International Journal of Fuzzy Logic and Intelligent Systems, vol. 5, no. 3, September 2005 pp. 268-273

Specialized Product-Line Development Methodology for Developing the Embedded System

Kisam Hong, Heebyung Yoon

Dept. of Computer & Information Science, Korea National Defense University

Abstract

We propose the specialized product-line development methodology for developing the embedded system of an MSDFS (Multi Sensor Data Fusion System : called MSDFS). The product-line methodology provides a simultaneous design between software and hardware, high level reusability. However this is insufficient in requirement analysis stage due to be focused on software architecture, detailed design and code. Thus we apply the business model based on IDEF0 technique to traditional methodology. In this paper, we describe the processes of developing Core-Asset, which are requirement analysis, feature modeling, validation. The proposed model gives the efficient result for eliciting features, and ensures the high level reusability of modules performing on embedded system.

Key Words : Embedded System, Product-Line, Sensor Data Fusion, Methodology, IDEF0

1. Introduction

The MSDFS(Multi Sensor Data Fusion System) is a specialized system that integrate and fuse the acquired data from multi sensor. This refined data offer a minimum error of identifying track. Identifying a lot of similarity among applications, This system acquire a high level reusability of compositions and this system needs a specialized development methodology. However current system engineering often leads a large number of inflexible, dedicated systems in the embedded system that together needs a low power, weight and installation space and produce a high maintenance costs. Thus we apply a specialized product line methodology for developing this MSDFS.

The product line allows the degree of reusability to be optimized across different systems while simultaneously preserving the overall quality. This supports the need to develop more integrated and flexible multi-functional systems quickly and economically. Thus the purpose of this paper is to report the results obtained from a case study in developing an MSDFS applied PLD.

The remainder of this paper is organized as follow : Chapter 2 explains the MSDFS and PLD. Chapter 3 shows the result of approaching of PLD for MSDFS. Chapter 4 and 5 show the validation method for proposal and conclusion. For explaining proposal we show context model, feature model and validation model.

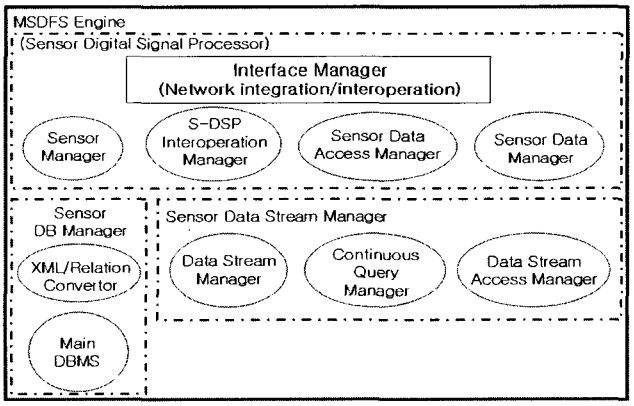
2. Related Work

2.1 MSDFS

The MSDFS is a specialized system that integrate and fuse the acquired data from multi sensor. This refined data offer a minimum error of identifying track[3].

2.1.1 Construction of MSDFS

[Figure 1] shows construction of MSDFS. This system is organized four factor : Sensor, Sensor data processor, Sensor data management system, Sensor stream management system[4].



[Figure 1] Construction of MSDFS

2.1.2 Sensor

In this paper we assume that sensors interfaced with MSDFS are an SPS-95K and DSQS-21BZ. An SPS-95K is a kind of surface radar and an DSQS-21BZ is a kind of SONAR. This radars obtain data video, voice, range zero trigger, etc from surface and undersea.

2.1.3 Sensor data processor

Sensor data processor processes a continuous sensor data. This provides a function that registers a sensor, manages a

Manuscript received Jul. 29, 2005; revised Aug. 30, 2005.

link information, manages sensor source data, browses sensor data, keeps a security and processes interoperation among other systems.

2.1.4 Sensor data management system

Sensor data management system processes a function that integrates, refines, stores and manages sensor data obtained from different sensors. this data can transfer XML data for interoperating among other systems.

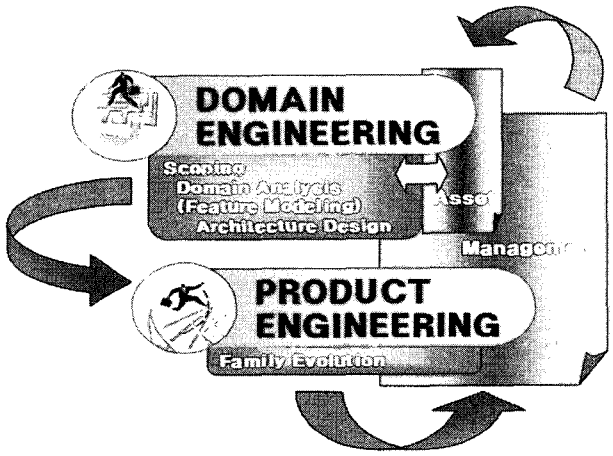
2.1.5 Sensor stream management system

Sensor stream management system processes a stream data obtained from SDP and provides function that manages a schema, parses continuous query, schedules jobs.

2.2 PLD (Product-Line Development)

The central theme of product lines is the development and reuse of assets of various kinds, including requirements, components, architecture and processes. These assets arise from analysis and identification of the commonalities within the product line. Reuse of assets leads to economies in both cost and time-to-market for product line development as compared with independently developed products.

Product line software engineering is an emerging software engineering paradigm, which guides organizations toward the development of products from core assets rather than the development of products one by one from scratch. PLSE consist of three major activities which are core asset development, product development using core assets and management. [Figure 2] shows this activities.



[Figure 2] Activities of PLSE

The paradigm of developing core assets for application development has been called domain engineering, in which an emphasis is given to the identification and development of reusable assets from an application "domain" perspective. Product line software engineering is similar to domain engineering in that they both attempt to exploit commonalities to build reusable core assets. However, PLSE differs from DE in that PLSE is founded on marketing. In PLSE, a product plan that specifies target products and their features from a market analysis is the primary input. Fielding products with features that the market demands in a timely manner and then evolving those products as the market evolves is the major driving force in the asset development in PLSE. Therefore, the scope of analysis and development in PLSE can be narrower and more focused than in DE. However, most engineering techniques used in DE can be applied in PLSE as both paradigms attempt to build flexibility and reusability into core assets. In order to develop reusable core assets for a product line, PLSE must have an ability to exploit commonality and manage variability. In this paper we use IDEF0 model as a method of domain analysis.

2.3 IDEF0 Model

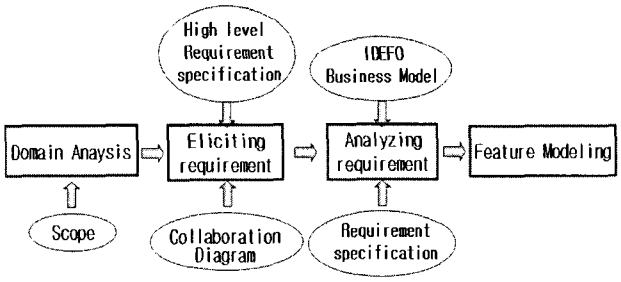
Domain analysis is first introduced by neighbors to denote studying the problem domain of a family of applications. Domain analysis is associated with reuse and a systematic approach for identifying the commonalties, similarities, and variabilities necessary to characterize and standardize a product line as a domain. Thus it is a key requirement for the creation of reusable core assets[2].

IDEF0 is a technique for producing a function model of a new or existing system or subject area. The modelling elements of IDEF0 are boxes and arrows. Boxes represent functions defined as activities, processes or transformations and arrows represent data or objects related to functions. A box describes what happens in a designated function. In IDEF0 a box as attached to it four types of arrows : inputs, outputs and mechanism

3. Approaching of PLD for MSDFS

3.1 Strategy of approaching

A procedure of the requirement analysis is domain analysis composing the scope of MSDFS, eliciting requirement using high-level requirement specification and collaboration diagram, analyzing requirement using IDEF0 model and requirement specifications, feature modeling phase. [Figure 3] shows this procedure.

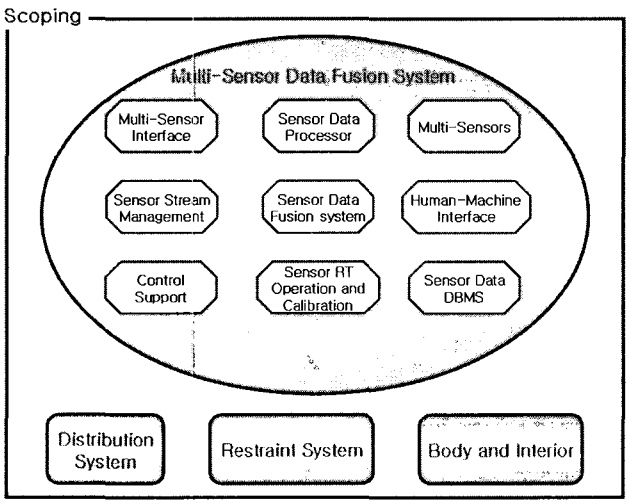


[Figure 3] Strategy of approaching PLD

3.2 Domain analysis phase

One of the major goals of scoping is to define the boundaries of the product line domain. In our context, the objective

of performing a scoping phase is to identify the business, organizational, technical, and legal requirements and constraints that are characteristic to MSDFS. The inputs for scoping are descriptions of Army AADS(Automatic Air Defense System), conference data related data fusion. As a result of this activity, we obtained an initial understanding of the functional scope for the MSDFS product line, as illustrated in [Figure 4].



[Figure 4] Scope of MSDFS

3.3 Eliciting requirements phase

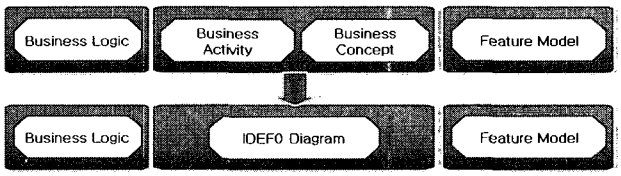
In order to elicit requirements we draw up a collaboration diagram and high-level requirement specification. These data provides basic data of requirement specifications and IDEF0 model. [Figure 5], collaboration diagram, shows procedure of processing sensor data of MSDFS and [Table 1], high level requirements, provides various views of developing MSDFS and uses to detail user requirement in analyzing requirements phase.

[Table 1] HLRS of MSDFS

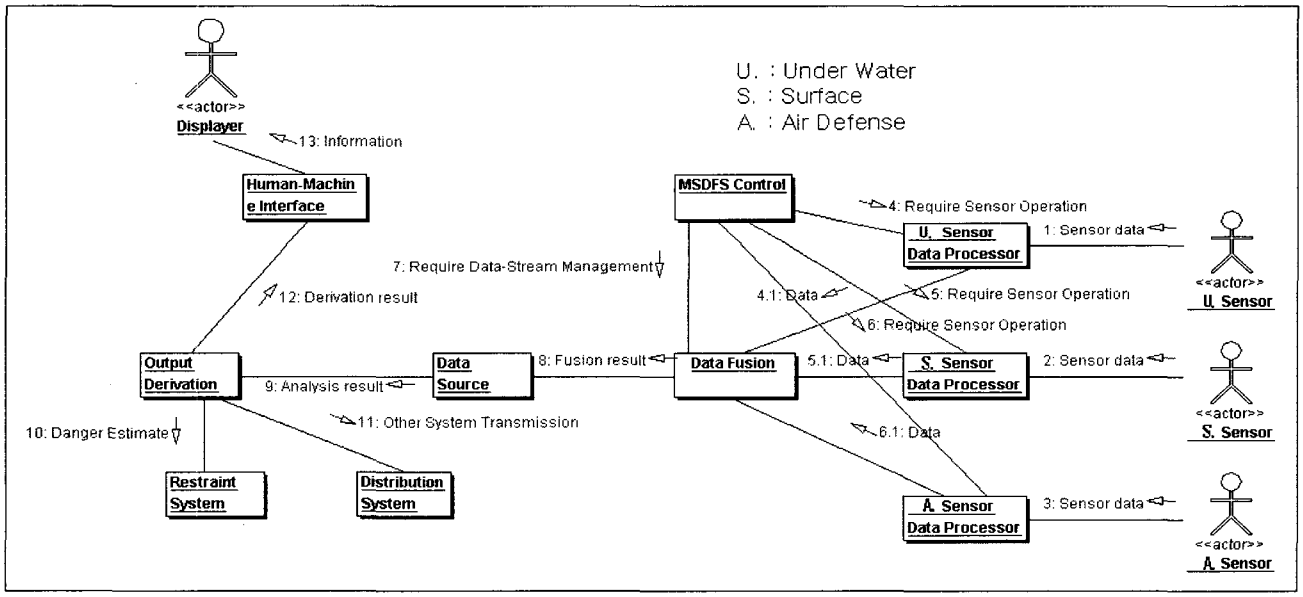
R1:	Sensor data processor shall integrate a linkage of different kind sensor through Network Interface.
R2:	Integrated track data shall provides a clear distinction between single and multi track
R3:	User shall manufacture the integrated track data through calibrating and refining
R4:	Fusion data stored repository shall be provided other system for maintaining condition and transferring data
R5:	Sensor data processor shall ensure hard-realtime and automation
R6:	Interface of MSDFS shall ensure extendability of different kind sensor
R7:	MSDFS shall provides function of TEC(Threat Evaluation Control) and WAC(Weapon Allocation Control)
R8:	Artifacts of MSDFS, architectures and components shall be reusable from developing similar system
R9:	Requirements of stakeholder must be satisfied in developing MSDFS
R10:	The policy and protocol in processing transaction must be worked out

3.4 Analyzing requirements phase

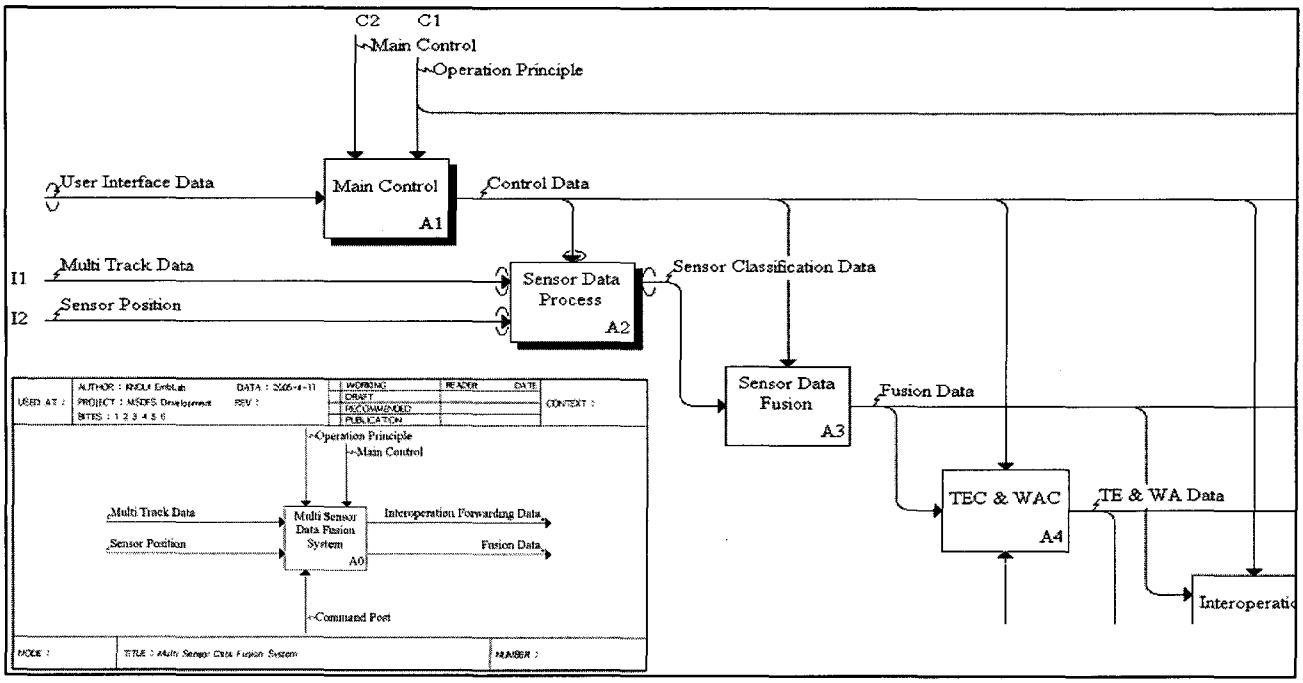
We perform a business modeling for analyzing requirements. Business modeling is process that define business logic and flow for understanding the performance of MSDFS and finally draw feature model. In this paper we apply IDEF0 model as a method of business model. Because IDEF0 model provides all views of system through upper and lower diagram. Normally business model applies activity and E-R diagram but these are restricted within narrow system views by swim lanes. This activity is present to [Figure 6] and result of activity is [Figure 7].



[Figure 6] Business Model Flowchart



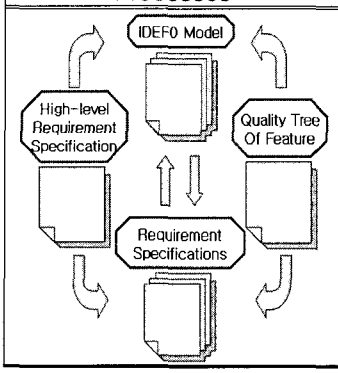
[Figure 5] Collaboration Diagram of MSDFS



[Figure 7] IDEF0 (A0) Diagram of MSDFS

In this paper we regulate a rule that factors of IDEF0 are used to analyze requirements and is shown on [Table 2] and [Table 3].

[Table 2] Requirement Analysis Processes

Requirement Analysis Processes	Contents
	• IDEF0 Diagrams ◦ Activities ◦ Concept (Input, Output, Control, Mechanism) ◦ Note • Requirement Specifications ◦ Explaining requirement

[Table 3] Mapping relation between IDEF0 and RS

IDEF0 Model	Requirement specification
Activity	head of requirements
Concept (input, output, mechanism)	head of functional requirements
Concept (control)	head of unfunctional requirements
Note	contents of requirements

[Table 2] shows process of requirement analysis that is recursive and [Table 3] shows mapping relation between IDEF0 model and Requirement specification. For example, Sensor data processor, activity of [Figure 7], has a concepts that are multi track data, sensor position, classification data, control data. This activity is effected by upper activity, main control, through control data. Thus we obtained a requirement that sensor stream processor processes sequentially different kind data with main control as facade controller of MSDFS. In this pattern we draw a [Table 4], requirement specification. The boxed related requirement of [Table 4] shows that input data (A : Activity, I : Input, O : Output) became a subject or content of scenario.

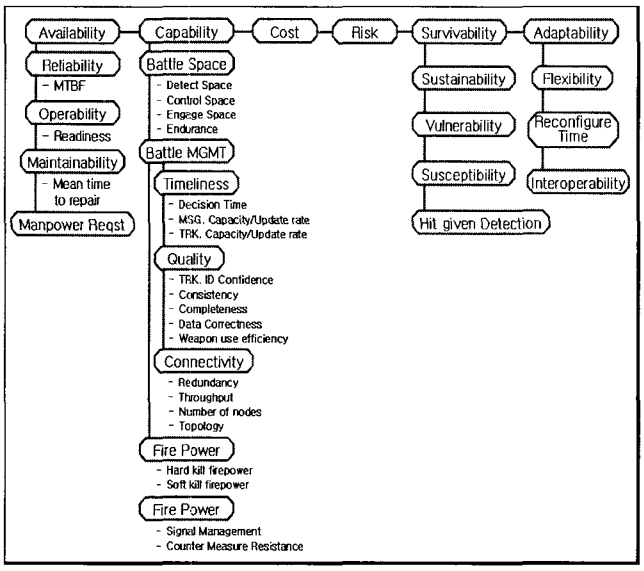
[Table 4] Requirement specification

name of requirement		ID	Class	related requirement	Scenario
Realtime of identifying		RF_5	Fun	R6	hall insure trackability for fusion data through All view antenna
expandability		RN_6	Unfun	R6,O1	At adding sensors(multi and other) perform smoothly extendability through reusing N-interface
Track Fusion	Data Fusion	RF_7	Fun	A3,I2,I3,O1	use data apply to standard format and normalize as 32-bit data format concerning extendability
	Data Stream Process	RF_8	Fun	A2,I2,I3,O1	erve a hokling function of processing tream data through standard data recieved ate and type of data is track signal, state, lass, speed
	Threat Evaluation	RF_9	Fun	A4,O1	erve a function of TEC, Common peration map
	Display	RF_10	Fun	A6,O1,I2,I3,I4	erve a function of LSD, distributed display 3-D display and knowing a states in real ime
	DBMS	RF_11	Fun	A7,O1,I5,I6	erver a function of limitation of DB ransaction and network registration

And then we drew a quality tree through a detailed Requirement specification. This quality tree is requirements

that describe how certain quality attributes such as reliability, operability, adaptability shall be satisfied by a system. This quality will be used to evaluate the feature model and architecture.

For each quality requirement we noted the specific goals of how it could be achieved in the MSDFS context as well as existing constraints and associated functional requirements. In [Figure 8], goals are represented by a boxed text "( )", whereas the non-boxed text are provided for constraints and related functional requirements, respectively. Note that we applied the approach described in to specify and detail the MSDFS quality requirements.

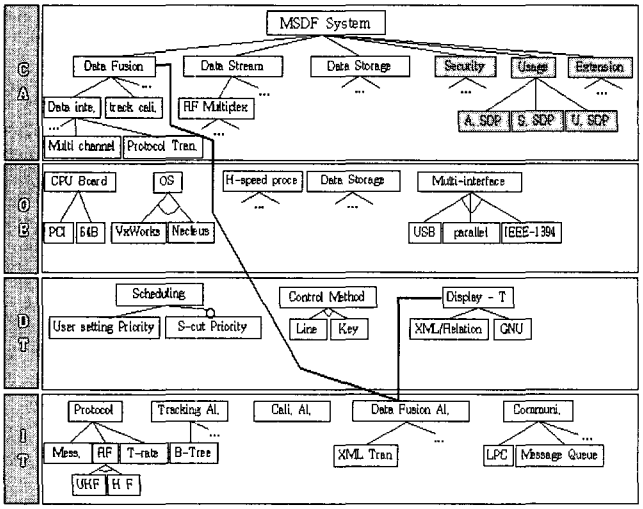


[Figure 8] Quality tree of MSDFS

3.5 Feature Modeling

Features include characteristic of system, functional and non-functional requirement, quality, capacity. Thus feature model, as core result of analyzing requirement of PL, provides multi-views of system and identifies easily commonalities and variabilities. And user can understand easily system. developers can make a various products. This feature model of [Figure 9] was drawn through procedures that elicit, analyze, refine, group and classify requirements of MSDFS.

This model supports to draw the conceptual architectures through the location of 'optional' and 'alternative' features. Because the location of features determines a number of subsystem.



[Figure 9] Feature model of MSDFS

4. Validation

The feature model should be validated before use. In this paper we have proposed a domain analysis method based on IDEF0 model and the result of comparing methods is showed below, [Table 5]. We decide a six factors of evaluation (e.g., All view, Feature identification of initial phase, Rationale for RS, Effort of initial phase, Readability, Connectivity between artifacts). The methods well known recently draw context model, structure model, entity relationship model and etc but our proposed method draw a scope model, collaboration diagram, IDEF0 model. In explanation of artifacts of each methods context model as simple architectural model decides the boundaries of system and dependency on its environment. Normally producing a context model is the first step in this activity. Structure model shows an overview of the system organization and composed of sub systems and the inter connections between sub systems. Entity relationship models have been widely used in database design.

[Table 5] Comparison of methods

Class	FODA	FORM	Proposed method
Artifacts of Context analysis	◦ Context Model ◦ Structure Model ◦ Entity Relationship Model	◦ Context Model ◦ Structure Model	◦ Scope ◦ Collaboration Diagram ◦ IDEF0 Model
All view	Yes	Yes	Yes
Feature Identification of initial phase	Mid	Weak	Strong
Rationale for RS	Indirect	Indirect	Direct
Effort of initial phase	Mid	Mid	Strong
Readability	Mid	Mid	Strong
Connectivity between Artifacts	Mid	Mid	Strong

In this result a proposed method can identifies easily features in initial phase because the concepts of an IDEF0 support directly to identify features. And user and developer can understand and connect efficiently between artifacts through rationale between requirement specification and concepts of an IDEF0. However this method is effortful rather than other method at initial phase of requirement analysis.

5. CONCLUSION AND FUTURE WORK

In this paper we proposed a specialized method that apply business model based on an IDEF0 model. This approach is to apply for an IDEF0 model for analyzing domain requirements instead of data model of other methods. The concepts

of an IDEF0 model apply directly to requirement specification. This method gives many advantages of all view of system, feature identification of initial phase, rationale for RS, readability and connectivity between artifacts. However this method is effortful rather than other method at initial phase of requirement analysis.

In the future we have to provide a efficient feature model based on service. This approach is expected to optimize architectures and components of embedded system. In other words the centering around service gives a minimized load of system and increases a reusability of artifacts (e.g., architectures and components). Therefore our approach will further enhance advantages of PLD for developing embedded system.

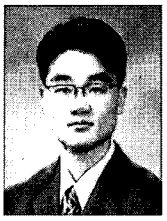
References

[1] Liliana Dobrica, Eila Niemela, "Attribute-based Product-Line Architecture Development for Embedded Systems," VTT Publications, 2000.

[2] Costin Badica, "A new formal IDEF-based Modeling," Proceeding of the First Balkan Conference in Informatic, BCI, 2003.

[3] Martin O.Hofmann, "Multi-Sensor Track Classification in Rotorcraft Pilot's Associate Data Fusion," Lockheed Martin Advanced Technology, 53rd AHSF, 1997.

[4] Steffen Thiel, Stefan Ferber, Thomas Fischer, "A Case Study in Applying a Product Line Approach for Car Periphery Supervision Systems," Proceeding of In-Vehicle Software 2001.



Ki sam Hong

Mar. 2000 : B.S. from Dept. of Civil Eng., Korea Military Academy.

Jan. 2004-Current : M.S. from Dept. of Computer & Information Science, Korea National Defense University.

Phone : +82-2-300-2138
Fax : +82-2-309-8118
E-mail : hong713@kndu.ac.kr

Heebyung Yoon

Mar. 1983 : B.S. from Dept. of Operational Research, Korea Naval Academy.

Feb. 1986 : B.S. from Dept. of Electronics Engineering, Yonsei Univ.

Jun. 1991 : M.S. from Dept. of Computer Engineering, Naval Postgraduate School.

Jul. 1998 : Ph.D. from Dept. of Computer Engineering, Georgia Institute of Technology.

Current : Professor of Korea National Defense University.

Phone : +82-2-300-2138
Fax : +82-2-309-8118
E-mail : hbyoon@kndu.ac.kr