International Journal of Engineering, Science and Metallurgy

(General article)

A Comparative Study of Requirement Engineering Approaches

Sunita Allen[a], Deepak Dudeja[a]

[a] Assistant Professor, Department of CSE, Technology Education & Research Integrated Institutions

Received 11 Nov. 2011; accepted 15 Nov.2011, Available online 1 Dec. 2011

Abstract

A product line is a set of group of products that have a majority of features in common and vary only in certain specific features. Developing a group of common products that have a majority of features in common reports a great deal of reuse in all the phase of system development, requirement engineering is much more important for a product line practice. These requirements engineering approaches for software product lines are classified in two categories as follows

Keywords: *Aspect and Non Aspect oriented study*

1. Non Aspect-Oriented RE approaches

a) Feature-Oriented Approaches
b) Use Case-Base Approaches
c) Use Case and Feature-Based Approaches
d) Orthogonal Variability Approaches
e) Viewpoint-Based Approaches
f) Goal-Based Approaches

1.1 Feature-Oriented Approaches

Feature Oriented Domain Analysis (FODA) is a Domain Analysis method focused on the description of variability and commonalities by mean of features. A feature is a prominent and user-visible aspect, quality, or characteristics of a software system or systems. A number of approaches are based on FODA. The table 1 summarizes how this approach satisfies above-mentioned evaluation criteria.

1.2 Use Case-Based Approaches

Use Case-Based Approaches are further classified into two categories: John/Muthing and DREAM.

John/Muthing approach

This method is selected in situation where user-level information is essential for the domain model, where variability should be expresses early and explicitly, and where information is needed.

*Corresponding author
Email: sunita.cse.teri@terii.in

DREAM Approach

DREAM defines a domain requirements development process with artifacts that can serve as a core asset of domain requirements in a product line. A central concept in DREAM is that of a primitive requirement (PR). A PR is a transaction that has an effect on an external actor. Its granularity is in between that of a use case and an atomic operation of a use case; to discover the variability and rationale of the domain requirements more concrete and to discover the variability and rationale of the domain requirements.

1.3 Use Case and Feature-Based Approaches

A number of approaches combine use cases and features in order to describe requirements for product lines. In general, such approaches share the view that the use case model and the feature model have distinct but complementary goals: the use case model is user oriented, whereas the feature model is reuse oriented. The former details functionality, whereas the latter expresses configurability.

1.4 Orthogonal Variability Model (OVM) approach

Orthogonal Variability Model represents variability apart from requirement artifacts, differentiates between variation points, variants, and constraints among these entities, and explicitly defined the variability of the p-product line. OVM is inspired by the meta-model described by Bachmann et al. and by the notion of product population [1].

 ISSN 2249 - 7366

Table 1 Evaluation:-Feature-Oriented approach

Evolvability	Not Supported
Verification	Not supported in FODE, but Benavides et al. and Alves et al. provide support for checking properties of the feature model.
Trade-off-analysis	Partially supported with feature rationale in FODA and with CSP in Benavides et al. approach.
Scalability	Partially supported since FODA lacks modular mechanisms for grouping features.
Traceability	Partially supported in FODA with vertical traceability. FORM adds horizontal traceability.
Conceptual Modeling	Supported with context diagram during information analysis of domain modeling phase.
Domain modeling	Supported with structure diagram.
Domain scoping	Supported in the context analysis phase
Requirements modeling	Not Supported
Commonality and variability modeling	Supported in feature analysis step of domain context phase.
Aspect scoping	Supported only in FORM
Validation	Supported by comparing feature mode instances to original application.

Table 2 Evaluation:-John/Muthig approaches

Evolvability	Not Supported
Verification	Not supported
Trade-off-analysis	Not supported
Scalability	Partially supported, but limited due to relationships such as extends and includes, which makes it difficult to visualize the configurability of large diagrams.
Conceptual Modeling	Not supported
Requirements modeling	Supported with use cases
Commonality and variability modeling	Supported in diagrams with variant use cases and actors, and by using <variant> tags in use case textual descriptions.
Domain modeling	Not supported
Domain scoping	Not supported
Aspect scoping	Supported by separating core use cases from application specific use cases.
Validation	Not supported

Table 3 Evaluation:-DREAM approaches

Evolvability	Not Supported
Verification	Not supported
Trade-off-analysis	Not supported
Scalability	Partially supported with context generalization in PR-context matrix.
Traceability	Not supported
Conceptual Modeling	Supported with domain terminology
Requirements modeling	Supported with PRs and use cases
Commonality and variability modeling	Supported with analysis of PR-Context and PR-use case matrices.
Domain modeling	Not supported
Domain scoping	Supported with domain terminology
Aspect scoping	Supported by separating core use cases from application specific use cases.
Validation	Not supported

Table 4 Evaluation- Use case and feature based approaches

Evolvability	Supported only in PLUS with features basis impact analysis and in PLUSS with change case impact analysis.
Verification	Not supported
Trade-off-analysis	Not supported
Scalability	Supported, but limited in PLUS because configurability also describe in use cases.
Traceability	Both horizontal and vertical in FeatRSEB, but only vertical in PLUS and PLUSS.
Conceptual Modeling	Not Supported
Requirements modeling	Supported with use cases
Commonality and variability modeling	Supported with feature molding. PLUS also supports it in use cases.
Domain modeling	Not supported
Domain scoping	Not Supported
Aspect scoping	Supported only in FeatRSEB
Validation	Not supported

Table 5 Evaluation: Orthogonal Variability Model approach

Evolvability	Supported
Verification	Not supported
Trade-off-analysis	Supported by impact analysis on architecture offering cost and alternatives to stakeholders.
Scalability	Partially Supported
Traceability	Both horizontal and vertical are inherently supported.
Conceptual Modeling	Not Supported
Requirements modeling	Supported by specified relationship.
Commonality and variability modeling	Supported by analysis of the application-requirement matrix.
Domain modeling	Not supported
Domain scoping	Supported during commonality and variability analysis
Aspect scoping	Supported during commonality and variability analysis
Validation	Not supported

Table 6 Evaluation: Viewpoint Based approaches

Evolvability	Not Supported
Verification	Not supported
Trade-off-analysis	Not supported
Scalability	Partially Supported
Traceability	Support vertical traceability only, by links connecting reusable requirements within viewpoints to requirements to specific products.
Conceptual Modeling	Supported by domain dictionary built during characterize the domain activity
Requirements modeling	Supported with viewpoints
Commonality and variability modeling	Supported during the analyze document viewpoint activity.
Domain modeling	Not supported
Domain scoping	Supported by definition of viewpoints during the scope domain activity
Aspect scoping	Supported during analyze document viewpoint activity
Validation	Not supported

Table 7 Evaluation:-Goal Based approaches

Evolvability	Not Supported
Verification	Not supported
Trade-off-analysis	Not supported
Scalability	Partially Supported
Traceability	Not Supported
Conceptual Modeling	Supported
Requirements modeling	Supported
Commonality and variability modeling	Supported
Domain modeling	Supported
Domain scoping	Supported
Aspect scoping	Supported
Validation	Supported

Table 8 Evaluation-RDL approach

Evolvability	Supported
Verification	Partially Supported with point cut tests
Trade-off-analysis	Partially supported by solving temporal constraint
Scalability	Supported with tool
Traceability	Only horizontal traceability is supported
Conceptual Modeling	Not Supported
Requirements modeling	Supported
Commonality and variability modeling	Partially Supported with composition mechanism
Domain modeling	Not Supported
Domain scoping	Not Supported
Aspect scoping	Not Supported
Validation	Not Supported

Accordingly, it defines variability and constraints in the context of a product line or a set of product line: a variant may be mandatory in one product line, but may be optional in another.

1.5 View Point-Based Approaches

The viewpoint-oriented Domain Requirement Definition method is an iterative method comprised of the following activities: scope the domain, characterize the domain, document the viewpoints, and analyze document viewpoints [2].

1.6 Goal-Based Approaches

Table 9 Evaluation-NAPLES/EA-Miner approach

Evolvability	Not Supported
Verification	Not Supported
Trade-off-analysis	Partially supported by solving temporal constraint
Scalability	Supported with tool
Traceability	Only horizontal traceability is supported
Conceptual Modeling	Not Supported
Requirements modeling	Supported
Commonality and variability modeling	Partially Supported with composition mechanism
Domain modeling	Not Supported
Domain scoping	Not Supported
Aspect scoping	Not Supported
Validation	Not Supported

The Goal-based approaches focus on modeling variability from the perspective of the problem domain. They effectively capture the alternative ways by which the stakeholders achieve their goals, where top-level goals represent very high level functional and non-functional requirements. These alternatives ways are represented and refined with the AND/OR decompositions of the goal graphs.

2. Evaluation of Aspect-Oriented RE Approaches

Requirement Engineering is primarily concerned with information collection and structuring tasks, which can be categorized into requirements gathering and requirement analysis. Aspect-Oriented Requirement Engineering (AORE) has emerged as a new way to modularize and reason about crosscutting concerns during requirements engineering. AORE extends the notion of separation of concerns in RE with that of requirements level aspects. Such aspects modularize requirements that affect and constraint other requirements. Requirements pertaining to these concerns are often scattered [3] in the statements of other requirements. This approach can be further classified in two approaches as:

Table 10 Comparison of RE approaches for SPL

Approaches	FOD	FEA	PLU SS	PLU S	JOH N	DRE AM	OV M	VIE WPO	GOA L	RDL	NAP LES/ EA- MIN
Evolvability	N	N	F	F	N	N	F	N	N	F	N
Verification	P	N	N	N	N	N	N	N	N	P	N
Trade-off-analysis	P	N	N	N	N	N	F	N	N	P	N
Scalability	P	F	F	P	P	P	P	P	P	F	F
Traceability	P	F	P	P	P	N	F	P	N	P	P
Conceptual Modeling	F	N	N	N	N	F	N	F	F	N	N
Requirements modeling	N	F	F	F	F	F	F	F	F	F	F
Commonality and variability modeling	F	F	F	F	F	F	F	F	F	P	F
Domain modeling	F	N	N	N	N	N	N	N	F	N	N
Domain scoping	F	N	N	N	N	F	F	F	F	N	N
Aspect scoping	N	F	N	N	F	F	F	F	F	N	F
Validation	F	N	N	N	N	N	N	N	P	N	N

Requirements Description Language (RDL) and NAPLES/EA-Miner.

2.1 Requirement Description Language (RDL)

This approach uses the richness of the natural language. This approach supports expensive semantics-based within a requirement description language. This utilizes the fact that the natural language itself has a clearly defined set of syntactic rules and semantic elements. RDL denotes the syntactic elements of the natural language and exploits the fact that each syntactic element has a designated semantic rule. These are semantic rules from the basis of expressive point cut expression in RDL [4].

2.2 NAPLES/EA-Miner

The NAPLES approach addresses SPL engineering throughout the lifecycle by using different techniques, e.g., natural language processing and aspect-oriented software development (AOSD), to provide automated support and separation of concerns during the SPL lifecycle.

3.1 Comparison of Requirement Engineering Approaches

Each approach has been described and evaluated against some criteria. Here an analysis is presented for each criterion by showing the finding regarding our comparison study of requirements engineering approaches for software product lines [5]. The comparison of RE approaches for SPL is shown in table 10. In this table, the following symbol has their meaning as

N-Not supported
P-Partially supported
F-Fully supported

Conclusion

This comparative study has shown that requirements engineering approaches offer a lot of challenges to be addressed, such as: 1) the adoption of different requirements and variation models without clear relationship between them; ii) lack of support for trade-off analysis; and iii) the difficulty to reason about maintenance and evolution scenarios. Most of the SPL approaches do not define a coherent and clear set of requirements and variation model and respective relationships between them.

Early Aspect approach of AO-approaches can help to deal with crosscutting concerns encountered at the requirements levels. Early Aspects approaches, presented in this thesis, have demonstrated their potential to improve the traceability between requirements.

Aspects modeled at the requirements level can be mapped directly to components, aspects or architectural decisions at subsequent stages. Trade-off analysis is another task; which can be improved by the use of AO approaches. Early aspect approaches address the separated modeling of different system concerns and emphasize the identification of their relationships. The explicit modeling of the concerns and their relationships can facilitate the analysis and identification of possible conflicts and inconsistencies between them.

References

1. João Araújo, Jon Whittle, Dae-Kyoo Kim (2004), Modeling and Composing Scenario-Based Requirements with Aspects, *Proceedings of the 12th IEEE International Requirements Engineering Conference* (RE'04).
2. Lirong Dai (2009), SSIRI 2009 Short Paper Security Variability Design and Analysis in an Aspect Oriented Software Architecture, *Third IEEE International Conference on Secure Software Integration and Reliability Improvement*.
3. T.Elrad, S.Clarke, and M.Aksit (2004) , Aspect- Oriented Software Development, *Addison-Wesley*.
4. Lidia Fuentes ,Nadia Gámez ,Pablo Sánchez (2009), Aspect-oriented design and implementation of context-aware pervasive applications, *Innovations Syst Softw Engg.*, 5, pp.79–93
5. Zave, P. (1997), Classification of Research Efforts in Requirements Engineering, *ACM Computing Surveys*, 29(4), pp.315-321.