Feature-Oriented Nonfunctional Requirements Analysis for Software Product Line*

Xin Peng[1], Seok-Won Lee[2], and Wen-Yun Zhao[1]

[1]School of Computer Science, Fudan University, Shanghai, 200433, China

[2]Knowledge-intensive Software Engineering Research Group, Department of Software and Information Systems, College of Computing and Informatics, University of North Carolina at Charlotte, Charlotte, NC 28223, USA

E-mail: pengxin@fudan.edu.cn; seoklee@uncc.edu; wyzhao@fudan.edu.cn

Abstract Domain analysis in software product line (SPL) development provides a basis for core assets design and implementation by a systematic and comprehensive commonality/variability analysis. In feature-oriented SPL methods, products of domain analysis are domain feature model and corresponding feature decision model to facilitate application-oriented customization. As in requirement analysis for a single system, domain analysis in SPL development should consider both functional and nonfunctional domain requirements. However, nonfunctional requirements (NFRs) are often neglected in existing domain analysis methods. In this paper, we propose a context-based method of NFR analysis for SPL development. In the method, NFRs are materialized by connecting nonfunctional goals with real-world context, thus NFR elicitation and variability analysis can be performed by context analysis for the whole domain with the assistance of NFR templates and NFR graphs. After variability analysis, our method integrates both the functional and nonfunctional perspectives by incorporating nonfunctional goals and operationalizations into the initial functional feature model. NFR-related constraints are also elicited and integrated. Finally, a decision model with both functional and nonfunctional perspectives is constructed to facilitate application-oriented feature model customization. The computer-aided grading system (CAGS) product line is employed throughout the paper to demonstrate the method.

Keywords software product line, nonfunctional requirement, domain analysis, feature-oriented method, variability analysis

1 Introduction

Software systems, aside from implementing all the desired functionality, must cope with nonfunctional aspects such as reliability, security, accuracy, safety, performance, etc [1]. Ineffectively dealing with nonfunctional requirements (NFRs) [1-4] is a common cause of software-intensive system failures and can badly affect the confidence on the system. It is the same

* This work is supported by the National Natural Science Foundation of China under Grant No. 60703092 and 90818009, the National High Technology Development 863 Program of China under Grant No. 2007AA01Z125.

as software product line (SPL) development and is even harder to properly deal with NFRs in SPL. The goal of SPL engineering is to support the systematic development of a set of similar software systems (called domain) by understanding and controlling their common and distinguishing characteristics [5]. Commonality in a specific domain provides the basis of proactive reuse in SPL, while differences among different application products (i.e. variations) demand proper variability mechanisms in the analysis and design model to enable application-oriented customization.

NFR analysis is a part of domain analysis in SPL development. Domain analysis is the requirements analysis activity for a specific domain to identify and document the commonalities and variations in related systems in the domain [5]. There have been many works focused on domain analysis and design in SPL (e.g. [5-9] and our previous work [10-11]). These works only concentrate on domain analysis and design for functional requirements and their variations, although the existence and influence of NFRs are also mentioned in some works (e.g. [5][7][12]). However, NFR-related variations also exist and have great influence on SPL design and implementation. For example, computer-aided grading system (CAGS) software for examinations at different levels and scales (e.g. regular in-school examinations, region-level unified examinations, and college entrance examinations, etc.) may require different levels of security, performance, privacy, etc. although their business functions are similar to each other. Furthermore, redundant NFR design and implementation usually mean much higher cost of ownership and maintenance, or influence on other more desired goals due to NFR conflicts. For example, it is unnecessary to include hard-certification-based authentication and encrypted storage/transfer in CAGS applications for in-school examinations, but they are essential for college entrance examinations, since the scores directly decide whether a student can be matriculated by universities. Therefore, nonfunctional variations should be identified and embodied in the domain analysis and design, so that each application in the targeted domain can have proper nonfunctional designs and achieve overall quality.

Functional variability is usually embodied in optional and alternative functionalities. Nonfunctional variability is different due to the characteristics of NFRs. Since NFRs can hardly ever be satisfied, they are more often treated as conceptual “soft” goals with the connotation of partial satisfaction [2]. On the other hand, NFRs are closely related to functional requirements and there exist conflicts and tradeoffs between different NFRs. Therefore, nonfunctional variations often exhibit different types and levels of sensitivity on nonfunctional properties and tradeoffs, e.g. normal and strong authentication. As for the influence on SPL design, NFRs in SPL first bring NFR-related operationalizations and constraints to the implementation of related functions similar to the NFRs for individual systems. Furthermore, nonfunctional variations will influence the commonality/variability ranges, which are crucial for variability-oriented design in

SPL architecture. From the aspect of application derivation, nonfunctional goals play different roles in leading the customization beyond functional perspectives.

NFRs have been frequently neglected or poorly considered in software design [1], both in traditional software development and in SPL. This situation is partly due to the vagueness of NFRs and complex dependencies among NFRs (e.g. positive or negative, partially or fully etc.) [1-4]. It is even more difficult in NFR analysis and design for SPL: not only should NFRs be identified and considered, but also NFR-related variations should be analyzed and properly treated in SPL design. Related issues include: 1) how to identify commonality/variability from SPL context; 2) how to integrate NFRs into the domain model; 3) how to provide different tradeoff choices for applications, etc.

In this paper, we propose a feature-based NFR analysis method for SPL development, supporting NFR elicitation, nonfunctional variability analysis, NFR integration, and decision modeling. The method takes a feature-centered viewpoint in NFR analysis, since feature-based methods provide good mechanisms for commonality/variability analysis and representation and have been widely adopted in domain analysis (e.g. [5-8][10]). On the other hand, the method introduces context model, which depicts real-world execution environment for functional features, as the anchor for NFR elicitation and variability analysis. In requirement engineering, the effect of real-world environment (or context) has been long realized (e.g. [13][14]). The environment is there as the reality, and the requirements, originating from the environments, are desired or optative conditions over the phenomena in the environment [13].

In the method, the initial functional feature model is assumed to be built beforehand and the method will first construct the domain context model. Second, nonfunctional goals related to each functional feature are identified from the context model by matching with NFR templates. Third, the NFR graph introduced in [2] is employed to refine nonfunctional goals into sub-goals at different levels, and identify operationalizations for NFR level and related dependencies (positive or negative). Fourth, nonfunctional variations are analyzed from the three levels of NFR goal, NFR level and operationalization according to feature contexts, NFR templates, NFR conflicts, and environmental dependencies. Then, NFRs are integrated into the feature model along with nonfunctional variations and constraints. Finally, the feature decision model with both functional and nonfunctional perspectives is constructed based on the analysis of NFR conflicts, dependencies and feature constraints. By incorporating both functional and nonfunctional requirements into domain feature model and decision model, we provide a comprehensive basis for application product customization. As a demonstrating product line, the CAGS product line is being employed throughout the paper to illustrate the method.

The remainder of this paper is organized as follows. Section 2 introduces the CAGS product line and analyzes the NFR-related problems

encountered in the domain, providing the background illustration for the issue of NFR analysis for SPL. Section 3 presents background introductions to feature model, nonfunctional goals and NFR graphs employed in our method. Section 4 gives the overview of the whole method, and the four method phases are introduced in section 5. Section 6 briefly evaluates our method and discusses related issues on NFR analysis for SPL development. Then section 7 introduces related work and compares them with our method. Finally, section 8 draws the conclusions and plans for our future works.

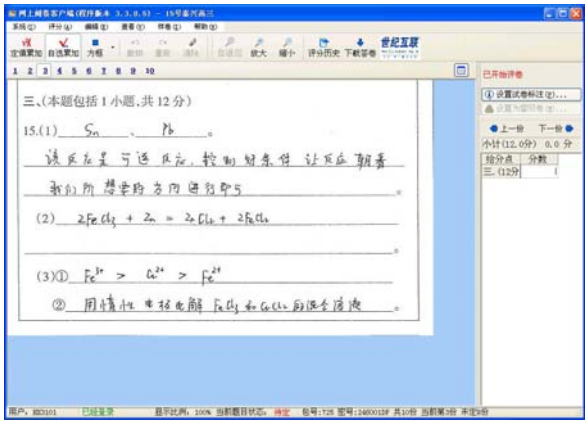
2 NFR Analysis for SPL: The Problems

2.1 The CAGS Product Line

CAGS [15] is a computer aided grading system for subjective examinations, in which an electronic answer is reviewed and graded by several graders (usually teachers) for fairness. By CAGS, electronic answers will first be collected and pretreated, and then be decomposed into data units, usually by questions. When grading, answer data units will be organized and dispatched to graders according to predefined rules and privilege settings [15]. After each grading, results (including scores and comments) are submitted automatically and following statistics and analysis can be performed. In some cases, additional checking and mediation may be performed to ensure the grading quality and fairness.

The CAGS software was first developed for spoken English examinations in Chinese colleges and achieved its initial success [15]. Later, another close examination mode, the traditional written examination, is also included in the consideration of the CAGS group, since these two kinds of grading systems are common in most aspects and only differ in the preparation and presentation of the electronic answers. Therefore, the initial CAGS software evolved to cover both spoken and written examinations (see Figure 1). On the other hand, the CAGS software is gradually applied to more examinations at different levels (e.g. middle schools, colleges, city and province) and different significances (including regular in-school examinations, and official college and senior middle-school entrance examinations). Ultimately, the CAGS software has evolved into the CAGS product line, including a complete series of electronic grading products for different examination modes, levels and significance.

(a) CAGS for spoken examination



(b) CAGS for written examination

Fig.1. Snapshots of CAGS application products

The initial functional feature model for the simplified CAGS product line is shown in Figure 2 (a brief introduction to the feature model can be found in section 3.1). From the feature model, we can see that "CAGS" includes five mandatory parts of "Login", "AnswerPrepare" (prepare electronic answers for grading), "AnsPackDispatch" (pack answers and dispatch to graders), "Grade" (examine and mark the assigned answer), "Query&Stat." (query and statistics), and an optional part of "Check&Mediation" (additional check and mediation for questionable answer grading). "AnswerCut" is another example of optional feature, which means an image answer may need to be cut into several parts in written answer preparation. Besides optionality, generalization is another mechanism of feature variability (*Alternative* and *OR* sub-features). For example, "ImgShow" and "AudioPlay" are two alternative sub-features of "AnsDisplay"; "subjectExam" (examination by subjects) and "syntheticExam" (examination with mixed subjects) are two OR sub-features of "writtenExam", which means, in the application for written examination, supported examination types include subject-oriented or synthetic or both (see the cardinality [1..2]). In Figure 2, another lightweight mechanism for feature generalization, facet (see section 3.1), is also employed, which denotes dimensions of precise description for functional features [10]. The value space of facets is called terms, which can also have their generalization structure (see underlined features in Figure 2). In Figure 2, "dispatchMode" is identified as a facet of "AnsPackDispatch", and can be "byPaper" or "byQuestion" in a certain CAGS application.

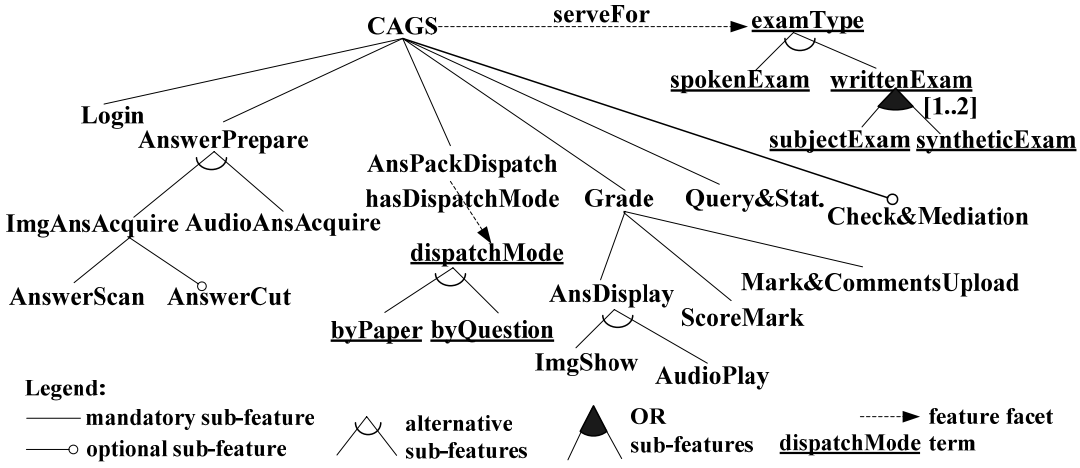


Fig.2. Initial functional feature model for the simplified CAGS product line

2.2 Nonfunctional Variations in CAGS

Due to the broad spectrum of the CAGS product line, a great deal of variations exist among different CAGS applications, so commonality/variability analysis must be performed to enable the design and implementation of the reuse platform for application derivation. In the beginning, the CAGS group concentrates on functional variations within the desired domain scope as shown in Figure 2, including different answer media (audios

or images), different distribution policies, etc. These variations have led to a series of variable design and implementation for different application products. Then, nonfunctional concerns, such as security, reliability and performance, etc, are gradually recognized. Furthermore, these concerns exhibit diversity in different applications.

Grading security is the primary nonfunctional concern in CAGS applications, including authentication, transfer security and storage security. However, it differs greatly in different applications. In applications for middle schools, the clients seldom mention the requirement of security, since the system is only used in their in-house examinations and users are all their teachers. On the other hand, in applications for official (college or senior middle-school) entrance examinations, grading security is emphasized greatly as the scores are significant for the students and the graders are from various organizations.

For spoken examinations, grading is performed by listening to the audio records, so speeding must be a user-friendly function in most of the time by enabling the grader to fleet over the record. However, in some official spoken examinations, speeding is not allowed, since it will bring up the issue of irresponsibility of graders. Therefore, by considering the reliability of grading, speeding should not be developed as a fixed function, but an optional one.

For written examinations, a primary concern on user experience and grading reliability is the definition of the answer images, which are generated by answer sheet scanning. High definition is always a desired feature for graders. However, the acquirement of high definition is highly tied to the resolution of the monitor used. Although monitor resolution is not a problem for most of the clients, there are still some small schools that have only old CRT monitors. For these clients, low-definition answer image is the only choice. Furthermore, high-definition answer images have a much larger demand on storage space and the price for the corresponding scanner is much higher, which can heavily influence on the choice of the client. Therefore, both high- and low- definition answer images should be supported by the CAGS product line to provide products suited for different clients.

2.3 The Problems

In SPL development, variability analysis is essential for both functional and nonfunctional requirements. Both functional and nonfunctional features are identified from the marketing plan [7]. If the SPL group decides to involve applications with different nonfunctional concerns in the SPL scope, nonfunctional variations will emerge. For example, most nonfunctional variations in CAGS SPL originate from the marketing plan that involves both high- and low-end examinations in the product line.

Specialization and optionality are the two basic variability mechanisms [8][10]. From section 2.2, we can see there are also nonfunctional variations in the CAGS domain.

- Grading Security. The clients are heavily concerned in formal official examinations,

but not so much concerned in other types of examinations. High security is always good, but it may bring additional cost and influence on the convenience of use. For example, in most of the common implementation for high security, each user must have a hard certification (usually a USB key) and take it when using the system. The additional costs and inconveniences are unacceptable in some applications for nonofficial examinations.

- Grading Reliability. It is always a big concern in all of the grading applications, but it differs in degree, e.g. high-reliability spoken examinations do not allow speeding when playing answer audios.
- User experience. High definition is always desired for its better user experience, but limitations on monitor devices and high equipment (the scanner) cost may bring up some other issues.

From these, we can see that there are nonfunctional variations in SPL development. Nonfunctional variations are not as obvious as functional ones. For example, an optional function can be definitely involved or removed in a product, but a NFR can never be said to be involved or removed but only high or low in the degree of their effectiveness. However, we can still consider NFRs from a realistic viewpoint, that is to say a NFR can be seen as not concerned if no special consideration is needed for it. For example, security can be seen as not concerned if simple password-based authentication and plain-text-based transfer/storage are acceptable. Besides optionality, NFRs can also be alternative in desired satisfaction levels or implementation ways. For example, high-level security demands hard-certification-based SSL transfer, while medium-level security demands only soft-certification-based SSL transfer (see the NFR graph in Figure 3).

In SPL design and implementation, separating and localizing variations are a primary respect to maximize the reusable parts while reserve necessary flexibilities for product diversity. The same as functional variations, nonfunctional variations (including optional and alternative NFRs) also have significant impacts on variability design and implementation. Nonfunctional variations can influence the commonality/variability boundaries in design and implementation. Optional NFRs can bring optional functions into the design and implementation, e.g. encrypt/decrypt for storage security. Alternative NFRs can bring variable functions, e.g. password-, soft-certification-, or hard-certification based authentication. Moreover, nonfunctional variations can also refine existing functions based on different constraints on them, e.g. the difference on user experience will bring a new variation of image definition into existing function of electronic answer preparation. These nonfunctional variations should be reserved in the domain model to meet different preferences and

different environmental constraints in different applications. For example, all of the password-based, soft- and hard-certification-based authentications should be supported by the SPL platform for possible uses. Therefore, nonfunctional variations must be identified, explicated and documented, just as what has been done for functional variations, and incorporated into the feature model to achieve a comprehensive domain model with both functional and nonfunctional perspectives.

3 Background

3.1 Feature Model and Feature Decision Model

Domain engineering and application engineering are the two essential phases in SPL development. Domain analysis, which is the domain-level requirements analysis activity for SPL development, constitutes domain engineering together with domain design and implementation. Domain analysis techniques can be used to identify and document the commonalities and variations in related systems in a domain [5].

In widely-used feature-oriented methods (e.g. [5-8][10-11]), product of domain analysis is feature model. Feature model describes the system-to-be as a combined set of feature variables [16]. It is the basis of domain design and implementation and can be tailored to produce the specification and design of application product [6]. A feature is a characteristic of software from user or customer views [6]. Feature model provides formal representations for features along with variations (*Optional*, *Alternative*, and *OR* features) and feature refinement structure according to decompositions and generalizations (see Figure 2). Optional features can be included or removed in application product. Alternative features mean that one and exact one of them should be bound for the parent feature in application product. OR features mean that any non-empty subset of them can be bound for the parent feature in application product. Besides, some methods also introduce another lightweight mechanism for feature generalization, called facet [10] (or characterization in [8]), which refines a feature by identifying its attribute features [8]. Facets can be construed as perspectives, viewpoints, or dimensions of precise description for features [10].

In SPL, variability is usually defined through variation points. A variation point defines a decision point together with its possible choices (functions or qualities), and the available functions or qualities for a variation point, i.e. available choices, are called variants [12]. The number of variants that can be chosen for a variation point is called cardinality [12]. In feature model, optional, alternative and OR features are variation points. For an optional feature, the variants are inclusion and removal of the feature. For an alternative or OR feature, the variants are its sub-features. In feature model, cardinality is only effective for OR features, since an OR feature can have different number of sub-features bound in different applications. Cardinality can be represented as [min..max], e.g. [1..2] for “writtenExam” in Figure 2.

Variations defined in domain engineering (whether functional or nonfunctional) are resolved in different phases of application engineering to

derive the application product. In order to help resolution decision in application development, decision model is proposed. A decision model structures the variability within a product line as a set of decisions to be resolved along with the interrelations between decisions [9]. With decision model, complexities of variability can be reduced greatly by providing a clear and natural decision path to follow. Therefore, after integrated into feature model, nonfunctional perspectives should also be incorporated into the feature decision model to provide full-scale variability resolution guidance. The incorporation can be implemented systematically by considering dependencies among NFR goals, functional features and environmental conditions.

3.2 Nonfunctional Goals and NFR Graph

There are fundamental differences between goals and features [16]. Goals represent stakeholders' intentions, which are manifestations of intent which may or may not be realized. Features in the product line represent system-to-be functions or properties. Goal model provides a natural way to identify variability at the early requirements phase by allowing the capture of alternative ways for stakeholders to achieve their goals [17]. NFRs can also be embodied in a goal model. Some NFRs might not be satisfied completely, and they are usually represented as soft goals [4]. These NFR-related goals and their variations should be further analyzed, refined and integrated into the feature model.

In our method, candidate NFRs are first captured in a goal model. A goal model consists of one or more root goals that represent stakeholder objectives, and each of them is AND/OR decomposed into sub-goals to form a forest [16]. We use NFR graph to represent the goal model for NFRs. The NFR graph provides refinement structure for various NFRs. In NFR graph, NFRs are viewed as goals (roots of an AND/OR graph) that are decomposed into sub-goals (sub-graphs) until all the necessary operationalizations (actions and information) are represented at the leaf-node levels of the graph [1].

NFR graphs in our method are adapted from NFR graphs used in [1] and [2] by adding mechanisms for variations in sensitivity of nonfunctional properties (NFR level). On the other hand, environmental dependencies are identified for each operationalization to help determine its applicability in the domain.

There are both general and domain-specific nonfunctional goals. For the former, corresponding NFR graphs can be reused in different domains. For the latter, NFR graphs should be constructed by considering domain-specific stakeholder concerns. NFR graphs for Security, Usability, Grading Reliability and Grading Usability in CAGS domain (segment) is depicted in Figure 3, in which "Security" and "Usability" are general goals while the other two are specific to the CAGS domain. In our NFR graph, there are four elements:

- **NFR goals** (including sub-goals) represent the refinement of abstract NFRs.
- **NFR levels** identified for each atomic NFR sub-goal denote the levels of typical sensitivity. An atomic NFR goal can have several levels or only one level (itself).

- **Operationalizations** are actions or suggested restrictions for the implementation of each NFR level. A NFR level can have multiple operationalizations as combined implementation policies.
- **Environmental conditions** specify necessary real-world conditions (e.g. devices) required by operationalizations.

In Figure 3, “Security” is decomposed into “Authentication”, “Transfer Security” and “Storage Security”. Three levels of Transfer Security are identified. They can be implemented by SSL transfer without client certification, with soft certification and hard certification, respectively. Grading Reliability and User experience are domain-specific NFR goals. In their decomposition structures, Electronic Answer Definition is identified as a common sub-goal of them, since high answer image definition contributes to both of them.

Similar to [1], static and dynamic operationalizations are distinguished, which are depicted by solid circle and dashed circle, respectively. Dynamic operationalizations are those that call for actions to be performed [1] (e.g. Audio Speeding), while static operationalizations express certain restrictions (e.g. the restrictions “No Additional Request for Use” and “High Image Resolution”).

In the NFR graph, there are four types of relationships between different elements: AND/OR decompositions between goal and its sub-goals; atomic goal and corresponding NFR levels; AND/OR decompositions between NFR level and its operationalizations; Environmental dependencies between operationalizations and environmental conditions. Similar to OR relationships between goals, OR decomposition between NFR level and its operationalizations represent alternative operationalizations, while AND decomposition represents all-included operationalizations. For example, dynamic “Grading Surveillance” and static “Must Finish the Whole Answer Record” in Figure 3 are identified as AND operationalizations for “Compelled Grading Control”. The former provides real-time grading surveillance of certain graders, while the latter restricts that each answer record must be played completely before the score is given.

In NFR graphs, conflicts between NFRs can be identified between different operationalizations. For example, we can find conflicts between “Transfer Security” and “Usability”. High transfer security has conflict with “Cost of Ownership” and “Convenience of Use”, since each user should get a hard certification (e.g. USB key) and take it when used. Another conflicting example is the restriction “Must Finish the Whole Answer Record” and “Audio Speeding”. It should be emphasized that we can only identify non-trivial conflicts and ignore others. For example, storage-performance-related conflicts are ignored here, since sufficient storage space can be seen as always available in CAGS domain due to great development of storage devices.

Environmental dependencies are identified to evaluate the applicability of each operationalization. In Figure 3, “USB Key” and

"High-Resolution Monitor" are identified as environmental dependencies of "Transfer with Hard-Cert.-SSL" and "High Image Resolution", respectively. These dependencies will be evaluated in NFR variability analysis.

In NFR graphs, original goal variations exist at different levels. An OR decomposition of a goal introduces a variation point, which defines alternative ways of fulfilling the goal [16]. Multiple NFR levels for the same atomic goal specify possible variations on the sensitivity level. There are also alternative operationalizations for the same NFR level. Besides these variations in the NFR graphs, application diversity in the domain can also bring nonfunctional variations. For example, nonfunctional goals concerned in an application may not be so concerned in another application, environmental dependencies for the same operationalization may have different status of satisfaction in different applications, and different applications may have different tradeoff decisions for the same NFR conflicts. All of these serve as starting points for nonfunctional variability analysis. Our method tries to identify these nonfunctional variations in the support of feature context and NFR template, and incorporate these goal-based NFRs into the feature model, along with nonfunctional variations.

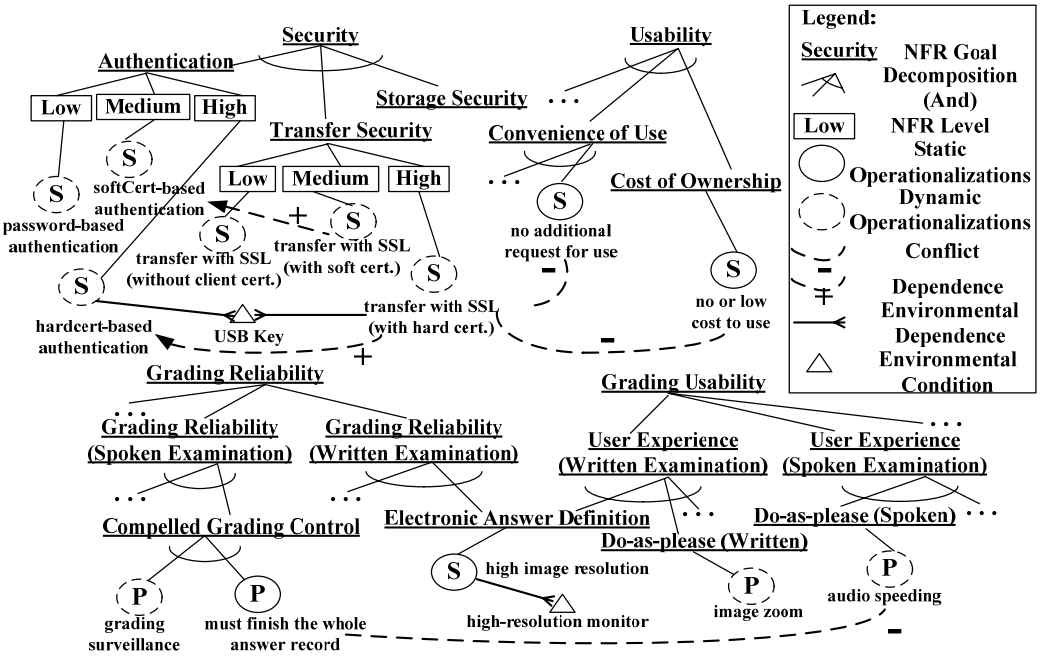


Fig.3. The NFR graph for Security, Usability, Reliability and User Experience in CAGS domain

4 Method Overview

T he whole process of our method is depicted in Figure 4, including the four phases with related artifacts. In Figure 4, arrows represent production and consumption relations between phases and artifacts and also the order of phases/activities. It is similar to the NFR analysis strategy for individual system employed in [1], namely constructing functional and nonfunctional perspectives separately and then integrating them together. The difference is that our method concentrates on nonfunctional variability analysis and NFR-oriented decision modeling. The method includes four major phases: 1) feature context

construction, 2) nonfunctional variability identification, 3) NFR integration and 4) NFR-oriented decision modeling. The starting activity is the feature context construction. For the whole method, the main input artifacts are initial functional feature model and NFR graphs, which represent initial functional and nonfunctional perspectives, respectively. The final outputs are NFR-integrated feature model and feature decision model (underlined artifacts in Figure 4).

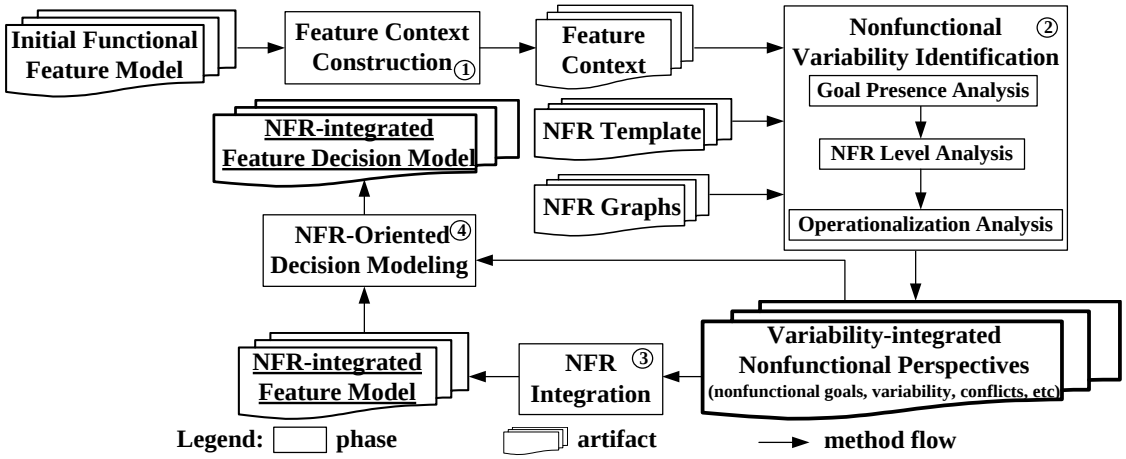


Fig.4. Process of NFR analysis for SPL

4.1 Feature Context Construction

The first phase, feature context construction, constructs feature context model for the initial functional feature model. All of the requirements are based on the environment [13]. Feature context models the real-world context where nonfunctional concerns are originated, including execution scenarios, related people, entities, events and their properties. Variations within the domain scope are also embodied in feature context. In our method, a feature context model is only constructed for the initial functional feature model and servers as an anchor for NFR elicitation and variability analysis.

4.2 Nonfunctional Variability Identification

Nonfunctional variability identification elicits nonfunctional variations from NFR graphs with the support of feature context and NFR templates. As mentioned in section 3.2, nonfunctional variations can originate from different levels of NFR graphs (see Table 1). They have different variation rationales, so different principles should be applied for them. We have different analysis activities for them: goal presence analysis, NFR level analysis and operationalization analysis. After these three activities, we can get complete nonfunctional perspectives, including all the necessary NFR goals in the domain, their variability settings, candidate NFR levels and operationalizations.

For NFR goals (and sub-goals), goal presence analysis will evaluate their presence status according to NFR templates and contextual information in the domain. NFR template in our method encodes generic contexts for nonfunctional concerns. It can be used to evaluate the existence of NFR goals in different applications, and provide heuristics for nonfunctional variability analysis, e.g. different existence status in different applications can make an optional NFR goal. After goal presence analysis,

not-concerned-goals are removed and reserved-goals are further evaluated to be mandatory (concerned in all the applications) or optional (concerned in part of the applications). Different from a single system, NFR tradeoffs for a SPL can be quite different in different applications. To accommodate these different tradeoff policies, initial presence conclusions may be adjusted to add more variations.

NFR level analysis further determines necessary levels for each NFR goal. Although higher NFR level is always better than lower ones, lower levels may need to be reserved due to possible conflicts between higher NFR levels and other NFRs.

Operationalization analysis evaluates the realistic possibility of each operationalization for reserved NFR levels by considering its environmental dependencies. The results can be "always satisfiable", "partially satisfiable" or "unsatisfiable". Accordingly, variability settings of related NFR levels may be adjusted, e.g. a NFR level can be removed if all of its operationalizations are evaluated to be "unsatisfiable".

Table 1. Nonfunctional variations and analysis activities

Variation Level	Variation Rationales	Analysis Activity	Analysis Result
NFR Goal	Alternative OR sub-goals	Goal Presence Analysis	Reserved NFR goals and their presence conclusions (mandatory or optional)
	Diversity on the nonfunctional concerns in different applications		
	Different tradeoff policies in different applications		
NFR Level	Diversity on the sensitivity level of the same NFR goal in the domain	NFR Level Analysis	Possible necessary NFR levels for each reserved NFR goal
	Different tradeoff policies in different applications		
Operationalization	Alternative operationalizations	Operationalization Analysis	Reserved operationalizations and their realistic possibility (always satisfiable, partially satisfiable or unsatisfiable)
	Different applicability in the domain due to the environmental dependencies		

4.3 NFR Integration

Similar to [1], the identified NFRs will finally be incorporated with the initial feature model along with variations from nonfunctional perspective. In NFR integration, NFR-related operationalizations will be incorporated into the feature model, and the existing functional features may be further refined according to the attached nonfunctional concerns. NFR-related feature incorporation and refinement will bring new dependencies, so necessary NFR-related constraints will also be added into feature model.

4.4 NFR-Oriented Decision Modeling

The goal of NFR-oriented decision modeling is to construct functional and nonfunctional perspectives that are integrated into the decision tree for variation resolution in application engineering. It first identifies NFR-related variability decision points by considering NFR conflicts, NFR dependencies and environmental dependencies. Then relationships between NFR-related decisions and functional decisions are investigated to incorporate the two parts of variability decision points. Finally, the NFR-integrated decision model can be obtained, which can provide comprehensive variability decision support for application engineering with both functional and nonfunctional considerations.

5. Method Phases

5.1 Feature Context Construction

5.1.1 Feature Context

NFRs are quality constraints imposed on the system. As for a NFR, whether users care for it or not and its sensitivity are primarily determined by the real-world context of related operations. For example, transfer security emerges when sensitive messages are transferred via public or local network that is also accessible by those who have the tendency of stealing or forgery. Furthermore, the sensitivity of NFRs heavily depends on contextual details. For example, the desired level of transfer security is usually determined by details such as sensitivity of the data transferred, publicity and accessibility of the network, etc.

Nonfunctional requirements, such as security and privacy issues, are originated from human concerns and intents, and thus should be modeled through social concepts [14]. In our method, real-world concept based context model is established for features to enable nonfunctional variability analysis. Feature context is a formal model depicting how the feature behaves and works in the domain that includes people/entities in the real-world, and the relationships among them. Feature context is more of use case scenarios, in that it extends with indirectly involved people/entities to provide a comprehensive context for NFR analysis.

The construction of context model is the process of investigating the real-life contextual elements and structuring them in a formal and knowledge-based model. Each functional feature and the whole system can have their context models at different level of abstractions.

5.1.2 Meta-model of Feature Context

Meta-model of feature context is depicted in Figure 5, providing the skeleton structure of feature context. The kernel of the context is a series of interactions between stakeholders (e.g. grader) and system nodes (e.g. grader client), or between different system nodes. An interaction can input, output or transfer some data via specific medium. Typical medium are devices (usually for interactions between human and system, e.g. keyboard for data input) and networks (usually for interactions between system nodes, e.g. LAN). The data involved in interactions reflect corresponding real objects, which are generated in real-life activities that specific actors participate. For example, answer files in CAGS reflect answer sheets generated by examinations that different types of students participate.

It should be emphasized that meta-model in Figure 5 defines only the skeleton structure of feature context. For specific domain, necessary entities and properties should be added into the context model, e.g. the device scanner and properties defined on examination in Figure 6.

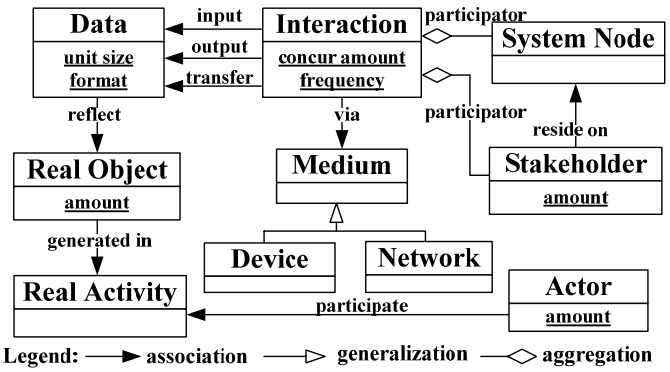


Fig.5. Feature context meta-model

5.1.3 Example of Feature Context Model

A segment of the feature context model for the CAGS domain is depicted in Figure 6. Each element in the Figure is identified with its types and interactions that are drawn as gray boxes for prominence. The ALT and OR nodes represent variable contextual information, namely alternative and OR choices, e.g. the OR node connected to "AnsDispatch" shows that the system can dispatch answers via LAN or Internet or both in an application. In the Figure, feature-interaction mappings are listed and it can be seen that "AnsPackDispatch" has two related interactions while others have exact one related interaction.

From the Figure, we can see that answer files of image format are produced by answer scan with the scanner. Image answer file and audio answer file are alternative inputs for answer package and answer display. "AnswerPack" will produce answer packages, which can be transferred between grading server and client in answer dispatch via LAN or Internet. After that, graders residing on clients can review answers by monitors (image answer) or earphones (audio answer), input mark and comments by keyboards, and then upload them to the server via LAN or Internet.

Properties for contextual entities are also specified, e.g. properties defined on "OralExamination" show that examinations served for in different applications can vary from in-school, region-level to entrance examinations. Descriptions of more contextual properties are omitted here due to the limitation of the space in this paper.

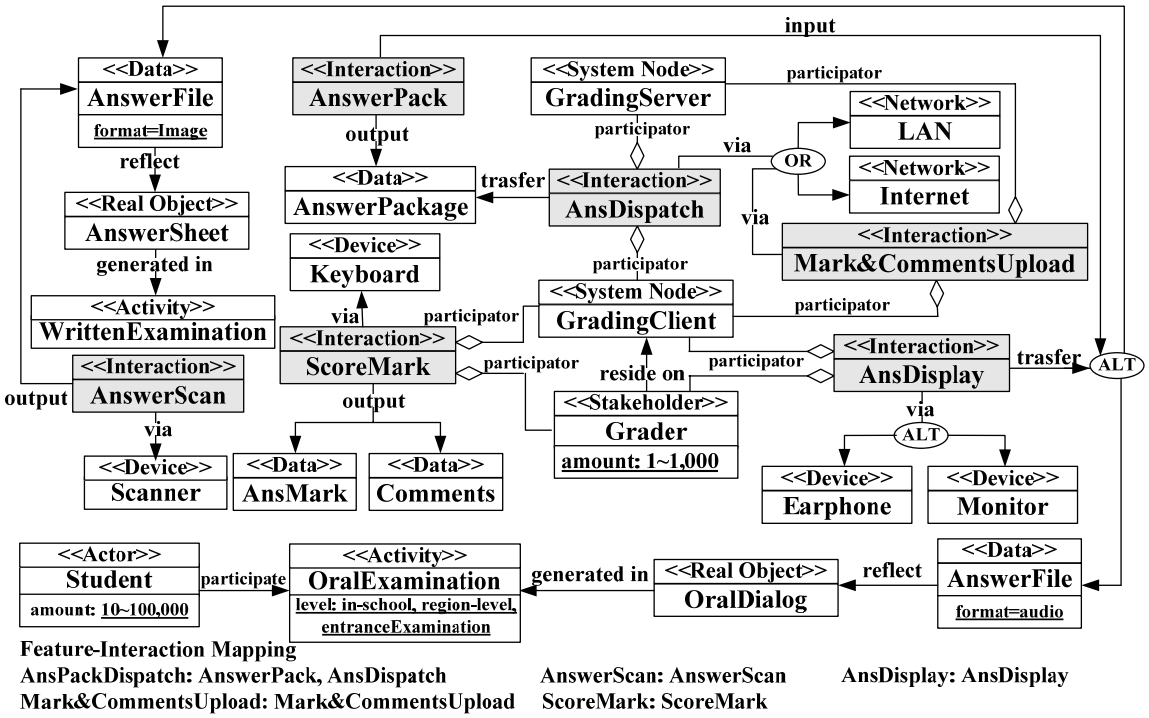


Fig.6. Segment of the feature context model for the CAGS domain

5.1.4 Contextual Variability

NFR analysis in our method is domain-oriented, so the feature context can involve domain-specific contextual variations. In our feature context model, contextual variations can be embodied by ALT/OR nodes and contextual properties.

ALT and OR nodes in feature context represent

alternative and OR contextual information, respectively. For example, in an application of the planned CAGS SPL, the system can provide one or two grading modes of LAN- or Internet-based. It means that LAN- or Internet-based grading is a variation point, and other network types (e.g. wireless network) need not to be considered in CAGS product line.

Contextual properties embody variations by their values, e.g. amount of students participating oral examination is estimated to be 10~100,000 in different applications. Therefore, both large- (over 10,000), medium- (thousands of examinees) and small-scale (hundreds of examinees) examinations should be well supported by the product line, e.g. in the consideration of server performance.

5.2 Nonfunctional Variability Identification

This subsection introduces the three nonfunctional variability analysis activities, then the example nonfunctional perspectives for CAGS as result of nonfunctional variability identification.

5.2.1 Goal Presence Analysis

Goal presence analysis evaluates presence status for each NFR goal. There are some NFRs that are always desired by stakeholders, e.g. usability in Figure 3. For these always-desired NFRs, the initial presence conclusion is always mandatory, but their variability will be further evaluated according to NFR tradeoffs. Other NFRs, such as transfer security are related to specific context. For these context-related NFRs, we introduce NFR templates (general or domain-specific) as heuristics for presence analysis. NFR templates characterize typical contextual situations where certain nonfunctional goals usually emerge. Then goal presence analysis can be performed by matching between feature context and NFR templates.

NFR templates provide heuristics for NFR analysis on two different levels. First, NFR templates specify basic context elements of the typical environment where certain nonfunctional goal emerges. For more particular context level, NFR templates identify key factors of contextual elements. These factors provide a basis for further variability analysis on levels of NFR levels and operationalizations, and NFR-integrated decision modeling.

NFRs can be general or domain-specific. Accordingly, NFR templates can also be general or domain-specific. NFR templates for transfer security and “Compelled Grading Control” are presented in Tables 2 and 3, respectively. The former is domain-independent, while the latter is specific to the CAGS domain. From Table 2, we can see that transfer security emerges in information transfer via certain medium, and basic context elements include such as information transferred, the medium and potential wiretapper. Typical qualifications on these context elements are also specified, e.g. the medium is also accessible for potential wiretapper, which can gain profit or curiosity from stealing or forgery. Factors identified for transfer security analysis include publicity of the system, sensitivity of the transferred data and the stealing/forgery capability of potential wiretappers.

Table 2. NFR template for transfer security

Context Elements	Qualification	Factors
Overall Context	Information transfer via the Medium	Publicity: publicity of the system
Data	Sensitive data	Sensitivity of data

Medium	Medium for data transferring	
Potential Wiretapper	Someone who can access the same medium and gain profit or curiosity by stealing or forgery	Capability: the capability of practicing stealing or forgery

For domain-specific NFR goals, corresponding templates are also domain-specific. For example, NFR template for "Compelled Grading Control" is given in Table 3, which is defined by domain-specific contextual information on "Examination". We can see "Compelled Grading Control" usually emerges in official examinations. These qualifications can help to determine the variability on the goal. For example, if only high-end examinations are targeted by the CAGS group, then it will be a mandatory NFR goal. However, the current CAGS group plans to cover both high- and low-end examinations, so "Compelled Grading Control" will be evaluated to be an optional NFR goal.

Table 3. NFR template for "Compelled Grading Control"

Context Elements	Qualification	Factors
Examination	Official examination	Examination Level Examinee Number Grader Number

Then NFR goals can be evaluated according to corresponding templates and feature contexts. Heuristic rules for initial NFR presence analysis are straightforward:

- NFR goals whose elements are common in all applications are evaluated to be mandatory;
- NFR goals whose elements only emerge in part of applications are evaluated to be optional.

NFR conflicts are a main origination of nonfunctional variations. After initial presence analysis, irrelevant NFRs are removed, and then tradeoff analysis should be performed for all the reserved goals. One of the main concerns is whether consistent tradeoff can be made for each conflict, by considering all the systems within the targeted SPL scope. If so, the conflict can be resolved similar to what is done for a single system. If not, the conflict will be suspended to be resolved in application engineering, and corresponding conflict-related variations that originate from.

Suspended NFR conflicts mean related NFR goals may be abandoned in tradeoffs, so an additional rule should be considered to embody this type of conflict-related variations: **If a mandatory NFR goal conflicts with other NFRs, then it should be adjusted to be an optional.**

Part of the results from NFR presence analysis for CAGS domain is presented in Table 4. "Transfer Security" is identified to be mandatory, since in all CAGS applications sensitive answers and marks will be transferred via networks. "Compelled Grading Control" is evaluated to be optional, since both high- and low-end in-school examinations are involved in the SPL scope. Other NFR goals can be determined to be mandatory at the moment, but will be evaluated further in the following steps.

5.2.2 NFR Level Analysis

NFR levels are the refinement of NFRs on sensibility. NFR level analysis determines necessary levels for each mandatory and optional NFR goals by evaluating context factors identified

in corresponding NFR templates. One of the characteristics of NFR levels is that the higher level is always better than the lower one. Therefore, from the aspect of single NFR goal, if multiple levels exist, higher one can always replace lower ones. However, in case if there are conflicts between NFRs, then lower NFR levels may also be reserved if higher one conflicts with other NFRs. Thus, reserved different NFR levels can constitute multiple candidate choices together with other related NFRs to meet different preferences. For example, medium "Transfer Security" can be involved in low-end applications with much convenience and low cost of ownership.

Then we can summarize principles for NFR level analysis as follows:

- If a NFR level doesn't conflict with any other NFR goal, then lower levels of the same NFR should be removed;
- If NFR levels higher than a level conflict with other NFR goals, the level should be reserved.

According to the principles, low "Transfer Security" can be removed, since the medium level does not conflict with others; both medium and high "Transfer Security" are reserved, since the high level conflicts with "Convenience of Use" and "Cost of Ownership" (see Figure 3).

5.2.3 Operationalization Analysis

Operationalization analysis evaluates operationalizations for all reserved NFR levels according to realistic possibility. Operationalizations provide real implementation support for different NFR levels. However, they may depend on certain environmental conditions, e.g. "High Image Resolution" depends on "High-Resolution Monitor". This kind of environmental dependencies should be further evaluated to eliminate inapplicable ones or adjust the options for NFR levels. Similar to NFR presence analysis and level analysis, operationalization analysis also evaluates environmental dependencies by considering all the systems within the domain scope. The results can be "always/fully satisfiable", "partially satisfiable" or "unsatisfiable". For example, "Transfer with hard-cert. SSL" depends on "USB Key", which is evaluated to be "always satisfiable" although it requests additional costs. "High Image Resolution" depends on "High-Resolution Monitor", which is evaluated to be "partially satisfiable", since in some small schools, it is impossible to replace the entire PC monitors only to meet the requirements of an electronic grading system.

Then we can summarize principles for operationalization analysis as follows:

- If an operationalization is evaluated to be unsatisfiable, then it should be removed;
- If all the operationalizations of a NFR level are removed, then it should be removed;
- If all the reserved operationalizations of a NFR level are partially satisfiable, then it should be adjusted to be optional.

From the last two principles, it can be seen that operationalization analysis can further adjust the variability setting of NFR levels. For example, after operationalization analysis, "Electronic

Answer Definition" is adjusted to be optional, for its operationalization "High Image Resolution" can only be enabled by part of the systems in the domain.

5.2.4 Variability-Integrated Nonfunctional Perspectives

After these steps, we can get the variability-integrated nonfunctional perspectives (see Table 4), including reserved NFR goals and their optionality settings (mandatory or optional), NFR levels and operationalizations. Variations in nonfunctional perspectives are embodied at different levels (see Table 4 and the NFR graph in Figure 3):

- Optional NFR goals: optional NFR goals are optional decomposition elements of their parent goals;
- Alternative and OR sub-goals: if a NFR goal has multiple reserved sub-goals of OR decomposition, they constitute alternative or OR sub-goals of it, depending on whether they are exclusive or not.
- Alternative NFR levels: multiple reserved NFR levels for the same NFR goal represent alternative implementation levels.

As listed in Table 4, "Transfer Security" is determined to be mandatory and alternative due to its two candidate levels. "Compelled Grading Control", "Electronic Answer Definition" and "Do-as-please (Spoken)" are evaluated to be optional, since each of them has only one applicable level. These nonfunctional variations will be integrated into the feature model and feature decision model. In order to provide decision bases for the decision model, related NFR conflicts and rational are also attached.

Table 4. Part of the variability-integrated nonfunctional perspectives for CAGS domain

NFR Goal	Optionality	Levels	Operationalization	Conflicts	Rational
Transfer Security	mandatory	medium	transfer with SSL (with soft cert.)	None	Concerned in all CAGS systems
		High	transfer with SSL (with hard cert.)	Convenience of Use	
				Cost of Ownership	
Authentication	mandatory	medium	soft-cert-based authentication	None	Concerned in all CAGS systems
		High	hard-cert-based authentication	Convenience of Use	
				Cost of Ownership	
Compelled Grading Control	optional	N/A	grading surveillance	Do-as-please (Spoken)	Concerned in high-end examinations
			must finish the whole answer record		
Electronic Answer Definition	optional	N/A	high image resolution	None	Always desired by users but only partially applicable in the domain
Do-as-please (Written)	mandatory	N/A	image zoom	None	Always desired by users
Do-as-please (Spoken)	optional	N/A	audio speeding	Compelled Grading Control	Always desired by users, but conflicts with other NFRs

5.3 NFR Integration

NFR goals are soft-goals, which are continuously decomposed into sub-goals until they can be satisfied (operationalized) by operationalizations [1]. In order to ensure the realization of desired NFR goals, related goals and operationalizations should first be incorporated into the feature model. Then variations embodied in NFR goals should be treated to be properly embodied in the feature model. Finally, NFR-related variability constraints should be elicited for NFR decision modeling.

5.3.1 Operationalization Incorporation

In NFR integration, it is necessary to consider where the initial conceptual model will be affected by the operationalizations and how to include them [1]. The first problem can be resolved by using the heuristics from NFR goal presence analysis, since each reserved NFR goal is evaluated to be mandatory or optional by matching between NFR templates and context information of certain feature. For example, "Transfer Security" affects functional features involved in sensitive data transfer on the network, i.e. "AnsPackDispatch", "Mark&CommentsUpload", "Query&Stat." and "Check&Mediation" in the CAGS domain. Therefore, reserved NFR levels and operationalizations for "Transfer Security" can be attached to these features. As for how to integrate operationalizations into the feature model, we have the following rules:

- Dynamic operationalization is integrated as a sub-feature of the affected functional feature;
- Static operationalization is integrated as restriction on the affected functional feature;
- Dynamic operationalizations for NFR goals affecting multiple functional features are considered as crosscutting features and interactions between affected features and these crosscutting features should be recorded.

5.3.2 Nonfunctional Variability Treatment

As shown in section 5.2.4, nonfunctional variations are embodied in NFR goals and levels. These variations should be further treated to be properly embodied in the feature model. Considering dynamic/static operationalizations and the three kinds of variations of optional/alternative/OR elements, we have the following rules:

- Integrated dynamic operationalization should be set to be optional if the responding NFR goal is optional;
- Integrated static restriction should be set to be optional if the responding NFR goal is optional, and then the optional restriction should be transformed to a facet to provide customizable choices and non-optional restriction transformed to an implementation note for the feature;
- If a NFR goal has multiple OR sub-goals, then operationalizations for these goals are integrated as alternative/OR sub-features;
- If a NFR goal has multiple reserved levels, then operationalizations for these levels are integrated as alternative sub-features.

In domain analysis and design, separating commonality/variability and localizing variations are the primary considerations to maximize the reusability of SPL assets. After operationalization incorporation and variability treatment, optional or alternative restrictions may be imposed on functional features, so an additional rule should be

considered: **functional features influenced by optional or alternative nonfunctional restrictions should be refined to further separate commonality/variability and localize variations**. For example, "Login" is identified as an atomic feature in the initial feature model (see Figure 2), since there are no variations within it. When the variable NFR goal "Authentication" (medium or high level) is attached to it, the feature "Login" should be refined to localize the variability within "Authentication" and reserve commonality in the separated feature "Initialization" (initialize the user and system information after successful login, see Figure 7).

Following the rules of operationalization incorporation and nonfunctional variability treatment, we can get the CAGS feature model with NFRs as shown in Figure 7 (some elements are omitted for clearness), in which NFR-related features are represented by dashed panes and NFR goals are connected with their operationalizations. In the Figure, "ImageZoom" is set to be mandatory since "Do-as-please (Written)" is mandatory, "AudioSpeeding" and "GradingSurveillance" are set to be optional since "Do-as-please (Spoken)" and "Compelled Grading Control" are optional. "Must Finish the Whole Answer Record" and "High Image Resolution" are integrated as facets imposed on corresponding features to provide different choices in application engineering, since they are for optional NFR goals. "Transfer with SSL" is a new abstract feature to represent the choice of the two levels of high and medium. It is also a crosscutting feature, so it is extracted from the three affected features and corresponding crosscutting interactions are recorded.

5.3.3 Constraint Elicitation

Variations embodied in the domain feature model are to be resolved in application engineering to acquire different customizations. The variability resolutions are embodied by determination of binding-states (bound, removed or pending) of optional, alternative and OR features. The customizations are not free, but restricted by feature constraints. Constraint is a kind of static dependency among binding-states of features, and provides a way to verify the results of requirement customization and release planning [8].

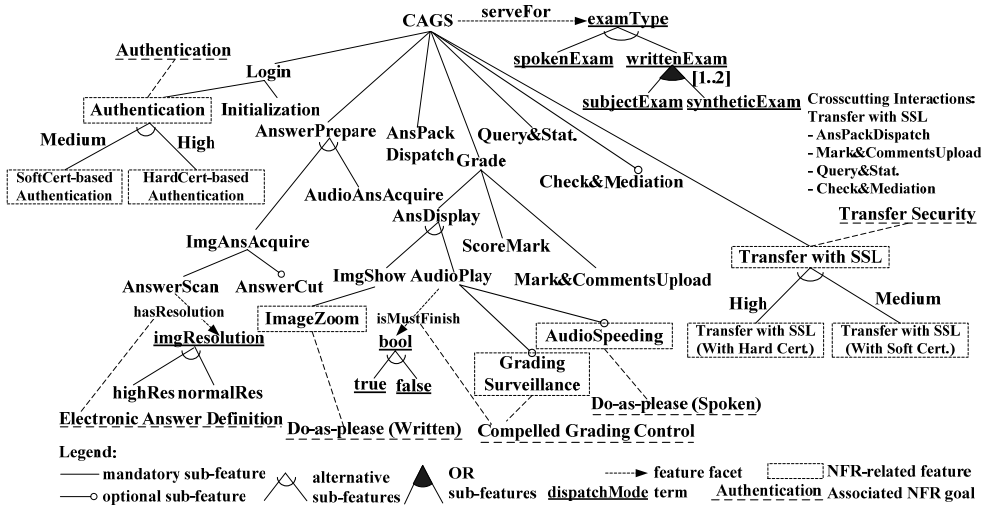


Fig.7. CAGS feature model with NFRs

The most-used constraints in the feature model are ***requires*** and ***excludes***, in which constraint "A ***requires*** B" denotes that if A is bound then B must be bound too and "A ***excludes*** B" means A and B can not be bound together. Detailed discussions on feature constraints can be found in [8].

Constraints first exist in the initial functional feature model, such as constraints C1-C5 presented in Table 5. For example, if the CAGS application serves for synthetic examinations, then the answer packages must be dispatched by questions according to C5 (synthetic examination paper is composed of questions of different subjects) and "ImgShow" and "ImgAnsAcquire" are bound according to C3 and C4 ("syntheticExam" must be "writtenExam"). These functional feature constraints can be identified in variability analysis, which is involved in most feature-based domain analysis methods.

In our method, NFR-related features and variations are integrated into the feature model along with corresponding NFR goals. These NFR goals may conflict with other NFRs or depend on certain environmental conditions (see Table 4). These kinds of conflicts and dependencies will bring new constraints into the feature model, such as constraints C6-C8 in Table 5. NFR-related constraints can be categorized as follows:

- NFR-dependency-related constraints. Operationalizations dependencies bring ***requires*** constraints between corresponding features, e.g. C6 and C7.
- NFR-conflict-related constraints. Operationalizations for conflicting NFR goals have ***excludes*** constraints between corresponding features, e.g. C8.

Table 5. Constraints in the CAGS feature model

C1: [CAGS serveFor spokenExam] ***requires*** [AudioPlay]
C2: [CAGS serveFor spokenExam] ***requires*** [AudioAnsAcquire]
C3: [CAGS serveFor writtenExam] ***requires*** [ImgShow]
C4: [CAGS serveFor writtenExam] ***requires*** [ImgAnsAcquire]
C5: [CAGS serveFor syntheticExam] ***requires*** [AnsPackDispatch hasDispatchMode byQuestion]
C6: [Transfer with SSL (With Hard Cert.)] ***requires*** [HardCert-based Authentication]
C7: [Transfer with SSL (With Soft Cert.)] ***requires*** [SoftCert-based Authentication]
C8: [AudioPlay_mustFinish=true] ***excludes*** [AudioSpeeding]

5.4 NFR Decision Modeling

Decision model provides a systematic decision path for application customization. In the decision model, each variation point in domain assets should be connected to an open decision in the decision model [9]. The decision model for feature model can be denoted by a decision tree, in which decision points are structured in a partial order. Feature decision model can be constructed by analyzing the partial orders between variations. For example, by considering C1-C5 only, we can see CAGS that requires to consider whether "spokenExam" or "writtenExam" should be determined first, then whether "syntheticExam" or "subjectExam" should be determined, since the latter choice is the refinement of the former.

After NFR-related features and constraints are incorporated into the feature model, we should also integrate NFR-related decisions into the decision model. Besides feature constraints, there are another three kinds of external dependencies that can influence the decision model:

- NFR conflicts. Suspended NFR conflicts bring decision points of different tradeoff choices;
- NFR dependencies. If NFR goal A depends on NFR goal B, then B will be chosen if A is bound by previous decisions.
- Environmental dependencies. If a NFR operationalization depends on certain environmental condition, then the condition should be involved in the decision model.

Decision model is constructed for variation resolution, so only optional and alternative goals and features are taken into account in decision modeling. NFR goals concerned by clients are primary decision points, so they should first be considered in NFR decision modeling based on NFR conflicts and dependencies. Usually, in NFR goal decision, higher goals should be evaluated first, if chosen then its mandatory sub-goals are chosen automatically and optional sub-goals are evaluated sequentially. In the process, if NFR dependencies are involved, then required NFR goals are also chosen; if a current NFR goal is involved in conflicts, then a tradeoff decision should be made. Usually there is a dominant NFR goal in NFR conflict, e.g. in conflict between security and cost/convenience, security is usually dominant. Therefore, in decision modeling, the dominant NFR goal in a conflict is evaluated first and conflicting goals are taken as tradeoff annotations on the decision point. For example, in Figure 8, “with some inconvenience and additional cost” is annotated with the choice “high” for “Transfer Security Level”.

NFR decisions can also be related to functional decisions, if the goal is related to an optional or alternative functional feature. In this case, NFR decision is related to the functional decision as a next decision. For example, “Compelled Grading Control” is a next decision point after “Exam Type”, since it is related to “AudioPlay” and should be determined after “AudioPlay” is bound. Environmental dependencies are also modeled as decision points according to similar policies, e.g. “High-Resolution Monitor” in Figure 8 is considered after “ExamType” is determined to be “written”. Besides these decision points, other functional or nonfunctional decision points can be placed in the decision tree according to inter-feature constraints.

The final NFR-integrated CAGS feature decision model is presented in Figure 8, in which decision points are denoted by ellipses with “?” mark, decisions are represented by arrows, bound features are listed in the pane after corresponding decisions are made. Decisions made at each decision point are labeled on the following arrows, and arrows without labels represent the next decision points. In the decision model, only variability-related features are involved. For example, “ImageZoom” in Figure 7 is not included in Figure 8, since it is a mandatory feature. From the decision model, we can see that “ExamType” and “Trasfer Security Level” are two main variations and they are orthogonal. “ExamType” is followed by decisions for “Compelled Grading Control” and “High-Resolution Monitor”. The decision for

"Transfer Security Level" also determines the authentication policy according to the dependency between these two NFR goals.

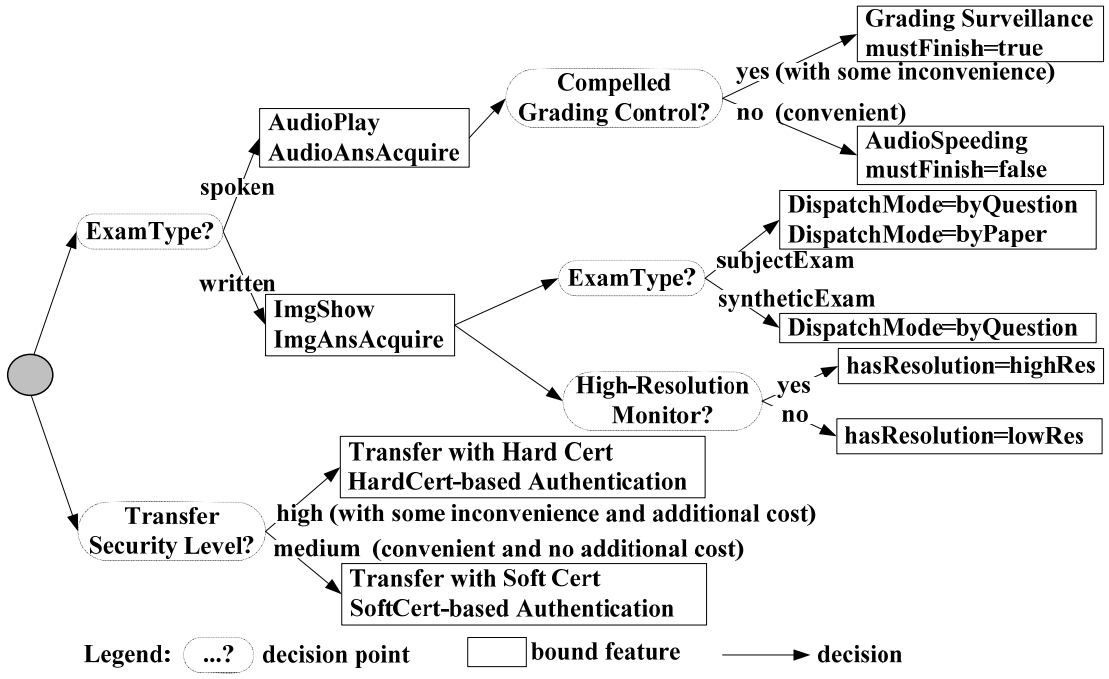


Fig.8. NFR-integrated CAGS feature decision model

6. Evaluation and Discussion

6.1 Evaluation

The feature model, as a product of domain analysis in SPL development, should comprehensively embody requirements of all the applications in the domain from both functional and nonfunctional perspectives. Sometimes, obvious NFRs can be naturally identified and incorporated in the feature model, but other vague NFRs may be omitted, not to mention nonfunctional variations, which often originate from implicit factors such as NFR conflicts, NFR dependencies and environmental dependencies, etc. Therefore, nonfunctional requirements analysis in SPL development is still a big challenge.

The method proposed in this paper provides a systematic process for NFR analysis in SPL development. It is a good complement to traditional feature-based domain analysis methods. The effectiveness of the method has been well recognized by related engineers through the comprehensive method application with the CAGS group, although additional analysis activities and considerations are requested. Feedbacks indicate that the method helped the product line development in several aspects:

1) The existence and variability of implicit NFRs embodied in NFR graphs, feature contexts and NFR templates help a lot on NFR elicitation and variability identification;

2) Nonfunctional variations are analyzed from multiple aspects of contexts where those nonfunctional concerns originate from, NFR conflicts and environmental dependencies, by providing comprehensive clues for nonfunctional variability analysis;

3) Incorporation of NFRs with their variations

in feature model help produce better product line design with more comprehensive variability considerations (e.g. commonality/variability separation and variation localization);

4) The decision model with combined functional and nonfunctional perspectives eases the communications with customers by explicitly stating all the decision choices on functional, nonfunctional concerns and external conditions in an integrated mode.

Besides the CAGS product line, we are also conducting a case study on another Web-based product line of college financial management system, which also embodies some critical nonfunctional requirements, e.g. security, traceability of financial affairs, etc.

6.2 Discussion

As stated in [1], NFRs have received little attention in literatures in spite of their importance. Sutcliffe et al. [3] also mentioned that existing works focus too much on user activity and system functionality, rather than system qualities, generally referred to as nonfunctional requirements. The neglect is even more serious in SPL development. This situation is partly due to very vague nature of nonfunctional requirements [1-4] and nonfunctional variations. Most NFRs are closely tied with functional requirements, so they are often difficult to distinguish from each other. On the other hand, nonfunctional variations often originate from NFR tradeoffs, which makes them more confused.

1) Nonfunctional Features vs. Functional Features

Most NFRs have close ties with functional specification and are gradually refined into the requirements specification, leading some to argue that NFRs are redundant [3]. For example, most NFR goals concerned in the CAGS domain are integrated into as functional features or restrictions imposed on functional features (see Figure 7). Another typical example for the confusion is “Check&Mediation”, which is also a great contribution to “Grading Reliability” by conducting additional grade checking and mediation when large divergence emerges. However, it is identified as a functional feature from the beginning (see Figure 2), because it has been an ingrained part of grading for most of the clients. This further illuminates that it is hard to distinguish functional and nonfunctional requirements in many cases.

However, NFR analysis is still essential in most cases, especially for SPL development. First, there are still some NFRs without obvious operationalizations, e.g. cost of ownership. Second, in most cases, NFR goals are first identified then operationalized into corresponding functions, e.g. security-related functions. Third, systematic analysis of NFR origination, contextual factors, and operationalizations can help to have comprehensive nonfunctional perspectives, including applicable NFR level, conflicts and variations. For example, NFR templates and feature contexts in our method provide the basis of nonfunctional variability analysis.

2) Nonfunctional Variations vs. NFR Tradeoffs

It can be observed that most nonfunctional variations originate from tradeoffs of conflicting NFRs, otherwise a high-level nonfunctional

quality is always desired. For example, the reason why high-level security is not chosen in some of the applications is due to the cost of additional hard-certifications and the influence on to the usability issues. These kinds of conflict detection and resolution for NFRs are also common in a single product development (e.g. [1]). However, nonfunctional variability analysis in SPL is more of conflicts identification and resolution. In a single product development, tradeoff decisions can be made only from the concerns in the current system. While in SPL development, NFR tradeoffs are first made in domain engineering for all the systems within the SPL scope, and then completely resolved in application engineering. Some NFR tradeoffs can be made in domain engineering, if consistent decisions can be made for all the systems. For example, if the CAGS group is targeting high-end markets only, then the tradeoff between security and cost/convenience can be made to choose hard-certification-based authentication and transfer in domain engineering. In this situation, authentication and transfer will not become variations, since a unified decision can be made for all the systems. However, if a consistent tradeoff can not be made, then corresponding conflicts will bring variations into the domain model, e.g. the conflict between security and cost/convenience in current CAGS SPL (see Figure 7). In domain analysis, these kinds of conflict-caused nonfunctional variations should be identified and integrated into the feature model to provide a basis for domain design and implementation.

7. Related Work

A NFR knowledge base is usually employed in NFR analysis due to the vagueness of NFRs and the similarity of NFR properties in different systems. For example, the knowledge base used in [1] records possible refinements and operationalizations for each NFR item, the method proposed in [3] contains a template for each high-level NFR with embedded heuristics for scenario creation, validation and benchmark assessment by metrics. Cysneiros et al. [1] propose a systematic process of NFR elicitation and incorporation into UML-based conceptual models. Their elicitation process is based on the use of a lexicon that will not only be used to anchor both functional and nonfunctional models, but also to drive NFR elicitation. In their method, NFRs are integrated into UML diagrams, including the Class, Sequence, and Collaboration Diagrams. NFR conflict identification and resolution is also involved in the method. Sutcliffe et al. [3] propose an analysis method that describes scenario templates for NFRs, with heuristics for scenario generation, elaboration and validation. Liu et al. [14] propose a method of analyzing security and privacy requirements based on social relationships between problem domain actors using the *i** modeling language. The method involves techniques that assist in attacker, vulnerability, countermeasure analysis and access control analysis.

These NFR analysis methods introduced above have similar concepts in context- or scenario-based NFR elicitation, separation then integration of functional and nonfunctional

perspectives, and the analysis with a NFR knowledge base. However, they all concentrate on NFR analysis for single software-intensive systems and do not consider nonfunctional variability analysis and integration for a series of similar systems.

Lee et al. [18] propose a framework that integrates the notions of goals, scenarios, and viewpoints by using the ontological domain modeling techniques. The framework has been applied to the information security requirements domain analysis for the US DoD (Department of Defense) information system Certification and Accreditation (C&A) activities [19]. The ontological domain requirements model is constructed from various regulatory documents at different levels of abstractions [20]. It is notable that the NFRs in the C&A process are modeled and represented as both taxonomical and non-taxonomical ways and later further relationships are inferred through the interaction with the domain analysts. We believe our work can adopt some of these modeling techniques with ontological expressive power to perform and improve the NFR analysis in SPL development.

In recent SPL researches, nonfunctional features are often referred together with functional features in domain analysis. For example, Kang et al. [7] propose that nonfunctional features are identified from the marketing plan, and they should be identified together with functional features in domain analysis for SPL asset development. Yu et al. [16] propose a process that generates a high variability software design from a goal model. The process is supported by heuristic rules that can guide the design. The method includes a method of generating feature model from variability-embodied goal model. Soft goals that represent stakeholder preferences are also mentioned, but nonfunctional variability identification and integration are not mentioned. Moon et al. [5] also indicate that any quality item that occurs in the nonfunctional requirements should have a relationship to functional requirements and other quality items. Liaskos et al. [17] propose a goal-based variability acquisition and analysis method based on the semantic characterization of OR-decompositions of goals. The method addresses variability-intensive goal refinement and derivation, but the integration with functional feature model is not considered. Halmans et al. [12] indicate that the category quality in SPL development subsumes variability aspects concerning nonfunctional/quality requirements such as security, availability or scalability. They propose use cases as communication medium for the SPL variability, and indicate that use cases are not suited to document essential variability concerning the quality. They emphasize that quality-related variability aspects must be interrelated with the functional and environmental variability aspects, but do not involve NFR analysis and representation in their method.

From these works, we can learn NFR analysis in SPL development has been recognized in some of the current SPL researches. In these works, it is treated similar to NFR analysis for a single system in most cases, e.g. NFRs in SPL are considered as holistic requirements of all the systems. However,

marketing plan of a SPL may cover applications with diverse nonfunctional concerns. For example, the CAGS group involves both high-end (high security and reliability with additional cost and some inconvenience) and low-end (convenient and without additional cost) examinations in their targeted markets. Therefore, nonfunctional variations exist in some SPLs and are essential in domain analysis. Our method proposed in this paper focuses on nonfunctional variability analysis and integration, providing a systematic way for NFR elicitation, variability analysis, integration and decision modeling.

8. Conclusion and Future Works

In SPL development, NFRs and nonfunctional variations should be carefully identified and represented in domain analysis, just as what has been done for typical functional requirements, to provide a comprehensive basis for variability-oriented design and implementation. NFRs are usually hidden in everyone's minds [1], so NFR elicitation is difficult, and even harder in SPL development for the requirements to analyze nonfunctional variations among different systems that exist within a domain. In this paper, we propose a context-based method of NFR analysis for SPL development. In the method, NFRs are materialized by connecting NFR goals with real-world context, so nonfunctional variability analysis can be performed by context analysis for the whole domain. After variability analysis, our method integrates both the functional and nonfunctional perspectives by integrating NFR-related goals and operationalizations into the initial functional domain feature model. NFR-related constraints are also elicited to construct the NFR-oriented decision model to facilitate application-oriented feature model customization.

In our future works, we will concentrate on more systematic NFR-oriented analysis and design method for SPL development, including NFR elicitation, variability analysis and NFR-oriented variability design. On the other hand, we will try to integrate the method into our existing tools for SPL analysis and implementation, and perform more case studies on SPL development for high-confidence domains. We will integrate the NFR analysis method proposed in this paper into our feature modeling tool [10] to enrich the nonfunctional feature modeling capability, including context modeling, nonfunctional variations identification and representation, NFR integration and decision modeling, etc. We will also integrate the comprehensive decision model into our recently developed product derivation tool [11] to provide a systematic feature decision support for application generation.

References

[1] Luiz Marcio Cysneiros, Julio Cesar Sampaio do Prado Leite: Nonfunctional Requirements: From Elicitation to Conceptual Models. *IEEE Trans. Software Eng.*, 2004, 30(5): 328-350.

[2] John Mylopoulos, Lawrence Chung, Brian A. Nixon: Representing and Using Nonfunctional Requirements: A Process-Oriented Approach. *IEEE Trans. Software Eng.*, 1992, 18(6): 483-497.

[3] Alistair Sutcliffe and Shailey Minocha. Scenario-based Analysis of Non-Functional Requirements. Workshop on Requirements

Engineering For Software Quality (REFSQ'98) at CAiSE'98 in Pisa, Italy, June 1998.

[4] Lawrence Chung, Brian A. Nixon, Eric Yu, John Mylopoulos. Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, 2000..

[5] Mikyeong Moon, Keunhyuk Yeom, and Heung Seok Chae. An Approach to Developing Domain Requirements as a Core Asset Based on Commonality and Variability Analysis in a Product Line. *IEEE Trans. Software Eng.*, 2005, 31(7): 551-569.

[6] K. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University, November 1990.

[7] Kyo C. Kang, Patrick Donohoe, Eunman Koh, Jaejoon Lee, and Kwanwoo Lee. Using a Marketing and Product Plan as a Key Driver for Product Line Asset Development. In Proc. 2nd Int. Conf. Software Product Lines (SPLC), San Diego, CA, USA, August 19-22, 2002, pp.366-382.

[8] Wei Zhang, Hong Mei, Haiyan Zhao. Feature-driven requirement dependency analysis and high-level software design. *Requirements Eng.*, 2006, 11(3): 205–220.

[9] Cristina Gacek, Michalis Anastasopoules. Implementing Product Line Variabilities. In Proc. 2001 Int. Software Reusability Symp. (SSR), Toronto, Ontario, Canada, May 18-20, 2001, pp. 109-117.

[10] Xin Peng, Wenyun Zhao, Yunjiao Xue, Yijian Wu. Ontology-Based Feature Modeling and Application-Oriented Tailoring. In Proc. 9th Int. Conf. Software Reuse (ICSR), Turin, Italy, June 12-15, 2006, pp.87-100.

[11] Xin Peng, Liwei Shen, Wenyun Zhao. Feature Implementation Modeling based Product Derivation in Software Product Line. In Proc. 10th Int. Conf. Software Reuse (ICSR), Beijing, China, May 25-29, 2008, pp.142-153.

[12] Günter Halmans, Klaus Pohl. Communicating the variability of a software-product family to customers. *Software and System Modeling*, 2003, 2(1): 15-36.

[13] Zhi Jin. Revisiting the Meaning of Requirements. *J. Comput. Sci. & Technol.*, 2006, 21(1): 32-40.

[14] Lin Liu, Eric Yu, John Mylopoulos. Security and Privacy Requirements Analysis within a Social Setting. In Proc. 11th IEEE Int. Conf. Requirements Engineering (RE), Monterey Bay, CA, USA, September 8-12, 2003, pp.151-161.

[15] Yijian Wu, Wenyun Zhao, Xin Peng, Yunjiao Xue. A Computer Aided Grading System for Subjective Tests. In Proc. 2006 Advanced Int. Conf. Telecommunications and Int. Conf. Internet and Web Applications and Services (AICT/ICIW), Guadeloupe, French Caribbean, February 19-25, 2006.

[16] Y. Yu, A. Lapouchnian, S. Liaskos J. Mylopoulos and J.C.S.P. Leite. From goals to high-variability software design. In Proc. 17th Int. Symposium on Methodologies for Intelligent Systems (ISMIS), Toronto, Canada, May 20-23, 2008, pp.1-16.

[17] S. Liaskos, A. Lapouchnian, Y. Yu, E. Yu and J. Mylopoulos. On Goal-based Variability

Acquisition and Analysis. In Proc. 14th IEEE International Conference on Requirements Engineering (RE), Minnesota, USA, September 11-15, 2006, pp.76-85.

[18] Lee, S.W. and Gandhi, R. A.. Ontology-based Active Requirements Engineering Framework. In Proc. 12th Asia-Pacific Software Eng. Conf. (APSEC), Dec. 15-17, 2005, pp. 481-490.

[19] Lee, S.W. and Gandhi, R.A. Requirements as Enablers for Software Assurance. *CrossTalk: The Journal of Defense Software Engineering*, 2006, 19(12): 20-24.

[20] Lee, S. W., Muthurajan, D., Gandhi, R. A., Yavagal, D., and Ahn, G.. Building Decision Support Problem Domain Ontology from Security Requirements to Engineer Software-intensive Systems. *Intl. Journal on Software Eng. and Knowledge Eng.*, 2006, 16(6): 851-884.