Towards requirements-driven autonomic systems design

Firas Alomari
CS895
Introduction

• Software systems are increasing in complexity and size.
  – Complex to management.
  – Expensive to maintain.

• Moving the management and maintenance responsibilities from humans to the software itself.

• Autonomic techniques can help address this problem
Autonomic Systems

• Three ways to build autonomic system.
  – Provide the system with planning and social capabilities so that it can delegate tasks to external software and agents.
  – Design evolutionary approaches to support autonomic system development.
  – Design the system to support a space of possible behaviors that can be selected at run time.

• Goal model requirement engineering can be used to design autonomic systems.
Goal-Oriented Requirement Engineering

• A framework used to capture stakeholders functional and non-functional requirements.
  – Non-functional (quality) requirements can be systemically represented.
  – Analyze the requirement using goals and actors.
  – Allow designers to explore explicit space of behaviors.
  – Soft goals
Goal Model
Scheduling of a meeting

- Relationships
  - And
  - OR
  - Help +
  - Hurt -
  - Make ++
  - Break --
Goal Model

• Variation points (OR)
  – Alternatives and considered the capabilities of the system
• Alternatives can be ranked based on preferences of the stakeholders
• Goal model help capture Stakeholder oriented terms as top-level goals in requirement analysis.
High Variability Software Design

• Goal model capture variability in the problem domain.
• Properly annotated goal model can represent variability in the solution domain.
• Goal model generates three design views
  – Feature Model: configuration variability.
  – State Chart: behavioral variability.
  – Component Connector Model: structural variability.
Models

Figure 2. A feature model generated from the goal model of Figure 1.

Figure 3. A fragment of the statechart generated from the goal model in Figure 1.
Goal Models to Autonomic systems

- Goal models represent many ways in which the objective can be met at the design time.
- Goal model allows for analyzing and ranking of the behavior space.
- Provide traceability mechanisms for AC system design to stakeholder requirements.
- Relate AE assigned goals to high level objectives.
Hierarchical Autonomic Architecture

- Similar to the goal hierarchy
- Tree structure
- Knowledge can represents high level goals

Figure 5. A hierarchical composition of AEs.
Autonomic system

• Advantages of Autonomic hierarchy similar to the goals hierarchy.
  – Partition the system variability.
  – Goal propagate from parent element to child elements
  – Child element can apply any set of actions if it’s within the parent policy.
  – Root element can change behavior if the child AE is not meeting some objectives.

• Employ goal reasoning approach
  – Top-down: find alternative that satisfy root goals.
  – Bottom-up: determine if an alternative meet system requirements.
Autonomic Behavior
Scheduling of a meeting

• Self-Configuration and Reconfiguration:
  – Provide feedback information to the system.
    • Cancelled, postponed, initiator, participant.
  – If certain criteria is met (cancelled meeting is more than 10%) find the cause of the problem (incorrect time tables).
  – Switch to a different behavior (manual time tables collection)
  – Human involvement is important in application software.
    • Stakeholder oriented goals vs system oriented goals.
Autonomic Behavior
Scheduling of a meeting

• Self-Optimization
• Feedback provide information like meeting evaluation survey.
  – Good quality meeting, productive meeting
• System switch to another behavior for improved feedback
  – Mainly address soft goals.
• Preferences of the stakeholder
Autonomic Behavior
Scheduling of a meeting

• Self-Repair examples
• Application level failure (missing time tables)
• Diagnose the problem
  – User not responding to requests
  – travel
• Switch to another behavior
  – Personal contact for time tables
  – Revise the list of participants