Current SAT Work in Architecture Evolution

Rick Kazman

representing the work of:

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

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

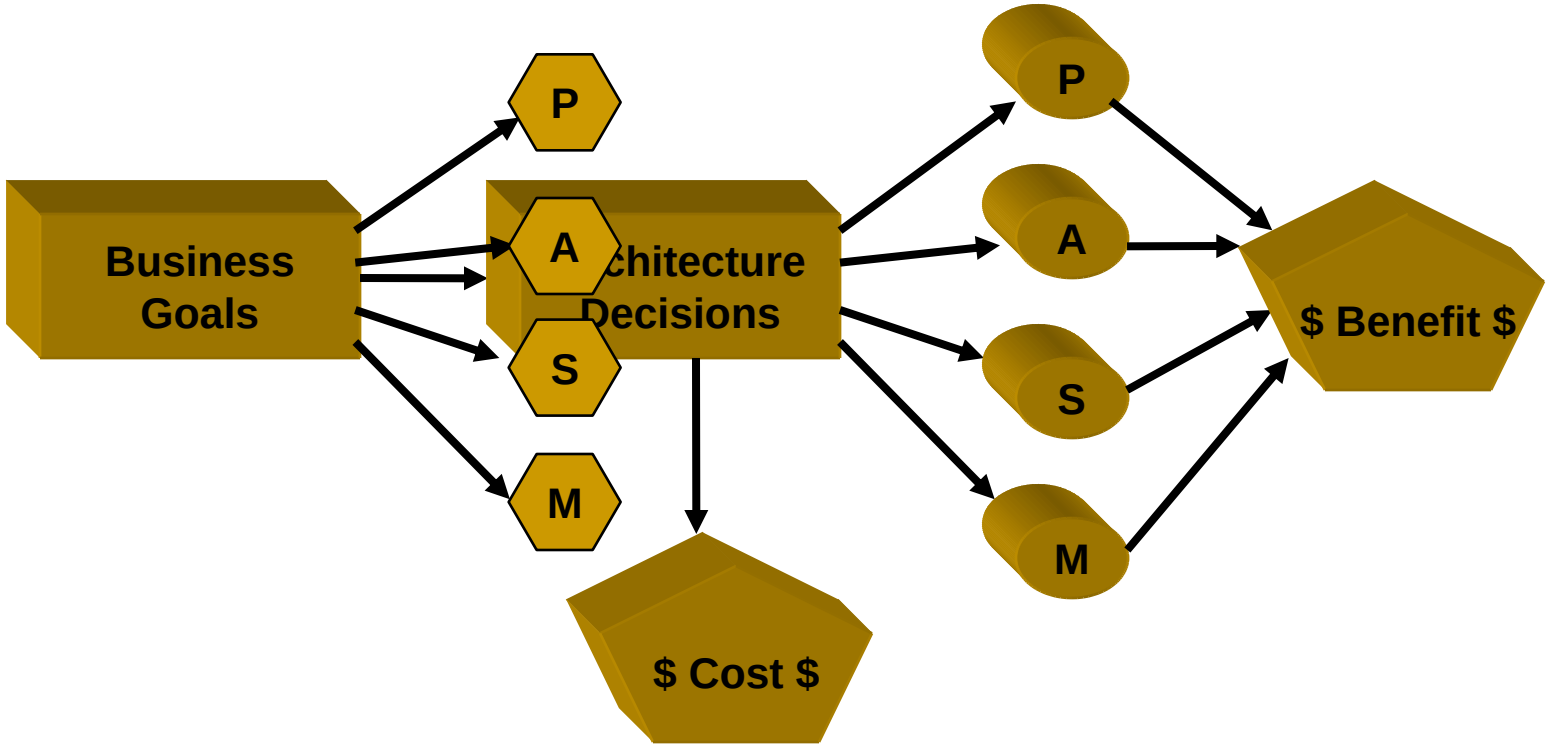
Software architecting cannot take place in a value-neutral setting.

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

	APPROACHES	DIMENSIONS: Transformation	Value	Sequencing	Uncertainty
practice	code metrics and refactoring	■			
	product lines *	◘		◘	□
	quality-attribute design and analysis*	■			
	ATAM® and QAW*	■			
	enterprise framework and migration planning*		□	◘	
transition	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.

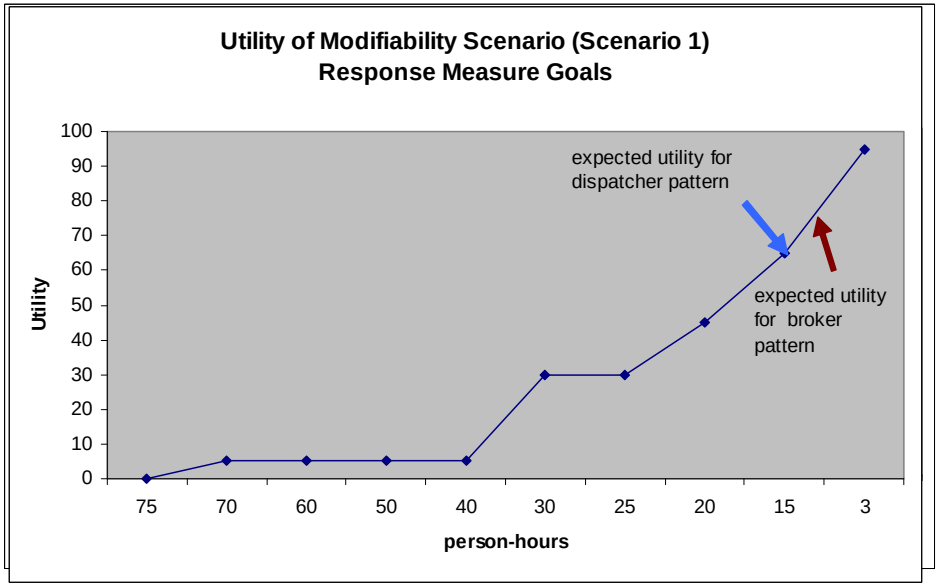
Common to all the work is the application of *value-based* activities

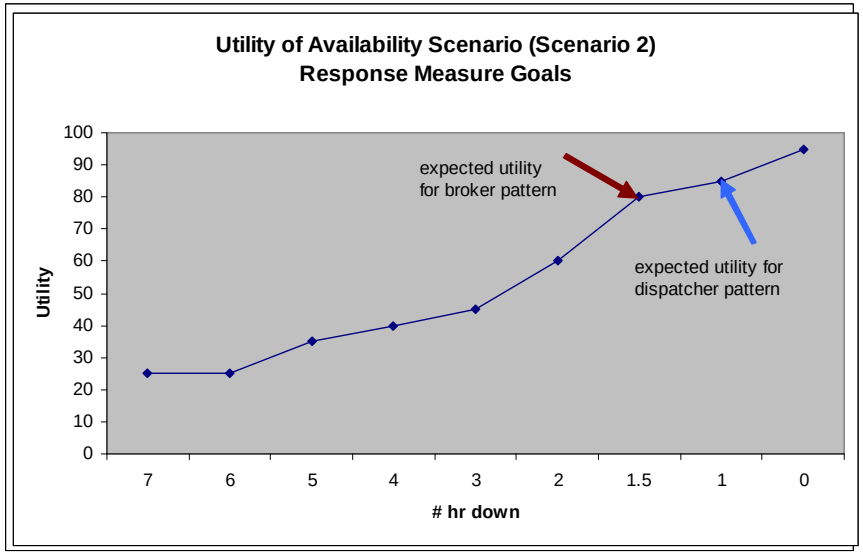
- that are practical and easily implemented
- on a firm, principled basis
- with simple, clear rationale

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.



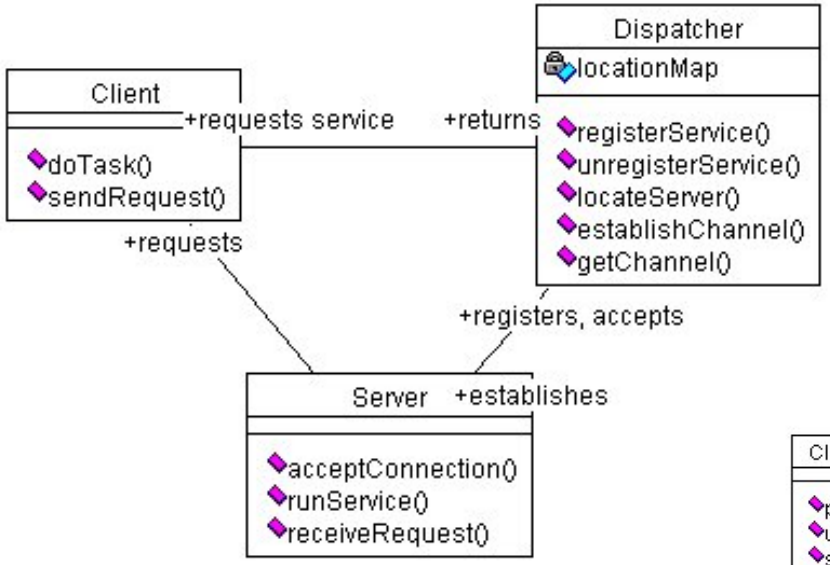


Future utility curves are affected by uncertainty.

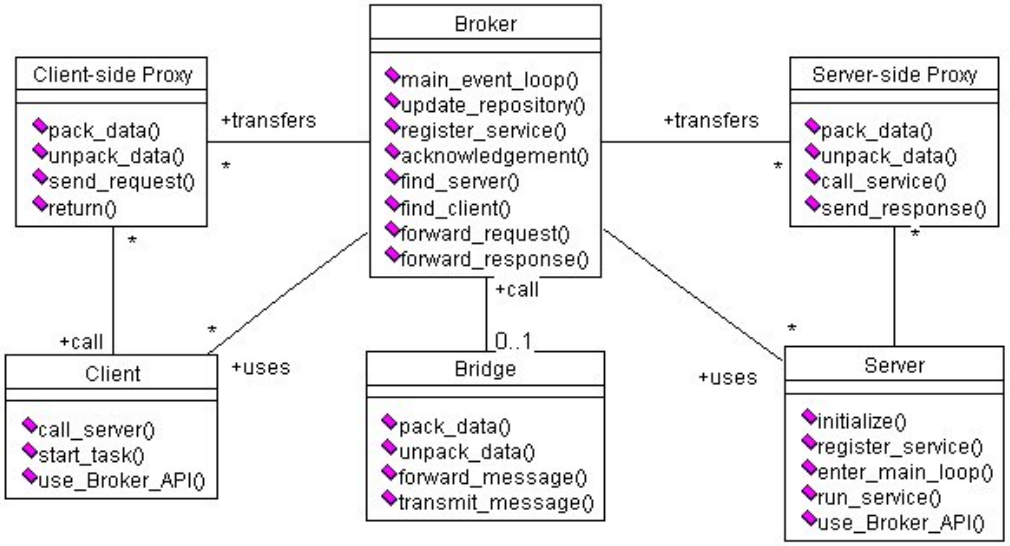
Utility level for broker pattern

Utility level for dispatcher pattern

Economics Influences Architectural Strategies



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

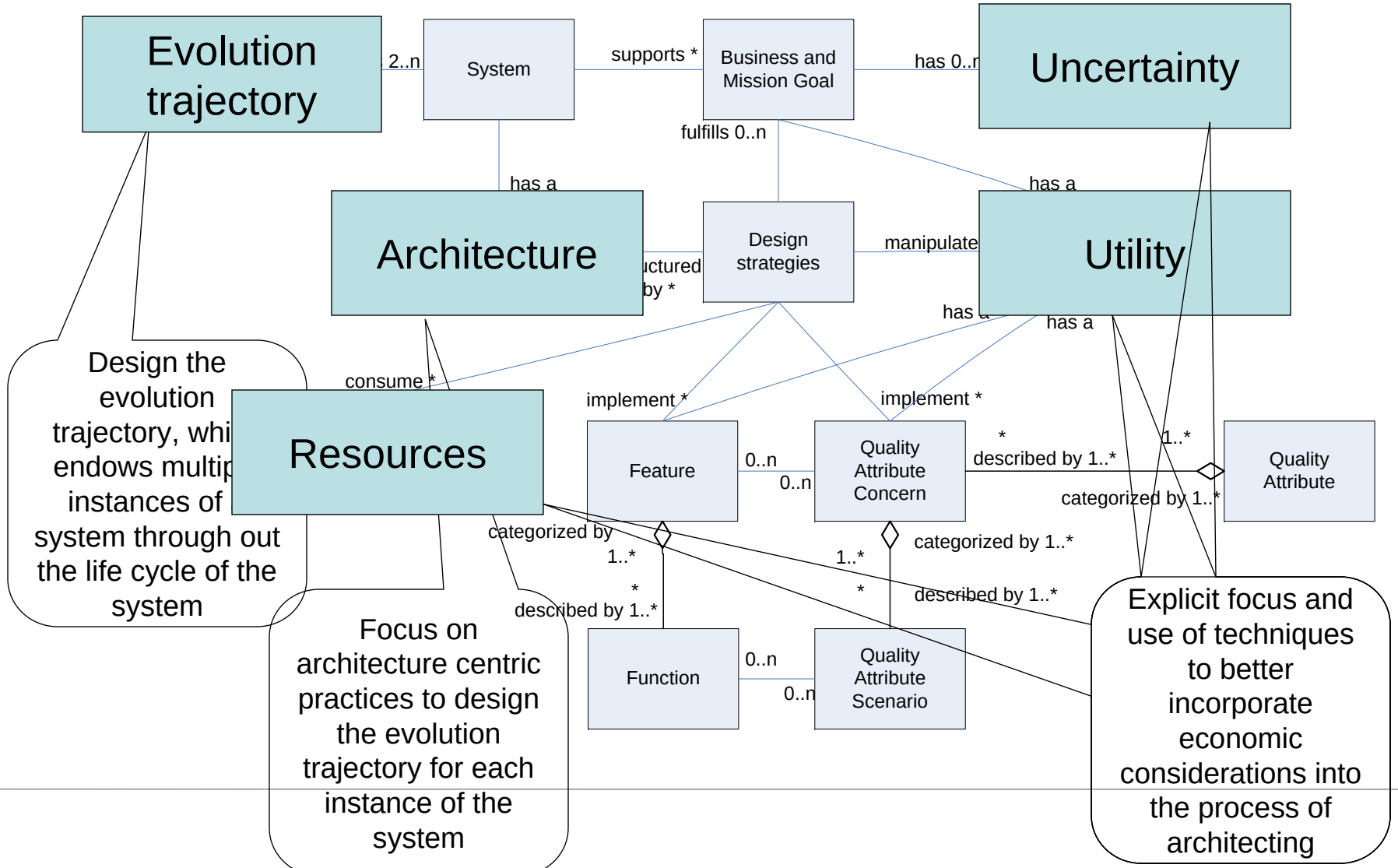


Economic considerations

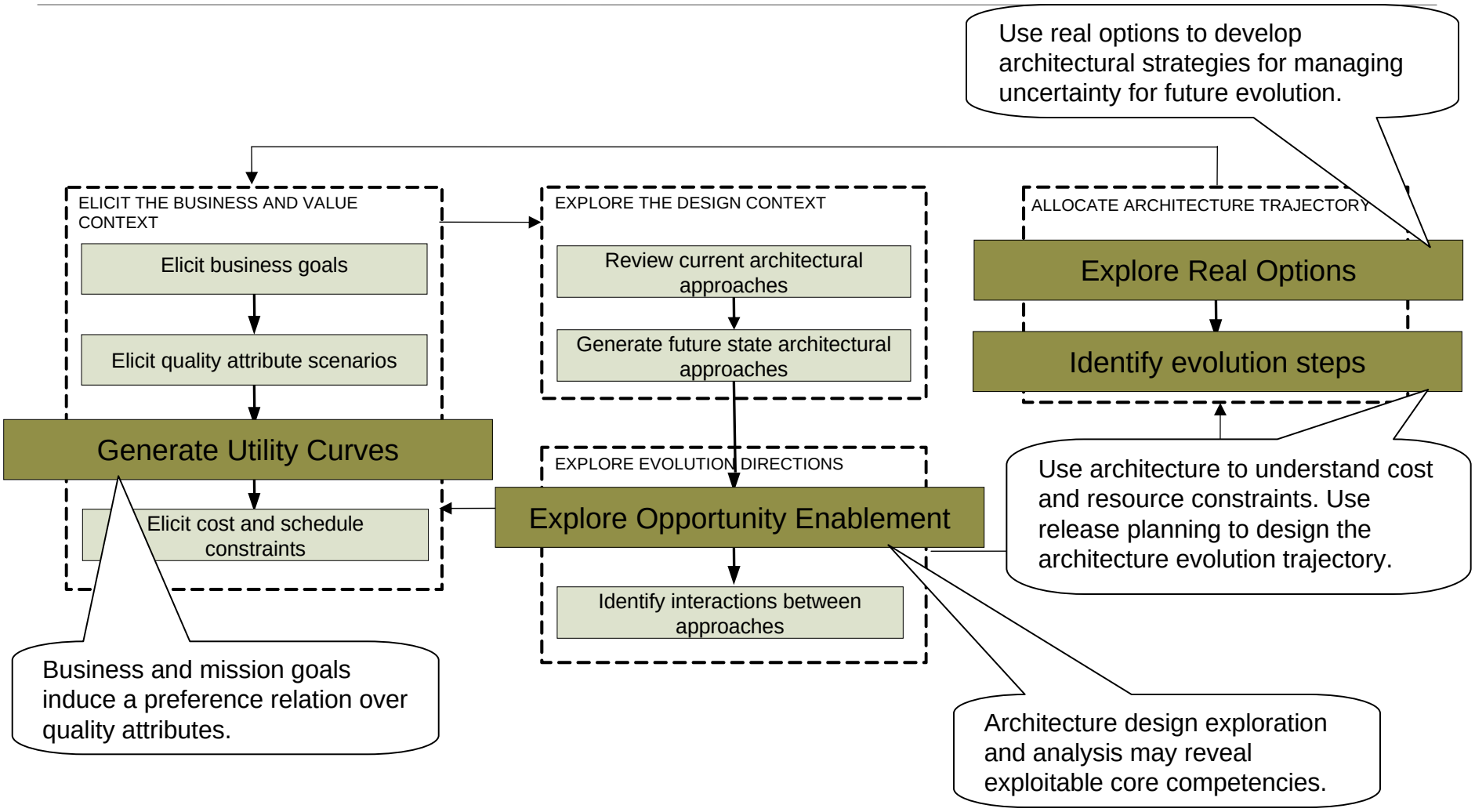
- Current business goals
- Uncertainty in future business goals

Source: Buschmann, F., R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. Pattern-Oriented Software Architecture: A System Of Patterns. West Sussex, England: John Wiley & Sons Ltd., 1996

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