Software Fault Tolerance

Chapter 5: Resourceful Systems

Sima Emadi
Resourceful Systems

- It is an artificial intelligence approach, sometimes called functional diversity, that exploits diversity in the functional space available in some applications.
- The resourceful systems approach is based on self-protective and self-checking components, and is derived from an approach to fault tolerance in which system goals are made explicit.
Resourceful Systems

- Taylor and Black’s aim in [9] was to make goals explicit for the sake of protecting the system from disaster, rather than for reliability. Bastani and Yen’s work [10] focused on decentralized control, rather than on system goals.

- Resourceful systems marry these ideas and a planning component, yielding an extended RcB framework.
Resourceful Systems

- The resourceful system approach requires that system goals be made explicit and that the system have the ability to achieve its goals in multiple ways.

- A resourceful system, like a system using RcB, has the ability to determine whether it has achieved a goal (the goal must be testable) and, if it has failed, to develop and carry out alternative plans for achieving the goal.
Resourceful Systems

- In an RcB, the alternatives are available prior to execution;
- however, in the resourceful system the new ways to reach the goal may be generated during execution.
- Hence, resourceful systems may generate new code dynamically. (Obviously, dynamic code generation raises additional questions as to whether the autogenerated code is safe (in the systems context) and whether the code generator itself is dependable.)
Resourceful Systems

- These issues require additional investigation.) The resourceful system approach is based on the premise that, although the system may fail to achieve the final goal in its primary or initial way, the system may be able to achieve the goal in another way by modifying plans and operations.

- Associated with the goal may be constraints such as ‘within x time units’ or ‘within k iterations’.

- Systems using RcB may be viewed as a limited form of resourcefulness, and resourceful systems may be viewed as a generalization of the RcB approach.
Resourceful Systems

- Execute a plan
- Modify plan, generate code
- Goal achieved?
  - Yes
  - No

Sima Emadi slides for Software Fault Tolerance
Resourceful Systems

- Abbott, provides the following properties a resourceful system must possess:
  - Functional richness
  - Explicitly testable goals
  - An ability to develop and carry out plans for achieving its goals
Resourceful Systems

- Functional richness: The required redundancy is in the end results;
- It is only necessary that it be possible to achieve the same end results in a number of different ways.
- Functional richness is a property of a system in the context of the environment in which it is functioning, not of the system in isolation.
Resourceful Systems

- Explicitly testable goals: The system must be able to determine whether or not it has achieved its goals.

- This is similar to the need for ATs in an RCB technique.
Resourceful Systems

- An ability to develop and carry out plans for achieving its goals: The system must be able to reason about its goals well enough to make use of its functional richness.

- The required reasoning ability must complement the system's functionality.

- The system must be able to decompose its goals into subgoals in ways that correspond to the ways it has of achieving those goals.
Resourceful Systems

- What is desired for a resourceful system is a broad set of basic functions along with the ability to combine those functions into programs or plans.

- In other words, one wants a system organized into levels of abstraction, where each level provides the functional richness needed by the potential programs on the next higher level of abstraction.

- The system itself does the programming, that is, the planning and reacting, to deal with contingencies as they arise.
Resourceful Systems

• Abbott contends that resourceful systems would be affordable because functional richness grows out of a levels-of-abstraction object-oriented (OO) approach to system design.

• He adds that OO designs do not appear to impose a significant cost penalty and may result in less expensive systems in the long run.

• Artificial intelligence techniques are used by the system to reason about its goals, to devise methods of accomplishing the task, and to develop and carry out plans for achieving its goals.
Resourceful Systems

- The resourceful system approach tends to change the way one views the relationships among a system, its environment, and the goals the system is intended to achieve. These altered views are presented by Abbott [8] as follows:
  - The boundary between the system and the environment is less distinct.
  - The system's goals become that of guiding the system and environment as an ensemble to assume a desired state, rather than to perform a function in, on, or to the environment.
  - System component failures are seen more as additional obstacles to be overcome than as losses in functionality.
Resourceful Systems

- The following are important features for any language used for programs that control the operation of resourceful systems, but not necessarily the language in which the system itself is implemented.
  - Components;
  - Ability to express the information to be checked;
  - Error reporting mechanism;
  - Planning capability;
  - Ability to generate and execute new code dynamically.
Resourceful Systems

- Logic programming (e.g., using the Prolog language) offers the best available language resources for developing fault tolerant software.

- The application areas in which resourcefulness has been most fully developed are robotics and game playing systems.

- Intelligent agent technology may hold promise for implementing resourceful systems.

- The resourceful system approach to software fault tolerance still suffers the same problems as all new approaches, that is, lack of testing, experimental evaluation, implementation, and independent analysis.
Question ?