewed at highly-ranked places, including UBC (Vancouver), UC Irvine, University of Toronto, and UT Austin. He had 2 early offers, and accepted one from UBC. Stoney Jackson had four offers from teaching schools, and signed on at Western New England College. Brian Toone applied only to Samford University, in Birmingham, Alabama, was interviewed, and has accepted their offer.

5 Future Directions

Systems in the future will need to adapt dynamically to cyber-security conditions, network outages, and natural/physical adversities. How can one design systems that will respond well to such adversities? Programming adaptations into large systems is difficult; changes may be required to many different components. In fact, changes to some components not under the application programmer's cognizance (e.g., to the operating system, or the network) may be required for efficiency or security reasons. I am developing an approach based on self-aware

and context-aware systems, whereby a system with a knowledge of its architecture, and of its interacting peers, can simplify the programmer's burden by automatically transforming and optimizing adaptations. I have just won a highly competitive IBM Faculty Partnership Award to support this research; I have also written an NSF proposal.

Bibliography follows. Some papers have appeared in highly competitive, peer-reviewed conferences with archival proceedings. Their bibliographic listings are followed an acceptance rate as in (15%).

References

- [1] Martel, C., Nuckolls G., Devanbu, P., Gertz, M, " A General Model for Authentic Data Publication". *Algorithmica*, accepted, to appear.
- [2] Devanbu, P., Gertz, M., Kwong, A., Martel, C., Nuckols, G., and Stubblebine, S., "Flexible Authentication of XML Documents", *Journal of Computer Security*, accepted, to appear.
- [3] Devanbu, P., Gertz, M., Martel, C., Stubblebine, S. , "Authentic Third-Party Data Publication", of *Journal of Computer Security*, 11(3), 2003.
- [4] Wohlstadter, E., Jackson, S., Devanbu, P.T: Design and Implementation of Distributed Cross-cutting Applications, Research Demonstration Track, *ICSE 2004*.
- [5] Wohlstadter, E., Devanbu, P., "DADO: a novel programming model for distributed, heterogeneous, late-bound QoS implementations", Proceedings, *Workshop on Secure Reliable Middleware (SRM)* , Catania, Italy, 2003.
- [6] Wohlstadter, E., Jackson, S., Devanbu, P, "DADO: Enhancing middleware to support cross-cutting features in distributed, heterogeneous systems", *ICSE 2003 (13%) Nominated for the ACM SIGSOFT Distinguished paper*.
- [7] Wohladter, E., Tai, S., Thomas, A., Rouvellou, I., Devanbu P., "GlueQoS: Middleware to Sweeten Quality-of-Service Policy Interactions", *ICSE 2004*, Edinburgh, UK. (13%)
- [8] Gould, C.R., Su, Z., Devanbu, P.T., JDBC Checker: A Static Analysis Tool for SQL/JDBC Applications. Research Demonstrations Track, *ICSE 2004*
- [9] Gould, C.R., Su, Z., Devanbu, P., "Static Checking of Dynamically Generated Queries in Database Applications", *ICSE 2004*, Edinburgh, UK. (13%). **ACM SIGSOFT Distinguished Paper**.
- [10] Toone, B., Gertz., M., Devanbu, P., "Trust Mediation for Distributed Information Systems", *Eighteenth International Information Security Conference*, 2003, (27%)
- [11] Devanbu, P., Gertz, M., Toone, B., "Static type-inference for Trust in Distributed Information Systems", *Tenth International Conference on Co-operative Information Systems (COOPIS)*, Catania, Italy, 2003. (23%)
- [12] Devanbu, P. "GENOA- a language and front-end independent source code analyzer generator", *14th International Conference on Software Engineering*, 1992. (12%). **Nominee and first-runner up for ICSE 2002 Ten-year most influential paper award**.