International Journal of Information Technology and Knowledge Management
July-December 2009, Volume 2, No. 2, pp. 347-352

COMPARISON OF DOMAIN ANALYSIS METHODS IN SOFTWARE REUSE

Aman Jatain & Shivani Goel

Domain analysis is recommended by many in the reuse research as a main process for achieving successful reuse. Domain analysis is accomplished by reengineering techniques and domain analysis methods. Domain analysis is the process of identifying, collecting, organizing and representing the relevant information in a domain based upon the study of existing systems and their development histories, knowledge captured from domain experts and emerging technology within a domain. This paper discusses some generally used domain analysis methods. After studying many domain analysis methods some criteria has been found based on which we compared domain analysis methods. Paper discusses the domain analysis methods that share the same objectives: analyzing the domain and developing domain models. However each technique defines a particular way of understanding the domain and capturing domain information as domain models. In general the process, the product and supporting tools, can characterize a domain analysis method. At the end we summarized methods according to their use in various domains.

Keywords: Domain, Domain Analysis, FODA, JODA, ODM, DADP, DSSA, Software Reuse, Domain Model, Commonalities, Variability's.

1. Introduction

Software development may not become an engineering discipline if it has not worked on a technology for developing products from reusable assets in regular manner, on an industrial scale. As discipline software reuse must define and promote the managerial, organizational, and technical standards that are required to achieve this goal [2]. One of the most important organizational decisions that having the greatest impact on reuse operations is whether the organizational structure is domain centered or application centered. Under domain-centered organization, domain-engineering team decides the development of reusable assets, leaving it to application engineering team to adjust their design discipline or activity to take the best advantage of available reusable assets. Under application-centered organization, the application engineering team delegates' development tasks to the domain engineering team, to serve the goals of their application development activity. In domain engineering, domain analysis, or product line analysis, is the activity that describes the commonalities and variability within a domain [1]. The term was given in 1980s by James Neighbors. Domain analysis is the first phase of domain engineering. It is a key method for realizing systematic software reuse. The key to reusable software in domain analysis is that it focuses the reusability of analysis and design, not code. In simpler words, domain analysis is same as systems analysis but instead of being applied on a single system it is done for multiple related systems.

* Computer Science and Engineering Department, Thapar University, Patiala. *E-mail: Jatain.aman8184@gmail.com*, *E-mail: shivani@thapar.edu*

This paper discusses different Domain Analysis (DA) methods, which are used in formal reuse to identify, organize and model knowledge about the solution of a domain to support its reuse among all elements in the domain.

Table 1.1
Domain Analysis Definitions

Domain Analysis Definitions	Identify object and operations of a class of system.
	System analysis for a set of systems.
	Characterize a software domain to support reuse.
	Identify, organize and model information to produce software requirements.
	Process in which a reusable software architecture and reusable code are defined.
	Identify domains in which reuse of certain experiences is effective.
	Determine characteristics that satisfy the optimum domain.

2. Domain

A domain is an area of knowledge or activity characterized by a family of related systems. The common managed features that specify specific market or mission can also define a domain [4].

The word "domain" can be used in several areas:

- Business area
- Problem domain
- Solution domain
- Area of knowledge with common vocabulary.

Domains can be of two types:

2.1 Horizontal Domains: Parts of a software system are categorized according to their functionality.

2.2 Vertical Domains: Software systems are categorized according to the business area.

3. DOMAIN ANALYSIS

The term "Domain Analysis "was given in 1980s by James Neighbors. It is a process by which w are able to exploit commonalities in applications in the domain, capture experiences, and identify variability's.

Domain Analysis is the foundation for reusability [5]. An important improvement of the reuse process happens when we succeed in deriving common architectures, generic models or specialized languages by using domain analysis that helps the software development process in a specific problem area.

A three-phase approach is suggested for the domain analysis [6]:

- Model the domain
- Architect the domain
- Develop software component assets

There are three main phases in domain analysis stage as shown in Figure 3.1.

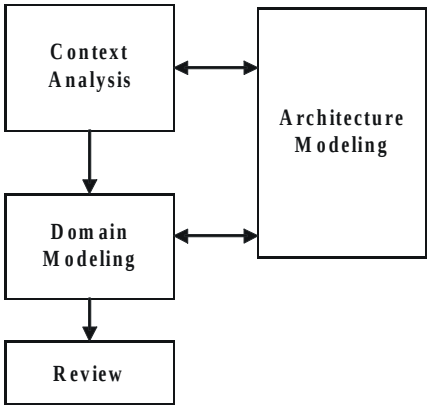

Figure 3.1: Domain Analysis Phases

- **Context Analysis:** The domain analyst discusses with domain expert and users to set the constraints of the domain and set a proper scope for the domain. The analyst also collects information for performing the analysis.
- **Domain Modeling:** The domain analyst makes use of information collected and other products of context analysis phase to create a domain model. The domain expert, system user and requirements analyst then review the model.
- **Architecture Modeling:** By using domain model the domain analyst makes the architecture model and then this model is reviewed by domain expert, requirements and the software engineer, user need not participate in this review.

Domain analysis can be seen as an extension of the conventional requirements analysis [12]. The primary advantage of domain analysis is that it provides flexibility. However it suffers some barriers due to the complexity and huge amount of time required for development. Domain analysis is suited for suitable, mature, and well-understood domains [15].

4. DOMAIN ANALYSIS METHODOLOGIES

There are number of domain analysis methods but nobody has ever tried to categorize them. Although some authors have compared different methods according to different criteria.

A domain analysis method should have two things:

- A domain theory, along with an anatomy of this theory both which will appear in domain model.
- A process, which, if strictly held, will allow construction of the generic domain model.

4.1 Purpose of Domain Analysis Methods:

- Improve reusable elements
- Populate libraries of reusable elements
- Integrate DA into the software process
- Decrease adaptation costs
- Construct reusable elements

Below is the overview of mainly used domain analysis methodologies:

4.2 Feature Oriented Domain Analysis Method:

The feature-oriented domain analysis method (FODA) was developed at the Software Engineering Institute. FODA emphasize on identifying features that characterize a domain and hence gives the approach its name. Application in a domain provides several capabilities [7]. These capabilities are modeled in FODA as features. To model these features, FODA defines a process for domain analysis that is based on three activities:

4.2.1 Context Analysis:

The context analysis activity defines the bounds of the domain with objective of scoping the domain under investigation [8]. Identifying the relationship between the application in the domain and the elements that are external

to the domain does scoping. Variability's are identified in terms of different data requirements and operation requirements imposed by the external environmental environment. Context analysis is used to define the inputs and outputs to and from the domain. The results of context analysis are usually context diagram and structure diagrams. Context diagrams relate the domain to the environment, while structure diagrams relate the domain to other domains, which could be part of the original domain.

4.2.2 Domain Modeling:

The domain modeling activity identifies commonalties and variability's that characterize applications within the domain, by modeling the functions, data and relationship between applications in the domain. The resulting models usually define what the applications are, what they do, and how they work. Domain modeling composed of feature analysis, feature analysis, and operational analysis, which produce features, information, and operational models, respectively. A feature model describes what the applications do in terms of operations. An informational model describes the applications in terms entities and their relationship. An operational model relates the information model and features model to the behaviors and function of applications.

4.2.3 Architecture Modeling:

Architecture modeling defines a framework for constructing applications in the domain. It identifies concurrent processes and common assets. It also allocates features, function and data objects to the processes and assets. As a result, the architecture model is developed, which defines the basic partitioning and interconnections necessary for constructing applications in the domain.

4.3 Joint Object Oriented Domain Analysis (JODA):

The joint object oriented domain analysis method advocates the idea that software objects are more understandable and customizable than traditional functions and subroutines. The Joint Integrated Avionics Working Group (JIAWG) reuse subcommittee developed JODA [9].

JODA is the domain analysis part of the reuse based software development approaches defined by JIAWG.

JODA consists of three phases as shown in figure:

4.3.1 Preparing the Domain:

It concerned with collecting information about the domain under consideration either by interviewing domain experts or by reengineering existing system. It also includes investigating the stability and maturity of technologies in the domain and their anticipated future.

4.3.2 Domain Scoping:

It includes definition of the domain glossary, services, dependencies, and the development of high whole part, subject, and inheritance diagrams.

4.3.3 Modeling the Domain:

The last phase in JODA in modeling the domain by defining the object life histories and state event response, investigating operation scenarios, and packaging and grouping reusable objects. The domain analysis activities within JODA are iterative [13].

4.4 Organizational Domain Modeling (ODM):

Organizational domain modeling prescribes a general method for conducting domain analysis in an organization. ODM offers a domain analysis method that is part of a larger domain engineering lifecycle [14]. Although ODM mainly focuses on organizational issues and transition to reuse discipline, it defines domain engineering technical activities.

ODM consists of three processes:

4.4.1 Domain Analysis:

The domain analysis approach focuses primarily on explicit descriptive and prescriptive analysis phases. "An explicit distinction is made between domain modeling activities that are descriptive, 'as is,' and prescriptive,' to be.' This distinction aims to prevent a modeler from unconsciously modeling aspects of legacy systems in terms of how they should be designed rather than how they are actually designed".

4.4.2 Architecture Modeling:

A descriptive domain model is developed from legacy systems, artifacts, and past experiences. This descriptive model is transform into a prescriptive domain model that documents the features that the domain architecture will support. Domain architecture is then created and represented in a concrete and analyzable format.

4.4.3 Asset Implementation:

Finally domain asset are developed that conform to the architecture.

4.5 Domain Analysis and Design Process:

Defense Information System Agency (DISA) develops the domain analysis and design process. The DADP method takes a problem or solution space approach [4]. The analysis aspect of DADP is concerned with identifying and defining problems within a group of related system in the domain. The design aspect of DADP is concerned with the

development of domain-specific solutions in terms of architecture and reusable assets.

DADP consist of four phases:

4.5.1 Identifying the Domain:

The outcome of the step is a set of business models, definition of system capabilities, domain models, and definitions of external interfaces, knowledge reuse opportunities, groups of systems sharing sane capabilities, and some descriptions and documentation of current systems and anticipated systems.

4.5.2 Scoping the Domain:

The domain is scoped with more than three systems in mind. In this phase, the domain analyst also identifies opportunities for reuse among systems in the same domain and reuse across other domains. The domain knowledge and team experiences are also documented.

4.5.3 Analyzing the Domain:

This step involves analyzing the problem space information. Commonalities and common objects adaptation requirements are identified. Domain models are constructed and verified.

4.5.4 Designing the Domain:

At this step the solution space is addressed by providing solution in terms of common designs and implementations of domain objects.

4.6 Domain Specific Software Architecture (DSSA):

The DSSA domain models are developed by DARPA for command and control applications [11]. DSSA domain analysis method is more concerned with defining the models to be produced rather than the analysis process. These domain models are function, dynamic and object model.

There are five stages in domain specific software architecture [10]:

4.6.1 Define Scope of the Domain

Define what can be accomplished-emphasis is on user needs.

4.6.2 Refine Domain Specific Requirements

Similar to requirement analysis-emphasis is on problem space.

4.6.3 Refine Domain Specific Design Implement

Similar to requirement analysis-emphasis is on solution space.

4.6.4 Develop Domain Models

Similar to high-level design-emphasis is on model interfaces and semantics.

4.6.5 Gather Reusable Work Products

Collection of reusable artifacts e.g. code and documentation.

4.7 Common Process followed by All Domain Analysis Methods:

All these Domain analysis methods at first glance seems to follow a different process for obtaining the domain model, but all the domain analysis methods follow some common processes as shown in Figure below.

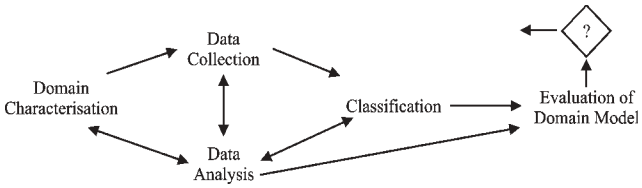

Figure 4.7.1: Representation of Common Process Followed by All Domain Analysis Methods

5. COMPARISON OF DOMAIN ANALYSIS METHODS

Table 5.1 below describes the use of domain analysis methods where they can be applied. These are not the only areas where a particular method can be applied. Research on domain analysis in recent years has produced many approaches.

Table 5.1
Comparison Table for Domain Analysis Methods based on their Use

	Area Where Domain Analysis Methods can be Applied	*Types of Methods Used*	*Domain*
1.	Family-oriented software engineering.	FODA	Functionality
2.	Object Oriented Software engineering	JODA	Functionality
3.	Diverse organizations and implementation technologies.	ODM	Functionality
4.	Domains based on Commonalities and Differences	DSSA	Functionality
5.	Opportunistic Domain engineering Approaches	DADP	Functionality

In Table 5.2 shown below comparison of domain analysis method is given based on their base model, Analysis steps and Main product produced and reuse methods.

Table 5.2
Comparison Table for Domain Analysis Methods

Base Model	*FODA* *ER*	*DSSA* *ER +OO*	*JODA* *OO*	*ODM* *OO*	*DADP* *OO*
Analysis Steps	1. Context Analysis 2. Domain Modeling 3. Architecture Modeling	1. Define scope of the Domain. 2. Refine Domain specific Requirements 3. Refine domain Specific Design Implement 4. Develop domain Models 5. Gather Reusable Work Products	1. Prepare Domain 2. Define Domain 3. Model Domain	1. Define the Domain 2. Acquire Domain Information 3. Develop Descriptive Model 4. Refine Domain Models 5. Scope the Asset Base 6. Architect the Asset Base 7. Implement the Asset Base	1. Identifying the Domain 2. Scoping the Domain 3. Analyzing the Domain 4. Designing the Domain
Main Product	1. Domain Model 2. Feature Model	1. Domain Model 2. Domain Specific Software Architecture 3. Domain Reusable Component	1. Domain 2. Domain Model	1. Descriptive 2. Prescriptive Domain Model 3. Domain Architecture	1. Business Model Definition Domain Model 2. Domain Models
Reuse Methods	1. Feature Model 2. Reuse library	1. Reusable Components 2. Active Repository	1. Domain Models 2. Design Reuse	1. Asset reuse Architecture	1. Reusable

6. CONCLUSION

In this paper, we have tried to summarize the common basic analysis steps, main product and various reuse methods used in various domain analysis methods. The common steps followed in all domain analysis methods are domain characterization, data collection, data analysis and classification. Some of areas where these methods can be applied are family oriented software engineering, object oriented engineering, diverse organizations and implementation technologies, domains based on commonalities and differences. The most important conclusion we have drawn is that there is no single "best" domain analysis approach for all types of product line software development. An organization should choose the one that best suits their software process needs, existing software base, and business objectives.

7. REFERENCES

[1] Domain Analysis and Software Systems Modeling by Ubén Prieto-Díaz, Guillermo Arango.

[2] Diary of a Domain Analyst: A Domain Analysis Case-Study from Avionics by. Kelly, W. Lam and B. R. Whittle Rolls-Royce University Technology Centre University of York, YO1 5DD, UK.

[3] An Evaluation of Domain Analysis Methods by X. Ferré S. Vegas Facultad de Informátic Universidad Politécnica de Madrid Campus de Montegancedo s/n, 28660 Madrid.

[4] Reuse Based Software Engineering: By Hafed Mili, Ali Mili, Edward Addy.

[5] Domain Analysis: An Introuction By Rubén Prieto-Díaz The Contel Technology Center.

[6] Domain Analysis: A System Approach to Software Reuse By Edward R. Comer.

[7] Feature-Oriented Domain Analysis (FODA) Feasibility Study By: Kyo C. Kang, Sholom G. Cohen, James A. Hess, William E. Novak, A. Spencer Peterson.

[8] Integrating 001 Tool Support into the Feature-Oriented Domain Analysis Methodology By Robert W. Krut, Jr.

[9] Joint Integrated Avionics Working Group (JIAWG) Object-Oriented Object Oriented Domain Analysis Version 3.1 Robert Holibaugh.

[10] CS 578 Software Architecture–Domain-Specific Software Architecture (DSSA).

[11] Domain-Specific Architecture for Software Agents T. Chithralekha, Reader, Department of Banking Technology, Pondicherry University, India.

[12] Software Reuse (A Standard-Based Guide): By Carma McClure.

[13] Object Oriented System Development By Deniss the Chempiox.

[14] Organization Domain Modeling Domain Modeling Life Cycle Mark A. Simos-Organon Motives, 36 Warwick Road, Watertown MA 02171.

[15] Framework Extraction with Domain Analysis By Department of Electrical and Computer Engineering–The University of Alberta ƒ Thera S.p.A., Brescia, Italy £ Coclea, Rovereto, Italy.

[16] Diary of a Domain Analyst: A Domain Analysis Case-Study from Avionics T. P. Kelly, W. Lam and B. R. Whittle Rolls-Royce University Technology Centre University of York, YO1 5DD, UK.