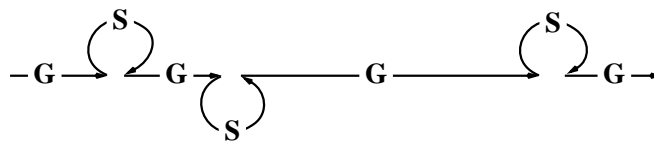


Possible Project Ideas

- Developing a DSSA
 - suggested domain: library based applications
 - data from your CS577a course
 - method described by Tracz
- Analysis of architectures
 - apply the SAAM method to the CS577a projects
 - show how the different solutions compare
- Develop a framework for understanding connectors
 - a taxonomy
 - denote the underlying conceptual mechanisms
 - determine the dimensions of variability
 - (where) do the dimensions overlap?
 - (where) are there undefined points in the multi-D space?

Possible Project Ideas (cont.)

- Extend the MBASE process to use (an) ADL(s)



- Mapping UML to an ADL
- Comparing architectural models within a notation
 - model a “real” architecture in UML (e.g., C4 Case Study)
 - what are the architectural mismatches?
 - what are the techniques for identifying/remedying them?
 - are any of the techniques automatable?
- Comparing architectural models across notations
 - model an architecture in an ADL *and* in UML
 - the rest of the problem is the same as the above

Possible Project Ideas (cont.)

- An ADL-independent internal representation for architecture interchange
 - use XML
 - semantically rich representation
 - enables description of the markup of different types of documents
 - extensible — new tag and attribute names for documents defined by specifying their syntax and semantics
 - structured — documents can be containers for other documents, with arbitrary nesting
 - verifiable — a document can include a description of its grammar to validate that it conforms to its specified structure
 - what are the benefits/drawbacks in comparison to ACME?
 - what is the relationship to UML?
 - (how) does XML help resolve architectural mismatches?

Possible Project Ideas (cont.)

- “Reverse-architecting”
 - a commonly used tool
 - e.g., Netscape, MS Word, Rational Rose
 - implement a subset of the original application based on the architecture
 - demonstrate the advantages of your design
- An architecture for a simulator
 - simulates a hardware execution environment
 - implement a meaningful subset
- Architecture-based OTS tool integration
 - integrate at least three heterogeneous tools
 - you may use any approach (e.g., C2)
 - what lessons did you learn w.r.t. the Garlan et al. “architectural mismatch” paper?

Possible Project Ideas (cont.)

- Enhancements to the DRADEL tool suite
 - a complete GUI front-end for DRADEL
 - use a GUI builder (e.g., GEF)
 - GUI-based mapping from method to message names
 - more flexible ADL parser
 - support for functions in the parser
 - hyperlinks into files to inspect type mismatches
 - highlight sources of mismatch
 - suggested mismatch fixes
- Integration of DRADEL and Robusta
 - promoting DRADEL-generated pre- and postcondition comments into automatically checkable assertions
- Develop alternative DRADEL *Code Generator* components
 - JavaBeans, CORBA or DCOM compliant interfaces

Possible Project Ideas (cont.)

- Change C2SADEL to use StateChart-based semantics
 - possibility of automatically generating the *entire* application
 - are the pre- and postconditions retained?
- Add protocol specs to C2 connectors
 - use, e.g., StateCharts, CSP, posets, ...
 - modeling transactions
 - (automatic) implementation of thus modeled connectors
- Delimit the “design space” for ADLs
 - what dimensions are there?
 - you may base it on the Medvidovic/Taylor taxonomy
 - may have a high risk of failure
- Internet-related ideas
 - adaptable Web server
 - adaptable Web browser