

# Domain Analysis and Scoping

Wadim Schleicher  
Universität Stuttgart  
Universitätsstr. 38  
D-70565 Stuttgart  
[schleiw@stud.informatik.uni-stuttgart.de](mailto:schleiw@stud.informatik.uni-stuttgart.de)

## Zusammenfassung

Das Produktlinienengineering (PLE) hat das Ziel, die Einrichtung der Produktlinien im Bereich der Softwareentwicklung zu systematisieren und entsprechende Methoden bereitzustellen. Diese Methoden umfassen zahlreiche Aktivitäten. Zwei wichtige, die oft auch als die wichtigsten angesehen werden, sind Domänenanalyse und Scoping. Diese werden in dieser Arbeit kurz vorgestellt.

## 1. Einführung

Die aus der herkömmlichen Produktionsindustrie bekannte Produktlinientechnologie gewinnt im Bereich der Entwicklung von softwareintensiven Systemen immer mehr an Bedeutung und wird erfolgreich in der Praxis umgesetzt [1].

Sie stellt eine weitere, vielversprechende Entwicklung in der Reihe der Bestrebungen, die Zeit der Entwicklung von einzelnen Produkten durch hohen Grad an Wiederverwendung zu verkürzen, die Qualität der Endprodukte zu steigern und die Organisation des Entwicklungsprozesses durch seine Strukturierung effizient zu gestalten, dar.

Dem Produktlinienansatz liegt die Erkenntnis zugrunde, dass die Organisationen, die softwareintensive Systeme entwickeln, sich oft sowohl auf die Bereiche der Aufgaben, die ihre Produkte bewältigen sollen, wie auch auf die Methoden und Technologien, die diese Organisationen anwenden, weitreichend spezialisieren[2].

Aufgrund dieser Tatsache ist man bestrebt, die Wiederverwendung auf die Ebene der einzelnen Produktkomponenten, der sogenannten Basiseinheiten (Core Assets), zu verlagern.

In der Praxis stellte sich aber heraus, dass die Herauslösung der Gemeinsamkeiten und die geeignete Gestaltung solcher Komponenten eine komplizierte Aufgabe ist, die nur durch eine Vielzahl einzelner methodisch durchzuführender Schritte zu bewältigen ist. In der Abbildung 1 sind diese Schritte schematisch

dargestellt. Sämtliche Aktivitäten werden zu zwei Phasen, Domänenengineering und Applikationsengineering zusammengefasst. Domänenengineering besteht dabei aus den Aktivitäten zur Domänenanalyse, Architekturentwicklung und der Domänenimplementierung. Das Ergebnis dieser Aktivitäten ist eine Produktlinien-Infrastruktur/Plattform. Darauf aufbauend wird in der Phase des Applikationsengineerings das spezielle Produkt der Produktlinie entwickelt. Applikationsengineering besteht aus den Aktivitäten zur Systemanalyse, zum Systemdesign und zur Systemimplementierung. Mittels Scoping werden die Grenzen für die Domäne und das System festgelegt.

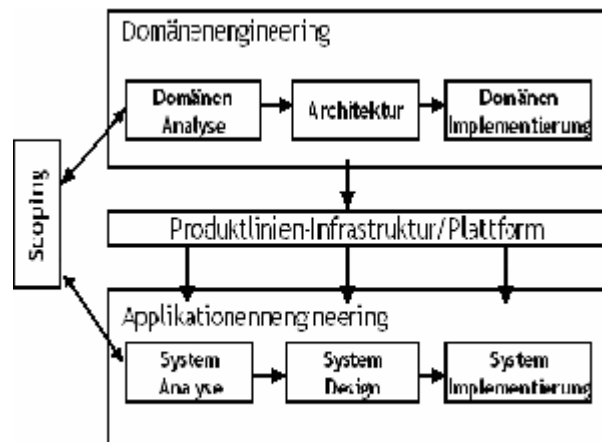


Abb.1 Phasen und Aktivitäten des Produktlinienengineerings.

Der Schlüssel zum Erfolg und Lebensfähigkeit einer Produktlinie liegt vor allem in einer gründlichen und aufwändigen Analyse des Anwendungs- und Lösungsbereichs und der richtigen Zusammensetzung der zu entwickelnden Produkte. Dies geschieht im Rahmen der Domänenanalyse und Scopings.

## 2. Domänenanalyse

Unter Domänenanalyse (Domain Analysis) im Sinne des PLE versteht man eine systematische Untersuchung eines Bereichs auf Zusammenhänge und Abläufe mit dem Ziel der Gewinnung der wiederverwendbaren Aspekte. Arango definiert Domänenanalyse wie folgt:

„Domänenanalyse ist ein Prozess zur Identifikation, Akquisition und Evaluation von wiederverwendbaren Informationen, die im Zuge der Entwicklung von Systemen für Klassen von Applikationen oder Problem domänen wiederverwendbar sind.“ [3]

Dabei wird eine Domäne als eine Menge von Konzepten und Technologien in einem Wissensbereich verstanden.[4]

Dazu gehören auch domänenspezifische Probleme, Lösungen und Begriffe.

Die Besonderheit einer Domäne einer Produktlinie ist die Tatsache, dass diese sich je nach Aufgabenstellung und Lösungsansätzen auf unterschiedliche Art aus den Teilbereichen der anderen Domänen zusammensetzt. Aus diesem Grund müssen zum einen die Grenzen einer Domäne für eine neue Produktlinie auch neu lokalisiert bzw. definiert werden und zum anderen muss die Relevanz der Informationen neu bewertet werden.

Das Ziel einer Domänenanalyse ist die Erstellung eines Domänenmodells, das die entsprechenden Informationen enthält und in einer geeigneten Form darstellt. In der Praxis stellt ein Domänenmodell eine Menge von Dokumenten dar, die das extrahierte Wissen aus unterschiedlichen Sichten repräsentieren. Zum einen dient das Modell als Grundlage für die Entwicklung der Softwarelösungen, zum anderen erfüllt es denselben Zweck, wie übliche Dokumentation.

Es erleichtert mittels eines Begriffslexikons die Kommunikation, sorgt für Klarheit in den Diskussionen, erleichtert Kundengespräche und präzisiert die Aufgabenstellungen innerhalb der Organisation. Mit anderen Worten, es schränkt den Freiraum für Missverständnisse und Unklarheiten ein. Außerdem erleichtert ein Domänenmodell die Einarbeitung neuer Mitarbeiter und nicht zuletzt dient es als eine Art „Versicherung“ gegen den Verlust der wichtigen Informationen z.B. durch Aussteigen einiger Mitarbeiter.

In der Vorgehensweise zur Erstellung eines Domänenmodells unterscheidet man in der Regel drei Phasen:

- **Kontextanalyse.** In der Phase der Kontextanalyse werden die Grenzen und der Umfang einer Domäne analysiert.

Außerdem werden die Beziehungen der Domäne zu den benachbarten und überstehenden Domänen und den Objekten, die außerhalb der Domäne stehen, identifiziert und beschrieben. Dazu gehören auch Datenaustausch, Beschränkungen und Informationsflüsse.

- **Domänenmodellierung.** In der zweiten Phase untersucht man die Beschaffenheit einer Domäne. Ihre innere Struktur wird unter unterschiedlichen Aspekten betrachtet und dokumentiert. Diese Phase ist auch der Kern der ganzen Domänenanalyse. Wie auch schon der Name andeutet, entsteht in dieser Phase das eigentliche Modell, das später als Grundlage für die Strukturierung der Produktlinie dient.
- **Architekturmodellierung.** In dieser Phase werden die Softwarelösungen für die zu lösenden Probleme aufgrund der Ergebnisse der Domänenmodellierung entworfen.

Diese Phasen werden im Idealfall nacheinander durchgeführt, da jede darauffolgende auf den Ergebnissen der vorhergehenden aufbaut. In der Praxis tauchen jedoch immer wieder Fragen auf, die man in der vorausgehenden Phase nicht berücksichtigt hat, so dass man immer wieder den Schritt zurück machen muss.

Die wichtigste Voraussetzung für die erfolgreiche Erstellung eines korrekten Modells ist fundiertes Wissen in den Bereichen, die eine Domäne umfasst. Falls dieses nicht im hinreichenden Umfang innerhalb der Organisation vorhanden ist, ist es notwendig nach geeigneten Informationsquellen zu suchen.

Dabei müssen verschiedene Informationsquellen in Betracht gezogen werden, wie zum Beispiel:

- Technische Literatur
- Kundengespräche
- Marktanalysen
- Die vom Auftraggeber bereitgestellte Kontaktpersonen
- Externe Experte (Berater)

Alle diese Quellen haben je nach Situation ihre Vor- und Nachteile. Oft sind die gegenwärtige Gegebenheiten (wie Verfügbarkeit von Ressourcen) maßgebend dafür, auf welche Quellen und in welchem Umfang zugegriffen wird.

Die gründliche Analyse einer Domäne ist essentiell für die Langlebigkeit einer Produktlinie, daher sind einige Risiken zu beachten.

**Analysisparalyse** – das Steckenbleiben im Prozess der Analyse, wobei immer detailliertere Modelle entstehen, ohne dass diese relevante Informationen erzeugen.

**Unzureichende Dokumentation** – falls die Ergebnisse einer Domänenanalyse nicht ausreichend dokumentiert sind, läuft man Gefahr, dass wichtige Informationen verloren gehen. Unzureichende Dokumentation gibt viel Freiraum für Irrtümer und Missverständnisse und kann zur Entstehung fehlerhafter Lösungen führen, außerdem erschwert dies die Kommunikation innerhalb der Organisation und schafft zusätzlichen Klärungsbedarf.

**Fehlendes Expertenwissen** kann dazu führen, dass das Domänenverständnis unzureichend bleibt und das dabei entstehende Modell unbrauchbar wird.

Außer den obengenannten Risiken gibt es eine Reihe von Schwierigkeiten. Es gibt zum Beispiel keinen Standard, was ein Domänenmodell enthalten muss. Es gibt auch keine festgelegten Regeln dafür, wie die entsprechenden Transformationen der Informationen ins Modell vorgenommen werden sollten, wie es auch keine Regeln dafür gibt, welche Informationen als relevant zu bewerten sind. Es gibt auch keine einheitliche Vorgehensweise, wie der Prozess der Domänenanalyse konkret durchgeführt werden muss. Stattdessen gibt es eine Reihe von Ansätzen. Auf einen von ihnen: Features Oriented Domain Analysis wird im nächsten Kapitel näher eingegangen.

### 3. FODA

Features Oriented Domain Analysis (FODA) wurde am Software Engineering Institut der Carnegie Mellon University(SEI/CMU) entwickelt. Dieser Ansatz entstand bereits im Jahre 1990 und zählt heutzutage zu einem der ausgereiftesten und verbreitetsten Ansätze auf dem Gebiet des PLEs.

FODA beschreibt die wichtigsten Eigenschaften (Features) von einer Klasse von Anwendungen aus der Sicht des Kunden oder Endbenutzers. Dabei versteht man unter einem Feature ein bestimmtes Verhalten des Systems, das durch Anforderungen spezifiziert wird. Diese Eigenschaften sind in FODA sehr breit definiert und können je nach Kontext funktionale oder auch nicht-funktionale Eigenschaften sein.

In der Analyse nach FODA werden die bereits vorgestellten drei Phasen der Domänenanalyse durchlaufen. Dabei werden die einzelnen Phasen in zahlreiche Unteraktivitäten aufgeteilt. Das Ergebnis jeder dieser Unteraktivitäten sind Dokumente oder Teilmodelle, die man der Abbildung 2 entnehmen kann:

Phase	Aktivität	Ergebnis	Darstellung
<i>Kontext Analyse</i>	Kontext-analyse	Kontextmodell	Datenfluss-diagramm
		Strukturmodell	Block-diagramm
<i>Domänenmodellierung</i>	Features Analyse	Features Modell	UND/ODER Baum Liste der Features
	Information s-analyse	Informationsmodell	Entity-Relationship Diagramm
	Funktionale Analyse	Datenflussmodell	Datenflussdiagramm
Zustandsübergangmodell		Zustandsübergangdiagramm	
<i>Architekturmodellierung</i>	Architekturentwurf	Architekturmodell	Block-diagramm

Abbildung 2. Zusammenfassung der in FODA definierten Phasen, Aktivitäten, Modelle und ihre Repräsentation.

In der Phase der Kontextanalyse werden die Grenzen einer Domäne festgestellt und ihr Verhalten gegenüber den benachbarten oder übergeordneten Domänen, wie auch gegenüber den Objekten, die außerhalb der Grenzen der Domäne liegen, beschrieben. Das Ziel ist einerseits, die Umgebung einer Domäne festzuhalten und zu dokumentieren, andererseits Datenflüsse und Informationsaustausch zu untersuchen. Aus diesen beiden Sichten ausgehend, unterteilt man in FODA die erste Phase in **Struktur- und Kontextanalyse** mit daraus hervorgehenden Teilmodellen in Form von Struktur- und Kontextdiagrammen.

In der Phase der Domänenmodellierung entstehen umfangreiche Modelle, die im Laufe der einzelnen in FODA definierten Aktivitäten entstehen und die Beschaffenheit einer Domäne beschreiben :

Das Entity-Relationship Modell ist das Ergebnis der sogenannten **Informationsanalyse**, bei der die Objekte und Relationen innerhalb einer Domäne beschrieben werden. Dies geschieht in der domänenspezifischen Sprache und dient der Veranschaulichung und Analyse der inneren Beschaffenheit des untersuchten Problembereichs.

Das funktionale Modell entsteht im Laufe der **funktionalen Analyse**, die sowohl die Datenbewegungen innerhalb der Domäne, als auch Zustandsänderungen der Objekte berücksichtigt. Das Funktionale Modell beinhaltet daher sowohl Datenflussdiagramme, als auch Zustandsübergangdiagramme.

Die obengenannten Techniken und Aktivitäten sind auch aus anderen Bereichen der SWT bekannt und werden intensiv eingesetzt. Die Besonderheit der FODA Methode ist das Featuremodell, das die Gesamtheit der Eigenschaften einer Klasse von

Produkten umfasst und im Laufe einer **Featuresanalyse** entsteht. Im folgenden werden unterschiedlichen Arten von Features kurz vorgestellt:

**Pflichtfeatures** – Eine Menge der Eigenschaften, die jedes Produkt der Klasse vorweisen muss.

**Wahlpflichtfeatures** – Eine Menge der Eigenschaften, aus denen das Produkt eine vorweisen muss.

**Wahlfeatures** – Eine Menge der Eigenschaften, die einzelne Produkte optional vorweisen können.

Diese Eigenschaften werden in einer Featuresliste beschrieben und ihre Abhängigkeit in einem UND/ODER – Baum dokumentiert. Ein einfaches Beispiel, zum Modellieren einer Autosteuerungssoftware in Abb. 3 veranschaulicht die Idee.

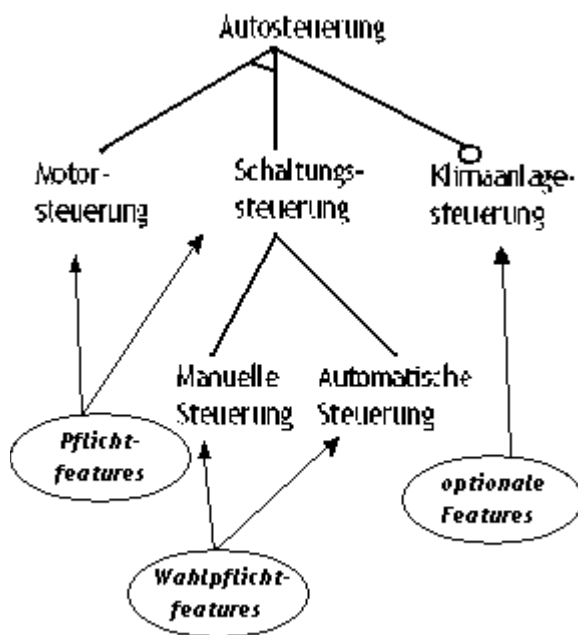


Abb. 3 FODA spezifischer Featurebaum für die Autosteuerungssoftware.

Der Ansatz ermöglicht die produktspezifischen Eigenschaften durch Verfeinerung zu definieren und Gemeinsamkeiten durch Abstraktion herauszulösen. Wie in Abb. 3 zu sehen ist, braucht jedes Fahrzeug die Steuerung des Motors und der Schaltung, wobei dies bei einem einzelnen Fahrzeug entweder Handschaltung oder automatische Schaltung sein wird. Die Klimaanlagesteuerung steht hingegen zur Wahl, so dass sie in einer konkreten Realisierung die Steuerungssoftware wahlweise vorhanden oder nicht vorhanden sein kann.

Diese Struktur zusammen mit einer Liste der Features, die ebenfalls ein Ergebnis der

Featuresanalyse ist, ermöglicht eine übersichtliche Beschreibung der Produkte und erlaubt die Komponenten in Hinsicht auf einzelne produktspezifischen Eigenschaften einzurichten.

FODA liefert als Ergebnis ein umfassendes und detailliertes Domänenmodell, in dem eine ganze Klasse von Produkten beschrieben wird. Sie gibt die Möglichkeit, die spezifischen Komponenten durch Verfeinerung zu entwickeln, liefert eine gemeinsame Funktionalität der Komponenten und unterstützt die Wiederverwendung auf der funktionalen Ebene.

Auf die dritte Phase der Architekturmodellierung wird im Rahmen dieser Arbeit nicht näher eingegangen. Der Grund dafür ist der, dass FODA nur relativ schwach den Architekturentwurf und Implementierung unterstützt. FODA ist eine herausragende und bewährte Methode zur umfassenden Domänenanalyse. Darin liegt ihr Schwerpunkt.

#### 4. Scoping

Eine weitere essentielle Aktivität des PLE ist Scoping, das nach Clements wie folgt definiert ist:

„Scoping ist eine Aktivität, die ein System oder eine Klasse von Systemen durch Festlegung der Eigenschaften oder Aspekten, die „enthalten“ oder „nicht enthalten“ sind, einschränkt.“ [2]

Diese auf den ersten Blick etwas ungenaue Definition ist bei der näheren Betrachtung sehr zutreffend und spiegelt die breite Präsenz des Scopings wieder, denn bereits mit Entstehung der Softwaretechnik beschäftigt man sich mit Scoping. Dies liegt in der Natur der SWT, wo man permanent an der Modellbildung arbeitet und Entscheidungen treffen muss, welche der Eigenschaften, der zu modellierenden Objekte, ins Modell übernommen werden. Diese werden ihrerseits im Wesentlichen von dem Zweck, zu dem das Modell gebildet wird, bestimmt. Diese Aktivität wird oft als eine inoffizielle Aktivität, als Teil der Anforderungsspezifikation, durchgeführt[7].

Im Bereich der Produktlinien Technologie wird der Begriff des Scopings oft in etwas engerem Sinn, als Gesamtheit der zu entwickelnden oder geplanten Produkte und ihren Eigenschaften, verwendet. Dabei spricht man vom „Product Line Scope“.

Dies liegt daran, dass das Ziel des PLE ist, ein Konstrukt zu bauen, das es erlaubt, die geplanten Produkte zu realisieren und die Fähigkeit besitzt, die Veränderungen an den geplanten Produkten, die in der Realität oft auftreten, wie auch weitere noch nicht geplante Produkte, aufnehmen zu können. Dabei geht man zum einen von den geplanten Produkten, zum anderen von den Produkten, die im Anwendungsbereich in der Zukunft realisiert werden können, aus.

Diese beiden Seiten bestimmen auch zwei unterschiedliche Vorgehensweisen beim Scoping der Produkte:

Eine Möglichkeit ist, von den geplanten Produkten auszugehen und, je nach Veränderungen zum Beispiel auf dem Markt oder auch im Bereich der Technologien, die die Veränderungen in den neuen Produkten erzwingen, den Scope immer wieder anzupassen. Entsprechende Analysen und schließlich Anpassungen und Veränderungen im ganzen Konstrukt der Produktlinie werden durchgeführt. Diese Vorgehensweise nennt man *reaktives Scoping*.

Eine andere Vorgehensweise, auf die z.B. FODA-Methode gerichtet ist, besteht darin, eine Domäne sorgfältig abzugrenzen, eine entsprechende sorgfältige Analyse durchzuführen und aufgrund der Ergebnisse eine Menge der möglichen Produkte festzulegen. Diese Vorgehensweise nennt man *proaktives Scoping*.

Die beiden Vorgehensweisen in reiner Form wird man in der Realität wohl kaum finden. Wenn man das reaktive Scoping betreibt, versucht man durch die Veränderungen auch die möglichen zukünftigen Veränderungen aufzufangen und umgekehrt, wenn man proaktives Scoping betreibt, berücksichtigt man in erster Linie die Produkte, die man in der nächsten Zeit entwickeln wird. Außerdem müssen durch die ständigen Veränderungen auf dem Markt und im Bereich der Technologien die Grenzen einer Domäne ständig neu angepasst werden.

Hier sieht man, dass Scoping eine permanent betriebene und durchgreifende Aktivität darstellt. Scoping wird in allen Phasen des PLE durchgeführt. Beinahe alle Strukturen einer Organisation sind von dieser Aktivität betroffen. Scoping ist eine kommunikationsintensive Aktivität. Das Management entscheidet welche Produkte zu entwickeln sind, die Domänenanalysten liefern entsprechende Modelle und Produktlinien-Ingenieure versuchen dazu, ein passendes Konstrukt zu entwerfen oder anzupassen. Dazu werden oft die Spezialisten der einzelnen technischen Bereiche hinzugezogen, um festzustellen, ob entsprechende technische Lösungen möglich und effizient verwirklicht werden können.

Umgekehrt kann es durchaus sein, dass im Rahmen einer bereits existierender PL es möglich ist, nur einen Teil der geplanten Produkte sinnvoll zu entwickeln. So kann es dazu führen, dass auf einige einzelne ursprünglich geplante Produkte oder ihre einzelne Eigenschaften verzichtet werden muss. Diese Entscheidungen sind nicht einfach. Maßgebend für sie sind wirtschaftliche Ziele und zur Verfügung stehende Ressourcen der Organisation. In jedem einzelnen Fall muss es aufgrund der gegenwärtigen Gegebenheiten entschieden werden, was u. U. zu schwerwiegenden Fehlentscheidungen aufgrund von ungenauen Abschätzungen führen kann.

Hier zeigt sich noch eine Eigenschaft des Scopings. Scoping im Rahmen einer PL ist entscheidungsintensiv.[5]

Aufgrund all dieser Eigenschaften wird auch klar, warum Scoping im Rahmen der PL-Technologie zu einer offiziellen und eigenständigen Aktivität erklärt wird.

In der Praxis sind zwei grundlegende Gefahren identifiziert worden:

Ein zu großer Scope, der Produkte mit nicht hinreichendem Anteil an Gemeinsamkeiten enthält, führt dazu, dass die Entwicklung der einzelnen Produkte nicht effizient durchgeführt werden kann. Da man in diesem Fall jedes Produkt nacheinander entwickelt und der Wiederverwendungsgrad auf ein Minimum schrumpft, ergibt sich durch den Produktlinienansatz kein Vorteil.

Ein zu kleiner Scope, der Produkte mit nur sehr wenig Varietät enthält, läuft Gefahr, eine nicht langlebige Produktlinie zur Folge zu haben. Diese kann dann nur mit großer Mühe und großem Aufwand neue Produkte aufnehmen.

## 5. Fazit

Domänenanalyse und Scoping sind zwei essentielle Aktivitäten des PLE, die die Schlüsselstellung in dem Prozess des Einrichtens einer Produktlinie annehmen. Auf den Ergebnissen dieser Aktivitäten bauen weitere Aktivitäten (wie z.B. Architekturentwurf) auf. Deswegen ist es wichtig, diese sorgfältig und gründlich durchzuführen. Es ist von großer Bedeutung für die Lebensfähigkeit einer Produktlinie, in die Durchführung dieser Aktivitäten hinreichend viel Ressourcen zu investieren. Diese Aktivitäten sind kompliziert, kommunikations- und entscheidungsintensiv. Sie sind mit vielen Risiken und Schwierigkeiten verbunden, da es noch keine ausgereiften und universellen Methoden dafür gibt. Jedoch gibt es in letzter Zeit zahlreiche Publikationen und Untersuchungen auf diesem Gebiet mit vielversprechenden Ergebnissen, die darauf schließen lassen, dass der Produktlinienansatz eine dynamische und vielversprechende Entwicklung im Bereich des industriellen Softwareherstellungsprozesses darstellt.

Einige der speziellen Techniken, wie z.B. FODA im Bereich der Domänenanalyse, haben sich bereits in der Praxis bewährt.

## 6. Literatur

- [1] Brownsword, L.; Clements, P. *A Case Study in Successful Product Line Development* (CMU/SEI-96-TR-016, ADA315802). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996.
- [2] Clements, P.; Northrop, L. *Software Product Lines: Practices and Patterns*. Boston, MA: Addison-Wesley, 2002.
- [3] Arango, G. Ch. 2, "Domain Analysis Methods," 17-49. *Software Reusability*. Hemel Hempstead, England: Ellis Horwood, 1994.
- [4] Bass, L.; Clements, P.; Donohoe, P.; McGregor, J.; Northrop, L. *Fourth Product Line Practice Workshop Report* (CMU/SEI-2000-TR-002, ADA375843). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000.
- [5] Kishi, T.; Noda, N.; Katayama, T. "A Method of Product Line Scoping Based on a Decision-Making Framework". Proceeding of the Second International Conference on Software Product Lines.(SPLC2). August 2002, pp. 348-365
- [6] Van Ommering, R.; Bosch, J. "Widening the Scope of Software Product Lines – From Variation to Composition". Proceeding of the Second International Conference on Software Product Lines.(SPLC2). August 2002, pp. 328-347.
- [7] Bosch, J. "Organizing for Software Product Lines," 117-134. *Proceedings of the 3rd International Workshop on Software Architectures for Product Families (IWSAPF-3)*. Las Palmas de Gran Canaria, Spain, March 15-17, 2000. Berlin, Germany: Springer, 2000.
- [8] Clemen, R. T. *Making Hard Decisions: An Introduction to Decision Analysis*. Boston, MA: PWS-Kent Publishing Co., 1991.
- [9] Kang, K. et al. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Technical report CMU/SEI-90-TR-021, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2000.