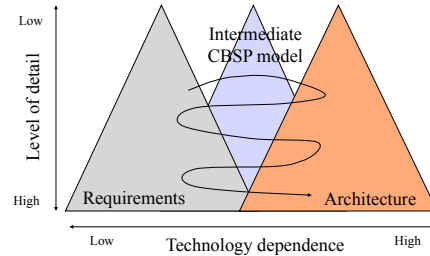


## Background: Conventional CBSP Method

- Iterative method of moving from requirements to architecture using USC's intermediate CBSP models
  - CBSP = component-bus-system-property
- Uses USC's WinWin requirements elicitation

**Our goal: Refine integration requirements into an integration proto-architecture, while retaining strong traceability between them**



## Solution Approach: Integration-CBSP Method

- Reconceptualize CBSP in the context of integration, mergers, and interoperability
  - C = constituent systems
  - B = integration and interoperability mechanisms
  - S = merged, integrated system or SoS
  - P = properties of the above

- Pre-step: filter requirements for integration*
- Step 1: stakeholders rate importance and feasibility*
- Step 2: architects rate architectural relevance*
- Step 3: architects negotiate and reconcile disagreements*
- Step 4: requirements rephrased and traced to proto-architecture*

## iCBSP in Action: Jail Information Management System (JIMS)

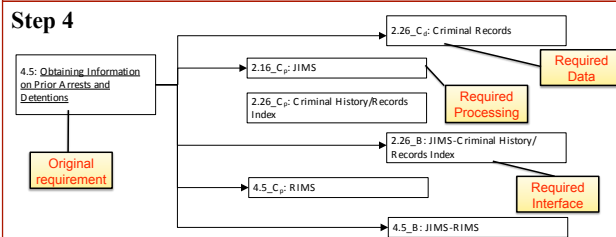
- Provides data consistency and availability at seven San Diego County detention centers
- JIMS needs to interoperate with multiple external systems (Warrant search systems, Criminal history system, etc.)

**JIMS requirement 4.5:** When the RIMS criminal history module initially becomes available, JIMS shall interface with both it and the Records Index and the Criminal History Systems. This dual interface shall be maintained until such time as all records in the Records Index and the Criminal History Systems have been moved into RIMS.

**Step 1**

	Architect	End user	Average	Disagreement
Importance	2	3	2.5	1
Feasibility	1	1	1	0

High importance requirement with low disagreement



**Steps 2 and 3**

	C	B	S	CP	BP	SP
Architect 1	3	3	0	0	0	0
Architect 2	0	3	0	2	3	2
Diff	3	0	0	2	3	2

High disagreement between the architects, reconciliation necessary

	C	B	S	CP	BP	SP
Architect 1	2	3	0	0	0	2
Architect 2	2	3	0	0	0	3
Diff	0	0	0	0	0	1

**Reconciled table**

Requirement related to constituent systems, their interconnections, and the interface availability

## iCBSP Results Summary

- Effectively filters requirements
  - 1800 total requirements → 22 integration requirements
- Supports requirements reconciliation and rewriting
  - 6 requirement with high architect disagreement
- Precisely relates integration requirements and architecture
  - 16 processing elements, 16 buses, 11 data elements
- Effectively maintains explicit traceability

## Contact

Ivo Krka ([krka@usc.edu](mailto:krka@usc.edu)), <http://www.scf.usc.edu/~krka/>  
Department of Computer, University of Southern California, 941 W. 37th Place  
Los Angeles, CA 90089, USA

## References

P. Grunbacher, A. Egyed, and N. Medvidovic. Reconciling software requirements and architectures with intermediate models, Software and Systems Modeling, 2004.