

Domain Engineering Process for Developing Multi-agent Systems Product Lines

INF 239

By

Vasumathi Gajji

What is Domain Engineering?

- *Domain Engineering is the activity of collecting, organizing, and storing past experience in building systems or parts of systems in a particular domain in the form of reusable assets (i.e. reusable work products), as well as providing an adequate means for reusing these assets (i.e. retrieval, qualification, dissemination, adaptation, assembly, etc.) when building new systems.*

Domain Engineering Components

Domain Engineering encompasses three components

- **Domain Analysis**
 - defining a set of reusable, configurable requirements for the systems in the domain
- **Domain Design**
 - Developing a common architecture for the system in the domain and devising a production plan
- **Domain Implementation**
 - Implementing the reusable assets, for example, reusable components, domain-specific languages, generators, a reuse infrastructure, and a production plan

Domain Analysis

- **Purpose:**
 - Select and define the domain of focus
 - Collect the relevant domain information and integrate it into a coherent domain model
- **Sources of domain information:**
 - Existing systems, domain experts, textbooks, experiments.....

Domain Model

- A domain model is an explicit representation of the common and the variable properties of the system in a domain, the semantics of the properties and domain concepts, and the dependencies between the variable properties

Domain Model Components

- **Domain definition:**
 - A domain definition defines the scope of a domain and characterizes its contents by giving examples of systems in a domain, counter examples and generic rules of inclusion or exclusion.
- **Domain lexicon:**
 - A domain lexicon defines the domain vocabulary.

Domain Model Components Cont.....

- **Concept models:**
 - Concept models describe the concepts in a domain expressed in some appropriate modelling formalism.
- **Feature models:**
 - Feature models define a set of reusable and configurable requirements for specifying the systems in a domain. Such requirements are generally referred to as features.

Domain Analysis Components

- **Domain planning:**
 - planning of the resources for performing domain analysis
- **Domain scoping:**
 - defining the scope of the domain
- **Domain modeling:**
 - developing the domain model
- **Domain identification:**
 - identifying the domain of interest

Domain planning and Scoping

Domain Analysis major process components	Domain Analysis activities
<i>Domain characterization</i> (domain planning and scoping)	<i>Select domain</i> Perform business analysis and risk analysis in order to determine which domain meets the business objectives of the organization.
	<i>Domain description</i> Define the boundary and the contents of the domain.
	<i>Data source identification</i> Identify the sources of domain knowledge.
	<i>Inventory preparation</i> Create inventory of data sources.

Domain Modeling

<i>Data collection</i> <i>(domain modeling)</i>	<i>Abstract recovery</i> Recover abstractions
	<i>Knowledge elicitation</i> Elicit knowledge from experts
	<i>Literature review</i>
	<i>Analysis of context and scenarios</i>

Domain Modeling Cont....

<p><i>Data analysis</i></p>	<p><i>Identification of entities, operations, and relationships</i></p>
<p><i>(domain modeling)</i></p>	<p><i>Modularization</i></p> <p>Use some appropriate modeling technique, e.g. object-oriented analysis or function and data decomposition. Identify design decisions.</p>
	<p><i>Analysis of similarity</i></p> <p>Analyze similarities between entities, activities, events, relationships, structures, etc.</p>
	<p><i>Analysis of variations</i></p> <p>Analyze variations between entities, activities, events, relationships, structures, etc.</p>
	<p><i>Analysis of combinations</i></p> <p>Analyze combinations suggesting typical structural or behavioral patterns.</p>
	<p><i>Trade-off analysis</i></p> <p>Analyze trade-offs that suggest possible decompositions of modules and architectures to satisfy incompatible sets of requirements found in the domain.</p>

Domain Modeling Cont....

<i>Taxonomic classification</i> <i>(domain modeling)</i>	<i>Clustering</i> Cluster descriptions.
	<i>Abstraction</i> Abstract descriptions.
	<i>Classification</i> Classify descriptions.
	<i>Generalization</i> Generalize descriptions.
	<i>Vocabulary construction</i>
<i>Evaluation</i>	Evaluate the domain model.

Domain Design

- **To develop an architecture for the family of systems in the domain**
- **Software architecture**
 - software architecture involves the description of elements from which systems are built, interactions among those elements, patterns that guide their composition, and constraints on these patterns. In general, a particular system is defined in terms of a collection of components and interactions among these components. Such a system may in turn be used as a (composite) element in a larger system design.

Architectural Patterns

- **Layers pattern:**
 - An arrangement into groups of subtasks in which each group of subtasks is at a particular level of abstraction.
- **Pipes and filters pattern:**
 - An arrangement that processes a stream of data, where a number of processing steps are encapsulated in filter components. Data is passed through pipes between adjacent filters, and the filters can be recombined to build related systems or system behaviour.

Architectural Patterns Cont....

- **Blackboard pattern:**
 - An arrangement where several specialized subsystems assemble their knowledge to build a partial or approximate solution to a problem for which no deterministic solution strategy is known
- **Broker pattern:**
 - An arrangement where decoupled components interact by remote service invocations. A broker component is responsible for coordinating communication and for transmitting results and exceptions.

Architectural Patterns Cont....

- **Model-view-controller pattern:**
 - A decomposition of an interactive system into three components:
 - A model containing the core functionality and data, one or more views
 - displaying information to the user, and one or more controllers that handle user input. A change-propagation mechanism ensures consistency between user interface and model.

Architectural Patterns Cont....

- **Microkernel pattern:**
 - An arrangement that separates a minimal functional core from extended functionality and customer-specific parts. The microkernel also serves as a socket for plugging in these extensions and coordinating their collaboration.

Domain Implementation

- **Domain Implementation**

- apply appropriate technologies to implement components, generators for automatic component assembly, reuse infrastructure (i.e. component retrieval, qualification, dissemination, etc.), and application production process

What is Domain?

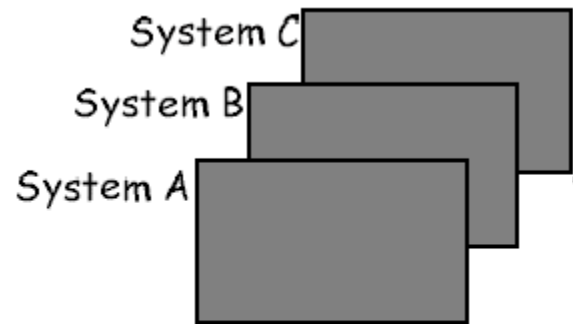
- **Domain as the “real world”**
 - An area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area
- **Domain as a set of systems**
 - knowledge “real world” + How to build software systems
- **Domain – An Area of Knowledge**
 - Includes a set of concepts and terminology understood by practitioners in that area
 - Includes the knowledge of how to build software systems in that area

Domain Scope

- **Horizontal scope**
 - How many different systems are in the domain?
- **Vertical scope**
 - Which parts of these systems are in the domain?
- **Encapsulated & Diffused domains**

Vertical Domains

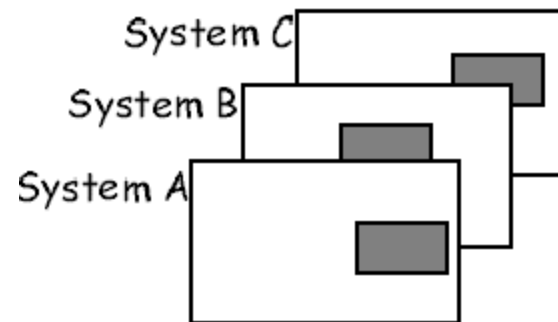
- Vertical domains contain complete systems



systems in the scope
of a vertical domain

Horizontal & Encapsulated Domains

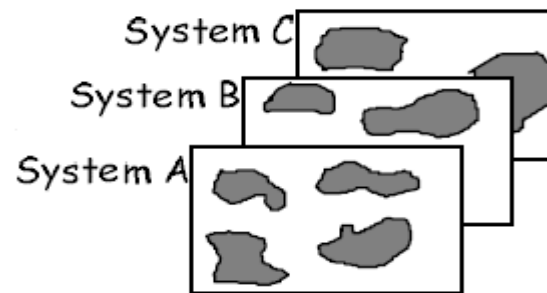
- Horizontal domains contain only parts of the systems in the domain scope
- Encapsulated domains are horizontal domains where the system parts in the domain are well-localized with respect to their



systems in the scope of
a horizontal,
encapsulated domain

Diffused Domains

- Diffused domains are also horizontal domains, but they contain several, different parts of each system in the domain scope



systems in the scope
of a horizontal,
diffused domain

Relationships Between Domains

- **Sub domains**
 - A is contained in B: All knowledge in domain A also belongs to domain B, i.e. A is a sub domain of B.
- **Support domains**
 - A uses B: Knowledge in A references knowledge in B in a significant way, i.e. it is worthwhile to represent aspects of A in terms of B. We say that B is a support domain of A.

Relationships Between Domains Cont....

- **Analogy domains**

- A is analogous to B : There is a considerable amount of similarity between A and B; however, it is not necessarily worthwhile to express one domain in terms of the other. We say that A is an analogy domain of B.

Features and Feature Model

- **Feature:**
 - a end-user-visible characteristic of a system
 - a distinguishable characteristic of a concept (e.g. system, component, etc.) that is relevant to some stakeholder of the concept

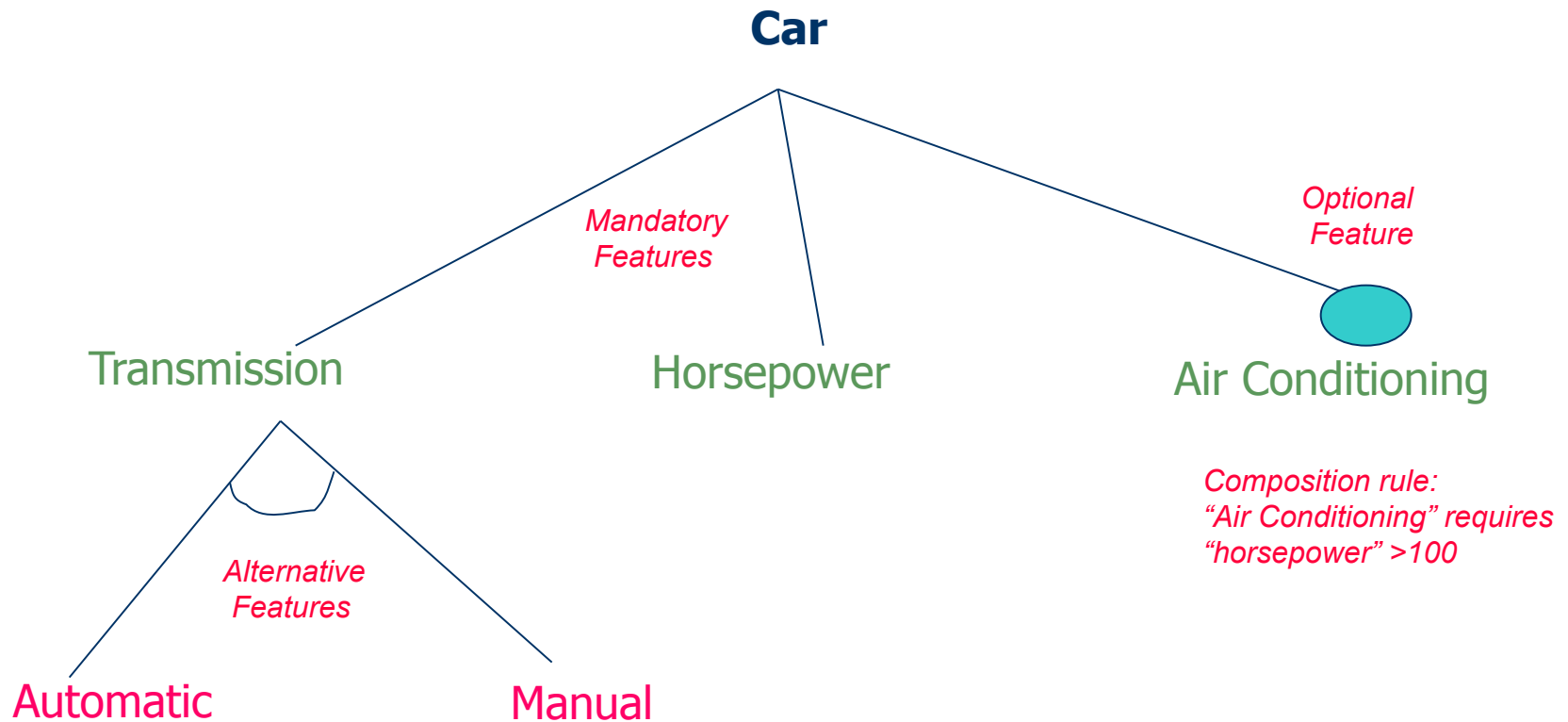
Feature-Oriented Domain Analysis (FODA)

- **FODA process consists of two phases**
 - **Context Analysis:** The purpose of Context Analysis is to define the boundaries of the domain to be analyzed
 - **Domain Modeling:** The purpose of Domain Modeling is to produce a domain model

Types of features

- **Mandatory features**
 - which each application in the domain must have, e.g. all cars have a transmission
- **Alternative features**
 - of which an application can possess only one at a time, e.g. manual or automatic transmission
- **Optional features**
 - which an application may or may not have, e.g. air conditioning

Feature Example



Rationale
"Manual" more
Fuel efficient

Feature Model Elements

- **features diagram**
 - a representation of a hierarchical decomposition of features including the indication whether or not each feature is mandatory, alternative, or optional
- **feature definitions**
 - for all features including the indication of whether each feature is bound at compile time, activation time, or at runtime
- **composition rules for features**
- **rationale for features indicating the trade-offs**

Organization Domain Modelling (ODM)

- **Plan Domain:**
 - This is the domain scoping and planning phase corresponding to Context Analysis
- **Model Domain:**
 - In this phase the domain model is produced. It corresponds to Domain Modelling
- **Engineer Asset Base:**
 - The main activities of this phase are to produce the architecture for the systems in the domain and to implement the reusable assets

Plan Domain (ODM)

- **Set objectives**
 - determine the stakeholders (i.e. any parties related to the project), e.g. end-users, customers, managers, third-party suppliers, domain experts, programmers, subcontractors
 - analyze stakeholders' objectives and project objectives
 - select stakeholders and objectives from the candidates
- **Scope domain**
 - scope the domain based on the objectives

Plan Domain (ODM) Cont....

- **Define domain**
 - define the domain boundary by giving examples of systems in the domain, counterexamples, as well as generic rules defining what is in the domain and what not
 - identify the main features of systems in the domain and the usage settings for the systems
 - analyze the relationships between the domain of focus and other domains

Model Domain (ODM)

- **Acquire domain information**
 - plan the domain information acquisition task
 - collect domain information from domain experts, by reverse-engineering existing systems, literature studies, prototyping, etc.
 - integrate the collected data, e.g. by pre-sorting the key domain terms, identifying the most important system features

Model Domain (ODM) Cont....

- **Describe domain**
 - develop a lexicon of domain terms
 - model the semantics of the key domain concepts
 - model the variability of concepts by identifying and representing their features
- **Refine domain**
 - integrate the models produced so far into an overall consistent model
 - model the rationale for variability, i.e. the trade-offs for using or not using certain features
 - improve the quality of features by clustering and experimenting with innovative feature combinations

Engineer Asset Base (OMD)

- **Scope asset base**
 - correlate identified features and customers
 - prioritize features and customers
 - based on the priorities, select the portion of the modeled functionality for implementation
- **Architect asset base**
 - determine external architecture constraints
 - determine internal architecture constraints
 - define asset base architecture based on these constraints

Engineer Asset Base (OMD) Cont...

- **Implement asset base**
 - plan asset base implementation
 - implement assets
 - implement infrastructure

Domain/Application Engineering

- **Domain engineering:**
 - Systematic approach to construct reusable assets in a given problem domain
- **Application engineering:**
 - Use the assets to build specialized software systems in the given domain

Domain/Application Engineering Cont....

- **Domain Engineering represents a valuable approach to software reuse and multi-system-scope engineering. Domain Engineering moves the focus from code reuse to reuse of analysis and design models.**

Software Engineering	Domain Engineering
<i>Requirements Analysis</i> ↪ requirements for one system	<i>Domain Analysis</i> ↪ reusable requirements for a class of systems
<i>System Design</i> ↪ design of one system	<i>Domain Design</i> ↪ reusable design for a class of systems
<i>System Implementation</i> ↪ implemented system	<i>Domain Implementation</i> ↪ reusable components, infrastructure, and production process

Software Development based on DE

