Component Integration

- Integrating components can be illustrated as a mechanical process of "wiring" components together to form assemblies.

- Standardization in form of component models like EJB, CORBA and COM.

- Still difficult to make components play well together.

"Architectural mismatch stems from mismatched assumptions a reusable part makes about the structure of the system it is to be part of. These assumptions often conflict with the assumptions of other parts and are almost always implicit, making them extremely difficult to analyze before building the system."

Component Integration

- Two key research papers:
    - AESOP
  - P. Inverardi, A.L. Wolf, and D. Yankelevich, Static Checking of System Behaviors Using Derived Component Assumptions
    - Compressing proxy

AESOP describes four classes of structural assumptions:

- The nature of components (infrastructure, control model, and data model)
- The nature of connectors (protocols and data models)
- The architecture of the assemblies (constraints on interactions)
- The run-time construction process (order of instantiations)

Component Integration

- Components must express assumptions made about it's environment such as data formats or buffer sizes
- Careful analysis of the assumptions components make about the context in which they are to be used can help prevent design errors and result in building the right product faster
From Integration to Composition

- All assemblies are potential subsystems
- *Predicting the emergent behavior of assemblies*
- The result of component composition is a component assembly which can be used as a part of a larger composition
- Composition goes beyond integration by allowing prediction of the emergent behavior of assemblies

Predictable Assembly from Certifiable Components (PACC)

- What types of system quality attributes are developers interested in predicting?
- What types of analysis techniques support reasoning about these quality attributes, and what component property values do they require as input parameters?
- How are these component properties specified, measured, and certified?

Prediction-Enabled Component Technology (PECT)

- A prediction-enabled component technology consists of a component model and an associated analysis model
- PECT integrates ideas from research in the areas of software architecture, component certification, and software component technology to create a prediction-enabled component technology
- Prediction-enabled component technologies exploit the relationship between structural restrictions and assumptions of analysis models to compute properties of assemblies based on trusted properties of the assembly’s constituent components.
Architecture-based Analysis

- Provides a foundation for reasoning about system completeness and correctness early in the development process and at a high level of abstraction
- Architecture Description Languages (ADL)
- Analytic models capture the fundamentals of compositional reasoning techniques: assumptions of the algorithms used as well as the algorithms themselves.
- Specification Refinement
- Mappings between analytic models and constructive models

Component Certification

- The National Security Agency (NSA)
- National Institute of Standards and Technology (NIST)
- Used the trusted computer security evaluation criteria (TCSEC), a.k.a. "Orange Book." as the basis for the Common Criteria, which defines criteria for certifying security features of components

Component Certification

- Trusted Components Initiative (TCI)
  - Loose affiliation of researchers with a shared heritage of experience in the formal specification of interfaces
- Use of pre/post conditions on API
- Quality attributes, such as security, performance, availability, and so forth, are beyond the reach of these assertion languages
  - Voas has defined mathematical models of component reliability based on statistical approaches to testing
  - Shaw proposed component credentials as an open-ended, property-based interface specification.
    - A credential is a triple <attribute, value, knowledge>

Architectural Styles and Component Models

- Architectural styles are sets of component types and constraints on how instantiations of those types can be interconnected.
- Component technology provides the means of realizing architectural style constraints in software
- Component models can be Architectural styles
  - Enterprise Java Beans