

Modelling Trust for Reusable Software Components

Adnan Bader, MComp
Monash University, 2005

Thesis
Submitted by Adnan Bader
for fulfillment of the Requirements for the Degree of
Doctor of Philosophy (0190)

Supervisor: Prof Christine Mings, Associate Supervisor: Dr. Sita Ramakrishnan

Abstract

The need for trust in software components is real and present. Software components have long been advertised as the prime tool for capitalizing on reusability benefits. Yet these benefits have not been realized as readily as might be possible with proper knowledge of how to gain trust in software. The lack of understanding about what trust means, whether software can be trusted at all and if yes how can it be achieved has contributed a great deal to mistrust in software. There has been some research work on trusted components, however any such work has only focussed on quality related issues with the objective of improving the product quality. In this thesis we address the very problem of trusting software. We do so with the aim of developing a methodology for consumers of software components that can encourage them to identify, document and manage their trust requirements in a component. Clearly, the trust problem is associated with some of the other well-researched areas of software engineering. We have reviewed some of these areas in order to develop and refine ideas contained in this thesis.

Significant research has been carried out in areas such as requirement engineering, software testing, metrics, software quality models and standards. All these areas contribute to the promotion of trusted usage of software components. We have founded this thesis on the well-established work in these areas to develop a methodology that provides a holistic approach to the consumers for managing their trust in software components. We examine some of this research in this thesis before investigating the main problem of trust in software in general and off-the-shelf components in particular. We then present a trust model for components that forms the basis for our proposed trust management methodology along with flexible framework to complement it.

We devolve this trust model to concrete attributes that can be monitored and managed by consumers to keep track of their level of trust in the product. The trust model allows component users to explicate their trust requirements and gives them a means of evolving trust attributes from the abstract to the concrete and monitoring them in a production environment, thus bridging the gap that has long existed between the philosophy of trust and the quantitative metrics available to measure trust attributes. This trust model means that both static and dynamic trust properties of software components are captured and monitored as they evolve over time.

