Designing to Attain Business Process Flexibility

Hitoshi Ozawa

Dec.12, 2009

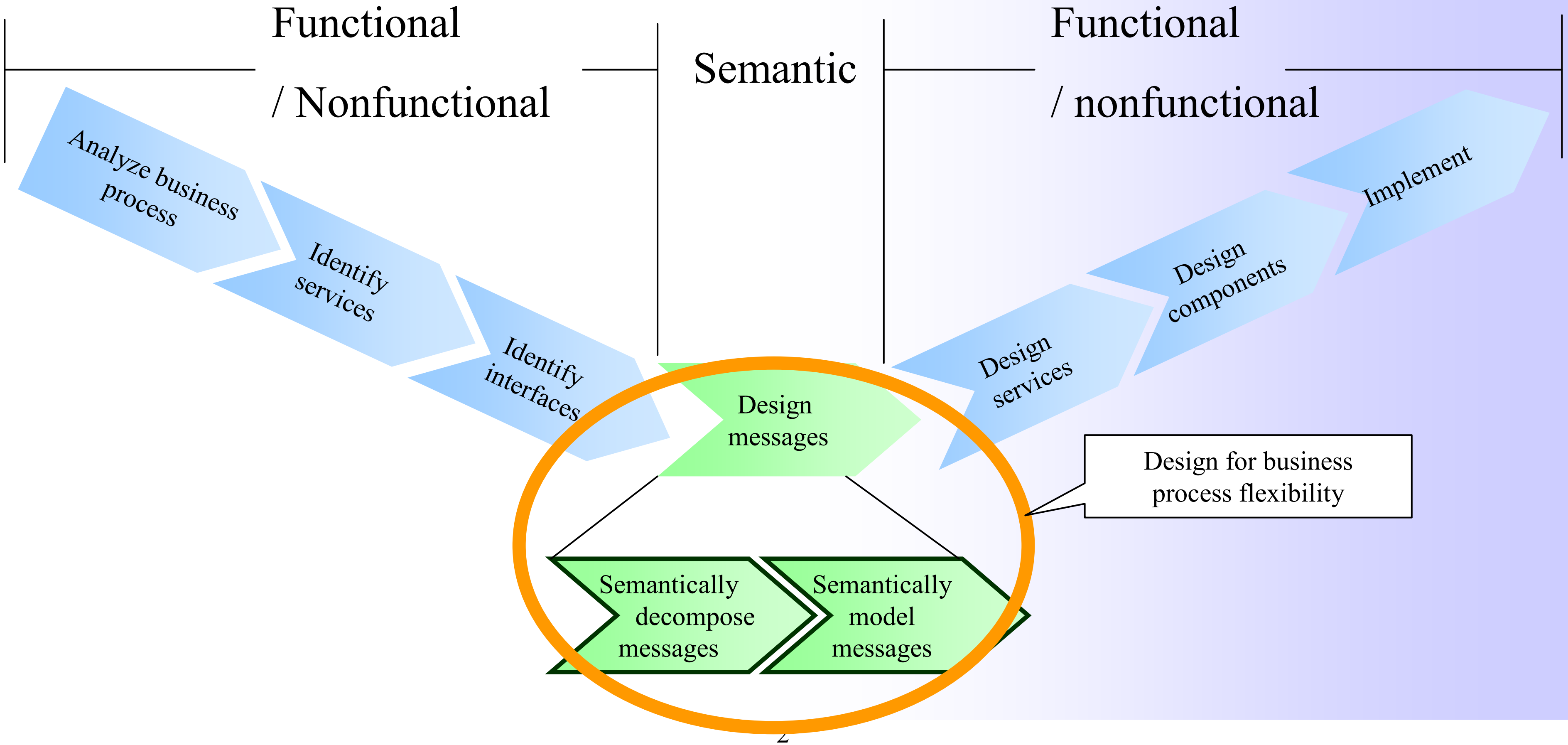
Goal of this presentation

The goal of this presentation is to present a development methodology to attain better business process flexibility

- Define development phase where business process flexibility should be considered
- Define targeted changes to identify services/interfaces that makes business process flexible
- Define methodology to design more flexible services
  - Procedure and tools to consolidate/decompose services/interfaces

Development methodology

1. Services are identified by functional/nonfunctional requirement of a business.
2. Semantic model of a message between services are used to adjust service/interfaces to attain flexibility.



Identifying relevant services and interfaces

Identify and classify services and interfaces that highly impact business process flexibility.

Type of changes	Description	Strategy
■ Anticipated modification		
– By design △	: deferred or anticipated changes with planned implementation strategies	Can incorporate into initial design
– **By under specification** ○	: anticipated changes without concrete implementation strategies	Decision contribute most to whether flexibility is attained
■ Unforeseen modification		
– By deviation △	: unforeseen changes but with minimal impact	Enable adjustment to infrastructure
– By change ✕	: unforeseen radical changes requiring adaptation	Unable to design for this type of changes

Types of “reusability” to be considered

- Different definition of “reusability”

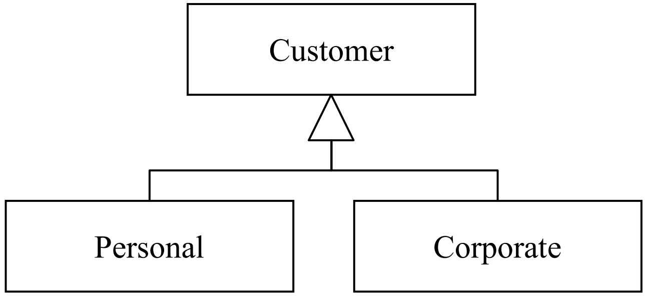
Change by design / by deviation → 1. Exchange current service with same/similar functionality in the same process

Change by under specification → 2. Exchange current service with a service with different functionality in the same process

Minimal impact to flexibility ~~3. Usage of service in other processes~~

Granularity problem

- To increase “reusability”, developers creates an interface model with coarser granularity



getCustomerInfo(List Customer)

getCustomerInfo(Personal Customer)

getCustomerInfo(Corporate Customer)

Provider and consumer need to predefine content of List so they can be properly processed.

Provider and consumer become tightly coupled resulting in less flexible process.

Semantic model of message assists to stop developers from designing coarse service with incompatible interfaces.

Notes

Definition: *Flexibility of a business process* is the ability to change a section of a process with minimum impact and minimum effort.

Goal: How to reduce impact on semantic model of messages

Flexibility is not a function of service functionality

It is possible to change function of a service with minimum impact if their semantic model of messages are equivalent.

Semantic decomposition method

- In a large project, UML diagrams may contain too many elements to efficiently model
- UML is good when doing top down decomposition but difficult to use to automate bottom up decomposition – i.e. consolidation of similar entities
- UML diagrams are difficult to create/maintain by many users in parallel (requirement in large-scale project)
- Need to teach end users on how to design in UML and to use UML tools

Domain specific semantic dictionary

- Easier to use with tools to enter and generate
  - Easier to use in bottom up decomposition
  - Possible to use tools such as Excel which end users already know how to use
- Possible to add entries in parallel by many developers and merge later

Semantic dictionary - WordNet

- Common dictionaries focuses on meaning of words, while WordNet focuses on *relationship* between sysnets

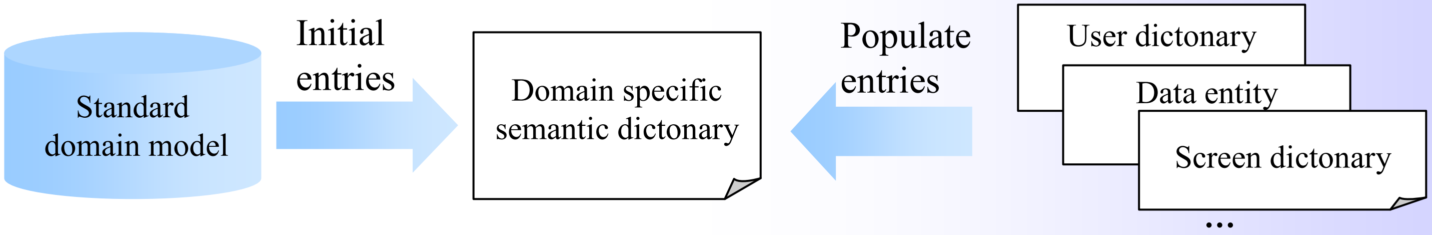
- Use concepts from WordNet to define *relationship between services* instead of focusing defining service functionalities

Use semantic dictionary to decompose message to semantic model and than to consolidate/decompose semantically equivalent / replaceable messages

Domain specific semantic dictionary

- General dictionary such as *WordNet* requires too much effort to create and maintain in a project
- Application development are usually domain specific

- Use domain specific semantic dictionary
  - Convert existing standard domain specific domain model (e.g. banking domain model)
  - Add words from existing dictionaries



Standard domain model

Initial entries

Domain specific semantic dictonary

Populate entries

User dictonary

Data entity

Screen dictonary

…

Relations

- Lexical relations
  - Synonymy/Antonymy : equivalence
- Semantic relations
  - Nouns (entries)
    - Hypernymy/Hyponymy : inheritance
    - Meronymy/Holonymy : aggregation/composition
  - Verbs (events)
    - Troponymy : inheritance
    - Entailment
      - Correlated : two or more events happens in parallel
      - Sequential : an event happens after another

        (pre-requirements and post requirements)

Consolidation/Decomposition

- Message model consolidation
  - Horizontal
    - Use lexical relations to consolidate semantic message models to canonical models composed from semantic primes
  - Vertical
    - Use semantic noun relations to adjust granularity of message models

- Neighboring message consolidation/decomposition
  - Use semantic verb relations to consolidate/decompose neighboring message models

Consolidation/Decomposition strategy

- Consolidate message model horizontally
- Identify key semantic primes
  - Concepts that have not been changed over long period (change by design/by deviation)
    - Comparison with past systems/Comparison with standard business model
  - Concepts that are anticipated to be changed but without clear implementation strategies (change by under specification)
- Consolidate/Decompose semantic model of messages to increase usage of key semantic primes
  1. Consolidate message model vertically
  2. Consolidate neighboring messages

Conclusion

- Designing by semantic model of messages after services are identified services by functional/nonfunctional requirements can enable processes to become more flexible
- Services related to change by specification usually impact processes more than changes by design or by deviation
- Domain specific semantic dictionary can be used to systematically decompose message models semantically and to consolidate/decompose messages

Thank you

Questions? Mail to: htshozawa@gmail.com