Master Thesis Defense

Speaker:	Gopinatha Jakadeesan
Supervisor:	Dr. Goswami
Examining Committee:	Drs. Rilling, Eavis and Dr. Witte (Chair)
Title:	FT-PAS – A Framework for Pattern Specific Fault-Tolerance in Parallel Programming
Date:	Tuesday March10, 2009
Time:	3:30 pm
Place:	EV3.101

ABSTRACT

Fault-tolerance is an important requirement for long running parallel applications. Many approaches are discussed in various literatures about providing fault-tolerance for the parallel systems. Most of them exhibit one or more of these shortcomings in delivering fault-tolerance: non-specific solution (i.e., the fault-tolerance solution is general), no separation-of-concern (i.e., the application developer's involvement in implementing the fault tolerance is significant) and limited to inbuilt fault-tolerance solution. In this thesis, we propose a different approach to deliver fault-tolerance to the parallel programs using a-prior knowledge about their architectural and behavioral patterns. Our approach is based on the observation that different patterns require different fault-tolerance techniques (specificity). Consequently, a pre-knowledge of patterns and their applicable fault-tolerance techniques can facilitate an application developer for incorporating suitable fault-tolerance support. Moreover, the core functionalities of these fault-tolerance techniques can be abstracted and pre-implemented generically, independent of a specific application. Thus, it separates their design and implementation details from the application developer (separation-of-concern). One such fault-tolerance model is implemented here to demonstrate our idea. The Fault-Tolerant Parallel Architectural Skeleton (FT-PAS) model implements various fault-tolerance protocols targeted for a collection of (frequently used) patterns in parallel-programming. From the protocol extension perspective, a hierarchical design of the model facilitates a protocol designer/developer to incorporate new fault-tolerance protocols as needed to an existing pattern. The usages of the model from the perspective of two user categories (i.e., an application developer and a protocol designer) are illustrated through examples.