Components and Aspects for Embedded Middleware Product Lines

**Problem Statement:**
Embedded systems developers can use RTZen, a CORBA implementation in Java, to satisfy complex requirements found in distributed, real-time, and embedded (DRE) systems. However, a full-featured CORBA implementation may not satisfy stringent requirements of a highly constrained embedded system: the resource requirements are too great, and unwanted features may reduce run-time performance.

**Approach:**
Component-based software development (CBSD) and aspect-oriented programming (AOP) can help reconfigure and improve modularity of the RTZen platform.

**Thesis:**
Jiazzi can modularize the implementation of concerns within RTZen and assemble the resulting modules in ways that exclude features not needed for particular applications.

**Concepts:**

- **Component-Based Software Development (CBSD)** describes software as assemblies of fully encapsulated software parts. CBSD tools can build different applications by connecting components according to their application programming interfaces (APIs), but it cannot modularize crosscutting concerns between components.

- **Aspect-Oriented Programming (AOP)** identifies and isolates the implementation of crosscutting concerns like security, logging, and domain-specific features. By modularizing these concerns, AOP enables better programming practices.

**Process:**
- Identify components and crosscutting concerns in RTZen.
- Adapt code for product line.
- Create Jiazzi “atoms” and “compounds.”
- Describe targets & build them from product line.

**Current Tasks Detail:**
Better understanding of what "components + aspects" means in general
- Improve Jiazzi to support AOP-like constructs.
- Componentize crosscutting concerns in RTZen with Jiazzi:
  - ... Support for anys.
  - ... Support for wstring.
  - ... POA policies.
  - ... Real-time implementation strategies.
  - ... Supported I/O protocols.

**Motivation**

- A Portable Object Adapter (POA) provides an additional layer of insulation for the high-level systems. It functions to provide object translation, persistence, object factories, and server reusing.

**Principles**

- CORBA middleware provides a standardized conduit for client-server communication over a network. Its power lies in its platform agnosticism and well-tested, rich functional suite. Many parts of a CORBA system can be componentized for system configuration.

**Results**

- The Interface Definition Language (IDL) describes client and server. A compiler generates client stubs and server skeletons from an IDL file.

**Challenges:**
- Componentization is problematic due to the high subsystem cohesion.
- Some concerns span multiple programs (e.g., IDL compiler and RTZen runtime).
- Jiazzi appears to lend itself to new development; legacy systems present unique challenges.