

TRUSTED SERVICE REPRESENTATION AND SELECTION FOR GENERATING DISTRIBUTED REAL-TIME AND EMBEDDED SYSTEMS

Lahiru S. Gallege, Dimuthu U. Gamage, (James H. Hill, Rajeev R. Raje)
Department of Computer and Information Science, Purdue School of
Science, Indiana University–Purdue University Indianapolis, Indianapolis,
Indiana 46202

Today's enterprise distributed real-time and embedded (DRE) systems are created from reusable software components and services. This approach is promising because of its economic advantages (e.g., maximizing reuse of existing business-logic). It, however, is plagued by the challenge of selecting a subset of software components and services from those that are readily available because the selection process can be both costly and time-consuming, and the description of available services is often ambiguous and easy to misinterpret. Moreover, there is always the chance that a selected service does not adhere to its promises. This implies that trust, which we define as the degree of confidence that a software component or service adheres to its specification, plays an important role in this selection process. We call this process *trusted selection*.

Current state-of-the-art methods use multi-level contracts made up of four levels (*i.e.*, syntax, semantics, synchronization and Quality of Service (QoS)) to facilitate service and component selection. This method, however, does not take trust into account thereby making it hard to support trusted selection. Our research therefore improves upon state-of-the-art in multi-level specification by incorporating trust contract into it. We incorporate trust into the multi-level specification by representing trust using subjective logic, which evaluates trust using a tuple of belief, disbelief, and uncertainty. Our current results show our trust-enabled multi-level specification reduces misinterpretation, mismatch, and misuse of selected services.

This work has been supported in part by the Software and Security Research Center and the United States Air Force Research Labs (AFRL).