



Current SAT Work in Architecture Evolution

Rick Kazman

representing the work of:

Joe Batman, Andrés Díaz-Pace, Mark Klein, Robert Nord, Ipek Ozkaya



Software Engineering Institute

Carnegie Mellon

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE MAY 2008		2. REPORT TYPE		3. DATES COVERED 00-00-2008 to 00-00-2008	
4. TITLE AND SUBTITLE Current SAT Work in Architecture Evolution				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Carnegie Mellon University ,Software Engineering Institute (SEI),Pittsburgh,PA,15213				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES presented at the SEI Software Architecture Technology User Network (SATURN) Workshop, 30 Apr ? 1 May 2008, Pittsburgh, PA.					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 15	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Our View of Architecture Evolution

Architecture evolution is the process of designing an architecture to meet today's and tomorrow's business goals, while maximizing expected value, in the face of uncertainty.

Architecture evolution therefore has two foundations:

- Architecture design
- Software engineering economics

In this talk I will sketch our approaches to both.

Evolution with a Design Focus

The *design aspect* of architecture evolution investigates how to measure and analyze the difficulty of reaching a desirable future state given today's state and how to design the system so that desirable states are easier to reach.

We wish to be able to answer questions such as:

- *How can the quality of a deficient architecture be improved?*
- *What alternative architectures provide the greatest flexibility to support future requirements?*
- *How resilient is the system to change?*

Evolution with a Design Focus – 3

Consider the following design scenario:

- We have an important feature to implement for our new system
- We intend to implement it in the 2nd release. This feature depends on an “unimportant” piece of infrastructure, which doesn't get done in release 1.
- This means that you have to do the unimportant infrastructure in release 2, and delay the feature to release 3 (by which time it is of less value).

What is the point of this example?

In designing for evolution we need to think about trajectories and dependencies, not single design steps.

Shameless plug: we will be discussing just such situations in the Evolution Workshop. Come join us!

Evolution with an Economics Focus

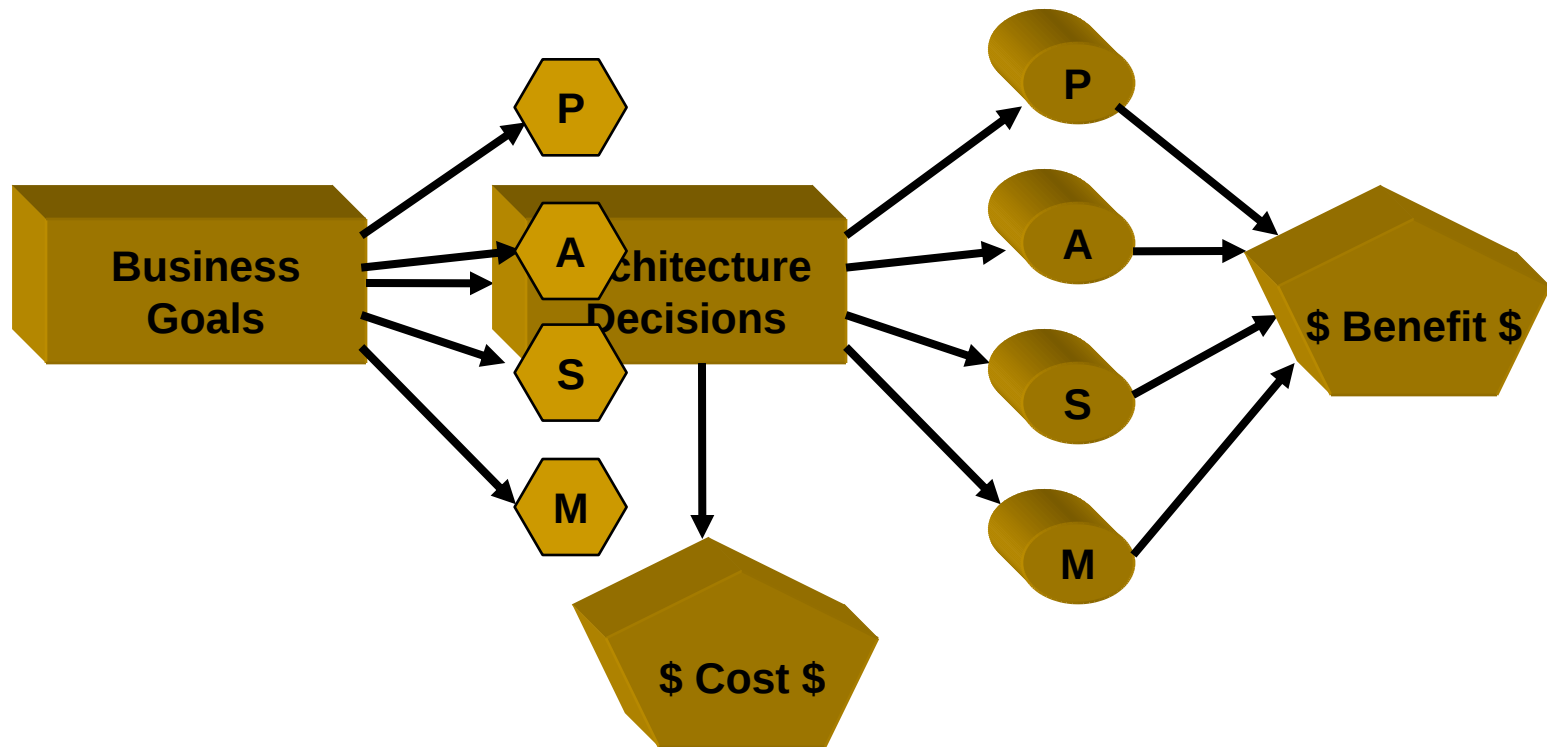
Software architecting cannot take place in a value-neutral setting.

Architecture decisions are the results of trade-offs and economic factors influence such trade-offs constantly.

Economics-driven architecting is a direction that must be rigorously pursued to produce systems that meet *all* their business goals. It is based on the following principles about architecture:

- *Principle 1:* Quality attribute requirements are a driving force for software and system architecture design.
- *Principle 2:* Using architectural patterns and tactics is a way to achieve quality in software.
- *Principle 3:* Business and mission goals endow quality with value.

Evolution with an Economics Focus - 2



Evolution with an Economics Focus - 3

Architecture evolution involves:

- exploring the design space, looking for risks and opportunities
- maximizing expected value of “reachable” designs under conditions of uncertainty
- minimizing the expected loss of “unreachable” designs under conditions of uncertainty
- staging of architectural strategies and features (release planning)

Related Work in Evolution

		DIMENSIONS			
APPROACHES		Transformation	Value	Sequencing	Uncertainty
<i>practice</i>	code metrics and refactoring	■			
	product lines *	■		■	□
	quality-attribute design and analysis*	■			
	ATAM® and QAW*	■			
	enterprise framework and migration planning*		□	■	
<i>transition</i>	model transformations	■		□	
	tactics and patterns *	■			
	impact analysis	■	□		
	plateaus and waves		□	■	□
	value-based software engineering		■		□
	CBAM*	■	■		
	history-based feature analysis				■
	conceptual framework for uncertainty		□		□
KEY: Relevance to architecture evolution: ■ - very relevant, ■ - relevant, □ - somewhat related. Square size indicates amount of activity/efforts. * - SEI-related approaches.					

Our Work in Evolution (with an Economics Focus)

Cost Benefit Analysis Method (CBAM)/Utility theory

Real options for valuing flexibility

Release planning and optimization

Architecture patterns/tactics as value-creating design operations.

Common to all the work is the application of *value-based* activities

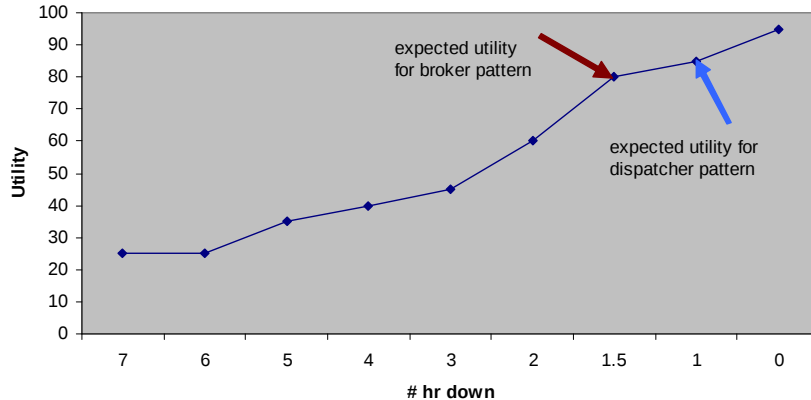
- that are practical and easily implemented
- on a firm, principled basis
- with simple, clear rationale

Consider how economics enters into evolutionary design decisions...

Quality Attributes, Utility and Business Goals

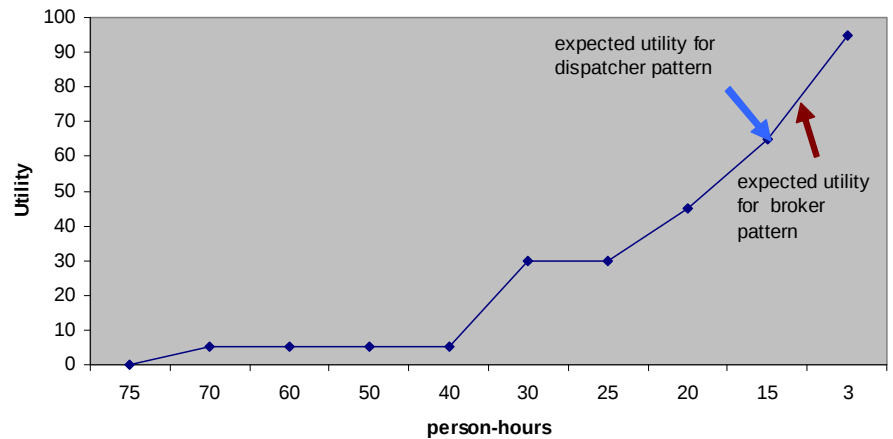
Different patterns exhibit different quality attributes which in turn offer different levels of utility.

Utility of Availability Scenario (Scenario 2)
Response Measure Goals



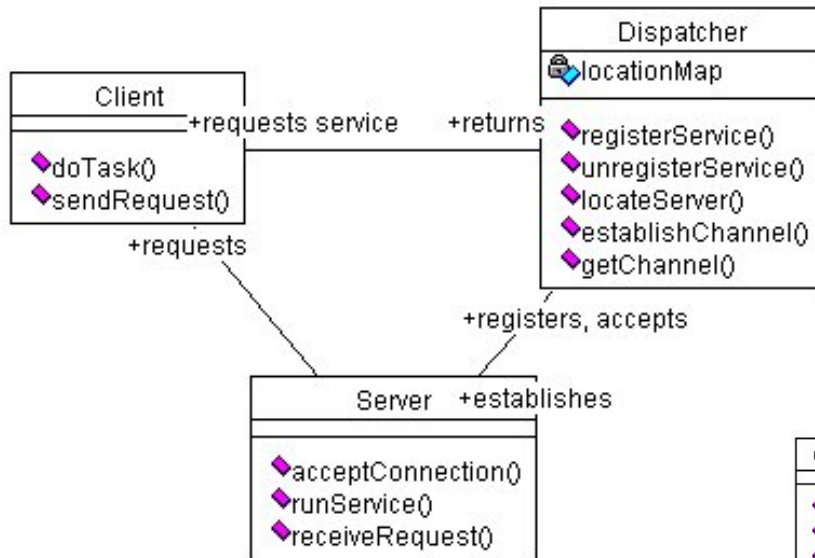
- ➡ Utility level for broker pattern
- ➡ Utility level for dispatcher pattern

Utility of Modifiability Scenario (Scenario 1)
Response Measure Goals



Future utility curves are affected by uncertainty.

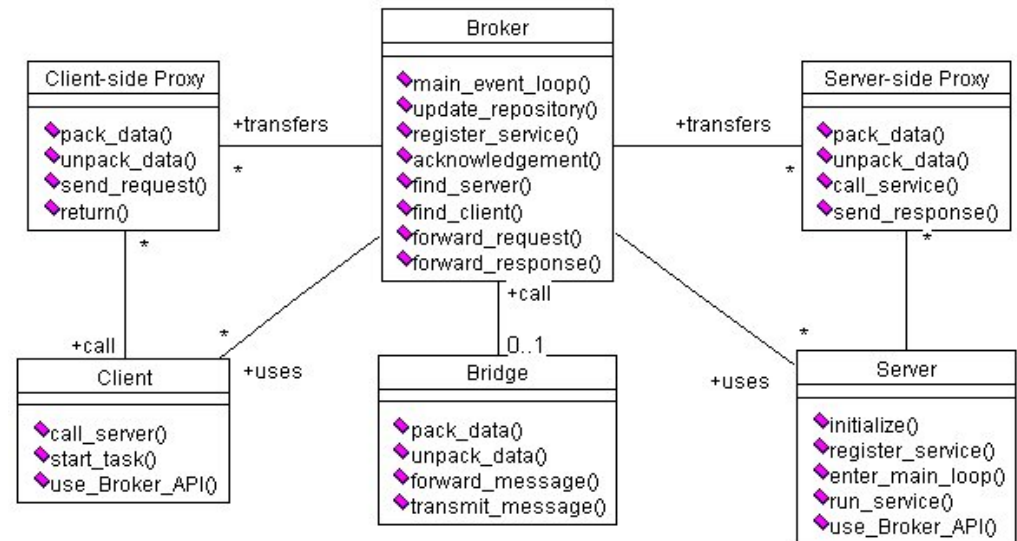
Economics Influences Architectural Strategies



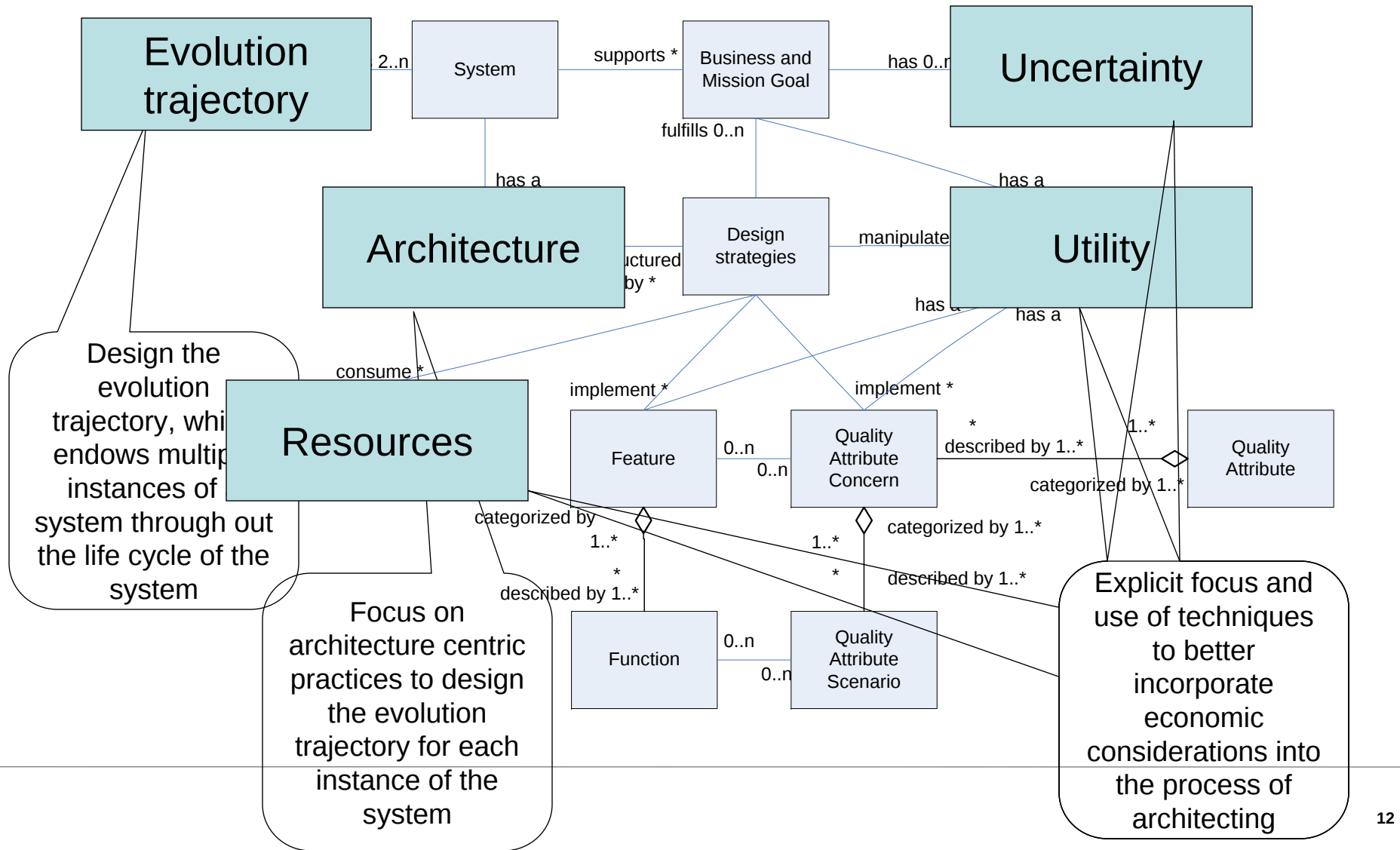
It is necessary, but not sufficient to consider the strengths and weaknesses of patterns with respect to quality attributes

Economic considerations

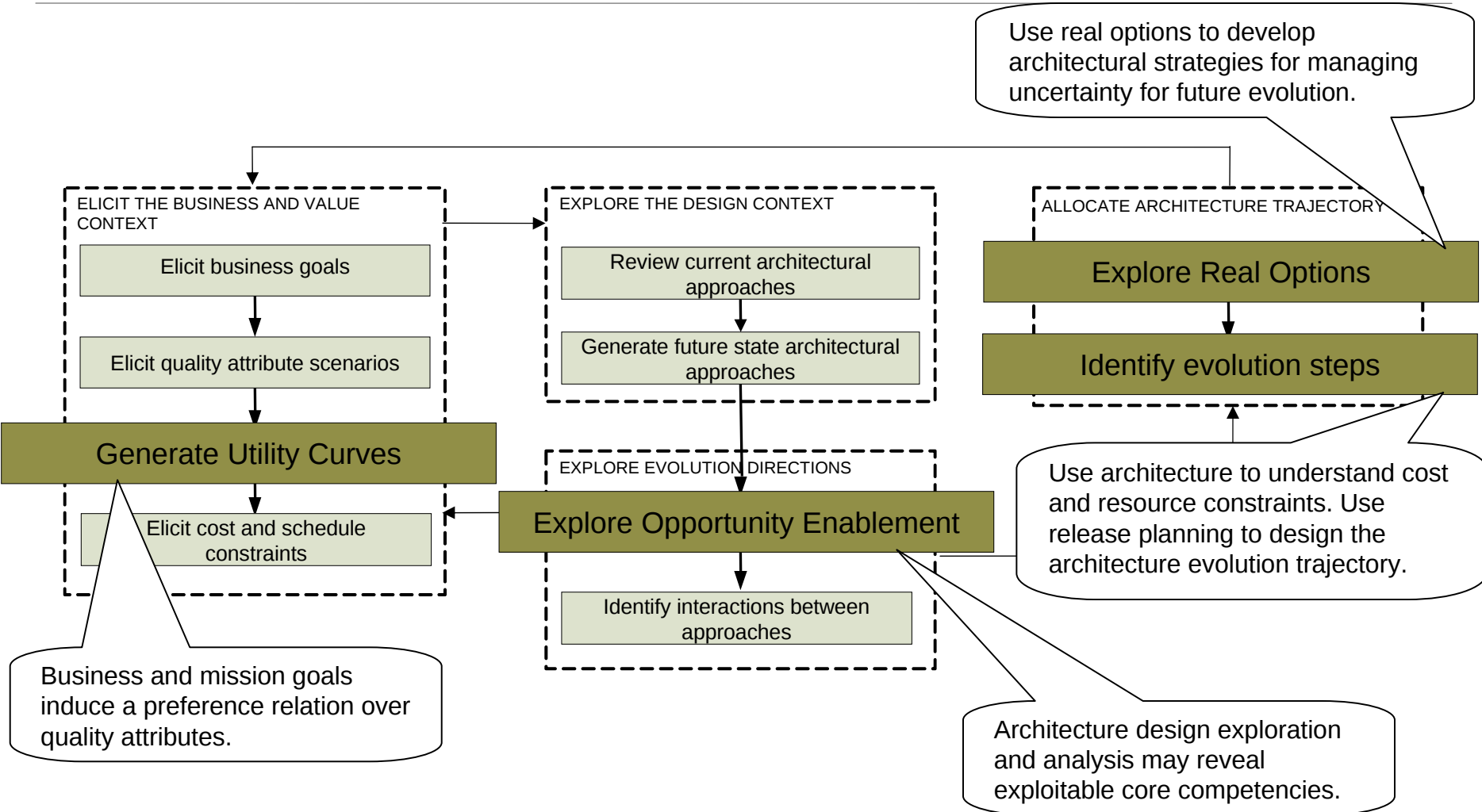
- Current business goals
- Uncertainty in future business goals



Key Concepts in Evolution



A Proto Architecture Evolution Method



Goals of Our Ongoing Work

Here are some of our goals for the coming year:

- Package our existing techniques as a method.
- Give guidance to practitioners to allow them to reason about uncertainty vis-a-vis architecture.
- Give guidance on the ramifications of design decisions and trajectories of design decisions.
- Aid practitioners in making a business case for the value of architecture.
- Provide tool support for exploring the design space.

Further Reading

- R. Kazman, J. Asundi, M. Klein, [Making Architecture Design Decisions: An Economic Approach](#), *Software Engineering Institute Technical Report CMU/SEI-2002-TR-035*.
- M. Moore, R. Kazman, M. Klein, J. Asundi, "Quantifying the Value of Architecture Design Decisions: Lessons from the Field", *Proceedings of ICSE 25*, May 2003.
- Nord, R.; Barbacci, M.; Clements, P.; Kazman, R.; Klein, M.; O'Brien, L.; & Tomayko, J. (CMU/SEI-2003-TN-038) Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004.
- http://www.sei.cmu.edu/architecture/products_services/cbam.html
- Ozkaya, I., Kazman, R., Klein, M. (2007) [Quality-Attribute Based Economic Valuation of Architectural Patterns](#), *Software Engineering Institute Technical Report CMU/SEI-2007-TR-003*