

## 品質要求駆動型のアーキテクチャ分析設計手法による WEB システム開発

繁在家 学\* 鷺崎 弘宜\*\* \*\*\*

ソフトウェア中心システムの開発における品質要求は、要求分析の段階で明確に識別し、さらにはソフトウェアのアーキテクチャ分析・設計において明示的に扱うことではじめて実現される。また、属人性の排除、トレーサビリティを考慮したアーキテクチャ分析・設計は、ソフトウェア工学の手法を適用する必要がある。しかし、各工程に対して複数の手法を適用する場合、手法間の整合性を考慮する必要があり、手法を適用する阻害要因となっている。上述の背景を踏まえ、本稿では、簡単な WEB システムの開発を事例として、複数の手法を用いた品質要求駆動型のアーキテクチャ分析・設計手法を提案する。具体的には、要求分析手法である KAOS、アーキテクチャ設計手法である品質特性駆動型設計(Attribute-Driven Design)の整合性を考慮し、独自拡張を加えたアーキテクチャ分析・設計手法を提案する。さらに、提案手法は、品質要求を考慮したアーキテクチャ分析・設計を達成する中で、アスペクト指向型アーキテクチャを抽出する技法を取り扱う。

### The development of the web system by the quality requirement driven method for the architecture analysis and design

HANZAIKE Manabu\* WASHIZAKI Hironori\*\* \*\*\*

Quality requirements for a development of a software centered system should be recognized in a process of requirements analysis and also those requirements are realized by dealt with explicitly in a software architecture analysis and design process. A method of software engineering should be used in the software architecture analysis and design process to clear individual techniques and realize traceability. However, if several methods are used in each process, consistency between methods should be considered. This is a disincentive to adapt methods into the development. With the background above, in this paper, we propose a quality requirement driven method of the architecture analysis and design process with several methods using an example of a simple web system development. Specifically, we propose it with unique expansion and considering consistency between KAOS, which is a method of requirements analysis, and ADD, which is a method of attribute-driven design. Furthermore, this method we propose includes a technique of picking up aspect oriented architecture during the architecture analysis and design process with satisfaction of quality requirements.

#### 1. はじめに

近年、ソフトウェアのアーキテクチャ(以下、アーキテクチャ)の重要性が叫ばれている。アーキテクチャとはコンポーネントとコンポーネントの間関係であり、アーキテクチャ分析・設計とは、これら

関係を定義することである[3]。しかし、アーキテクチャ分析・設計によるアーキテクチャの導出は、無数のアーキテクチャ候補の中から、ユーザがソフトウェアに求める機能(機能要求)、品質・コスト・納期といった機能以外の要求(非機能要求)を満足するよう、様々なアーキテクチャ上の決定を積み重ねて実現される。従って、アーキテクチャ

\* 日本アイ・ビー・エム株式会社, IBM Japan. Ltd.

\*\* 早稲田大学, Waseda University.

\*\*\* 国立情報学研究所 GRACE センター,  
National Institute of Informatics GRACE Center

分析・設計は、様々な要求を考慮する必要があり、同時に分析・設計者の豊富な経験が必要となる困難な作業といえる。

以下、アーキテクチャ分析・設計を取り巻く問題点を列挙する。

#### 問題1:品質要求の曖昧な定義

アーキテクチャ分析・設計は、機能要求、非機能要求を明示的に扱うことで実現される。ソフトウェアに求められる機能要求、コストや納期といった制約は、要求分析の段階からソフトウェアを利用するユーザの中心的関心事となりうる。これに対して、品質要求は、要求分析の段階においてユーザの中心的関心事にはならず曖昧に扱われる傾向がある。

#### 問題 2:属人的な分析・設計とトレーサビリティの欠如

アーキテクチャ分析・設計は、設計者に対して多くの経験を求めることから、属人的になる傾向がある。また、系統的な手法を適用しないアーキテクチャ分析・設計は、機能要求、非機能要求に対する修正・変更等に対してその波及を追跡することができず、アーキテクチャ上の影響箇所を特定することが困難となる。いわゆるトレーサビリティの欠如した分析・設計となる。

#### 問題 3:各手法間の整合性の必要

ソフトウェア工学の手法を各工程に対して複数適応する場合、手法間の整合性を考慮する必要があり、手法適用の阻害要因となっている。品質要求の分析手法からアーキテクチャ設計・評価へとつなげる取り組みはあるものの[10-12]、依然として効率的・効果的な設計・評価までの落とし込み方法など不明瞭な点がある。また、同一の要求分析の枠組みの中でアーキテクチャ設計・評価まで扱う取り組みが多数であり、各工程でそれぞれ異なる手法を既に導入している場合における整合性確保など課題が多い。

上述の問題を解決するには、①品質要求を要求分析の段階で取り扱う必要がある。また②ソフトウェア工学の手法を適応して、属人性を排除し、各工程全般に亘るトレーサビリティを確保する必要がある。さらに③ 各工程に対して複数のソフトウェア工学の手法を適応する場合、手法間の整合性を考慮する必要がある。

本稿では、こうした問題と解決を踏まえたアーキテクチャ分析・設計手法を提案する。提案にあたり、簡単な WEB システムの開発を事例として取り上げた。本事例に対して最適なソフトウェア工学の手法を抽出し、それら手法を用いた品質要求を駆動とするアーキテクチャ分析・設計手法を提案する。具体的には、要求分析手法として KAOS[1][7]、アーキテクチャ設計手法として品質特性駆動型設計(Attribute-Driven Design)[2][4]を取り上げ、整合性を考慮し、独自拡張を加えたアーキテクチャ分析・設計手法を提案する。また、品質要求を要求分析の段階で明示的に扱った。

さらに、提案手法は、品質要求を考慮したアーキテクチャ分析・設計を達成する中で、アスペクト指向型モジュールを抽出する技法を取り扱った。この技法は、要求分析手法である KAOS モデル上で適応可能であり、モデルの観察により、アスペクト指向型のアーキテクチャを抽出し、アーキテクチャの候補とすることが可能となる。

尚、本稿では、機能要求・非機能要求を以下の分類に基づいて扱った。

- 機能要求
- 非機能要求  
品質要求 (ISO9126 品質特性)  
制約 (コスト, 納期等)

非機能要求は、品質要求と制約の2つに分けて扱った。さらに、品質要求は、ISO9126 の品質特性に基づいて分類した。

以下、第 2 章では、適応事例として簡単な WEB システムの開発について述べる。第 3 章では、本稿で提案する手法について述べる。また、アスペクト指向型のアーキテクチャを抽出する技法について述べる。第 4 章では、提案手法に対する評価を行った。

## 2. 適用事例

本稿では、提案する手法を説明するにあたり、簡単な WEB システム開発を事例として取り上げた。図 2-1に、事例として取り上げた WEB システムの概要を示す。

図 2-1において、本開発の対象となるシステムは、リクエストのリダイレクトを行うシステムである。一般に、WEB ページは複雑にリンクが張り巡らされており、ある WEB ページの URL を変更した場合、当該の WEB ページを参照する多数の WEB ページのリンク元を変更する必要がある。本システムは、こうした問題に対処することを目的として

おり、PC 及び携帯電話端末(以下、携帯端末)からのリクエストを自動判別し、該当するページにリダイレクトする。これにより、管理対象の WEB ページ遷移先情報を一元管理することができる。また、誤ったリクエスト・アクセスを許可されていない PC 及び携帯端末からのリクエストは、エラーページにリダイレクトする。さらに、WEB システムは、不特定多数からのリクエストを処理する必要が求められ、24 時間・365 日の連続運用を実現する必要がある。また、短納期での開発を求められる。

以下に、本 WEB システムの開発に要求されるニーズを示す。

- PC 及び携帯の WEB ページ・リダイレクトシステムを実施する
- 管理対象 WEB ページの遷移先 URL 情報を一元管理する。これにより、遷移先 URL 情報の変更に対する影響をリンク先 WEB ページに与えない
- 誤ったリクエスト・アクセスが許可されていない PC 及び携帯端末からのリクエストは、エラーページにリダイレクトする
- 各々のリクエストを解析後、PC のブラウザ情報、携帯端末の機種情報を自動判別し、適切にリダイレクトする
- 不特定多数からのリクエストを処理
- 24 時間・365 日の運用実現
- 短納期での開発
- 遷移先 URL が変更された場合に、容易にシステムに反映できる

上で列挙したニーズは、「PC 及び携帯端末からのリクエストをリダイレクトする」等の機能要求、「24 時間・365 日の運用の実現」といった品質要求、「短納期での開発実現」といった制約を含む。こうしたニーズは、アーキテクチャ上の決定に大きく影響を与えるため、要求分析の中で明確に扱う必要がある。さらに、ニーズは手法の中で系統的に扱われ、アーキテクチャ分析・設計の入力として扱われる。

次章では、本稿で提案するアーキテクチャ分析・手法を説明する。

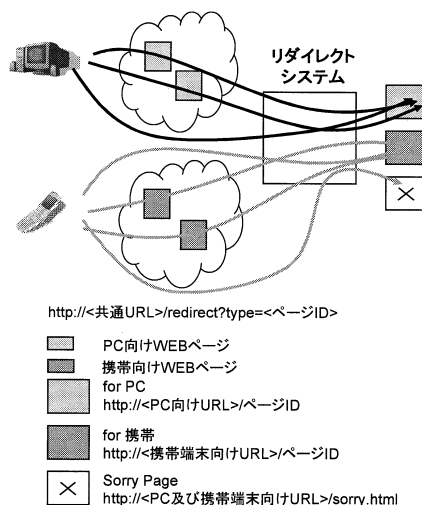


図 2-1. WEB システム

### 3. 品質要求駆動型のアーキテクチ分析・設計手法

本章では、本稿で提案する品質要求駆動型のアーキテクチャ分析・設計手法を説明する。図 3-1 に、提案手法の概要を示す。以下、図 3-1 の工程に沿って詳細を説明する。

#### 3.1. 品質シナリオ[5]

最初に、本手法では、2章で取り上げたニーズから、品質要求を具体的なストーリーで記述した品質シナリオを作成する。表 3-1に、品質シナリオの記述例を示す。

表 3-1 品質シナリオ記述例

品質特性	効率性
品質副特性	時間効率性
番号	A1
シナリオ	(環境) 通常稼働時に、(刺激) JMeter で高負荷をかけた場合、(応答) HTTPRequest を秒間 1000 トランザクションを処理できること

一般的に品質シナリオは、後述する品質特性駆動型設計手法において作成される。本提案手

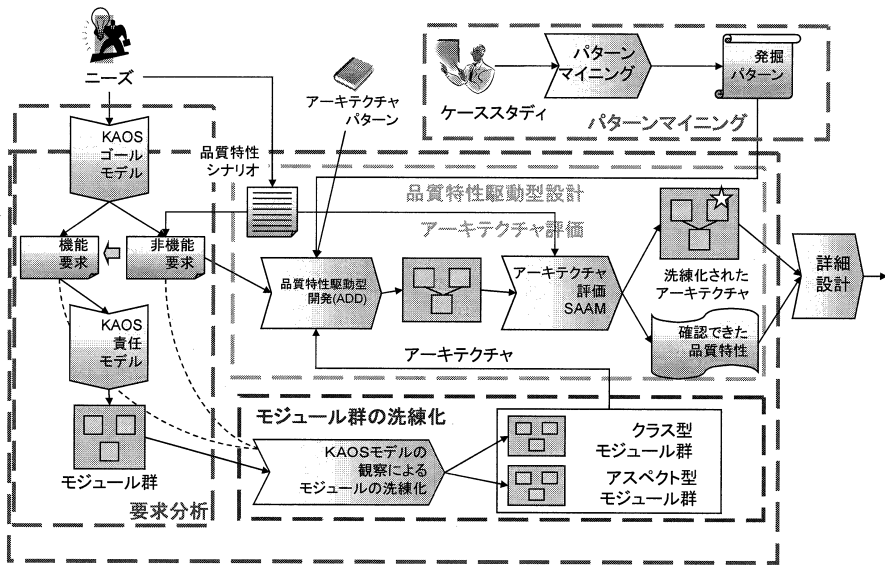


図 3-1 品質要求駆動型のアーキテクチャ分析・設計手法

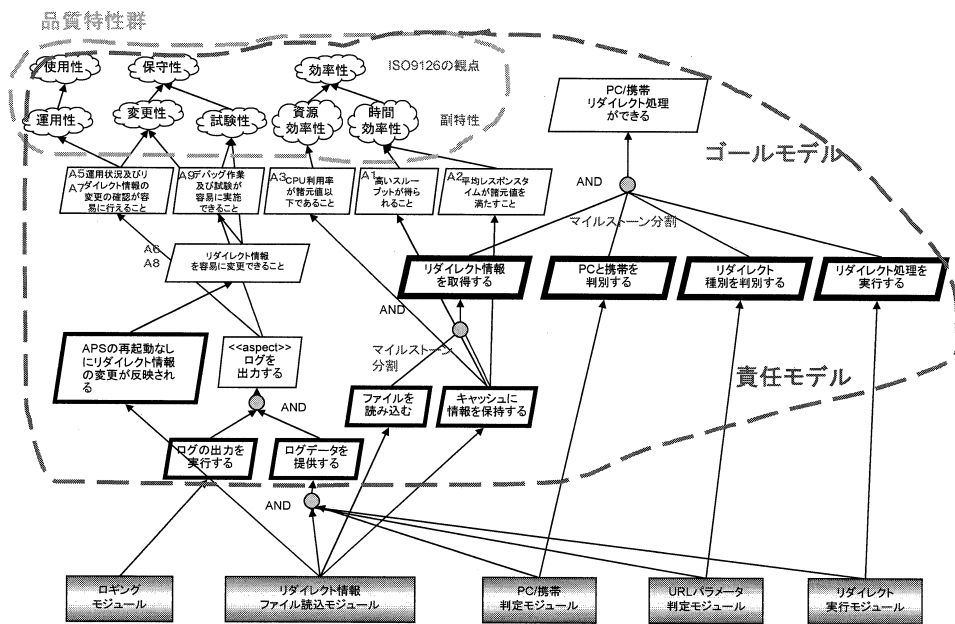


図 3-2 KAOS による要求分析モデル

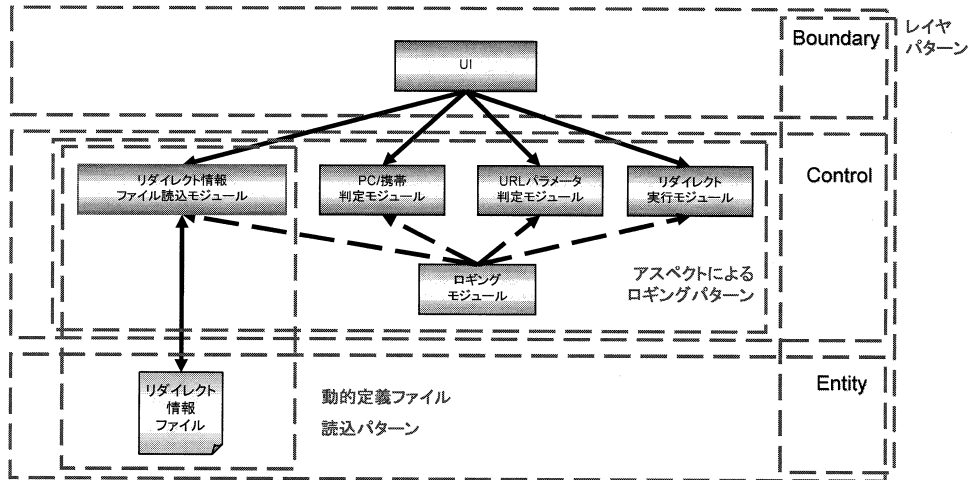


図 3-3 設計したアーキテクチャ法では、手法

間の整合性を考慮し、要求分析に品質シナリオを投入する。表 3-1において、品質要求は、刺激・応答・環境を含む具体的なストーリーに結び付けて記述される。これにより、本来曖昧になりがちな品質要求を具体的な状況を想定した形式で記述し、品質記述上の問題に対処した。

### 3.2. 要求分析

本事例では、機能に対するニーズが簡単であり、かつシステムの最終的な要求を系統的に扱う必要があることから、ゴール指向要求分析手法の適応を検討した。ゴール指向要求分析手法は、ステークホルダ間の利害関係を扱うことができる i\*[8]や、システムゴールを系統的に扱うことができる KAOS がある。本事例では、ステークホルダ間利害関係を考慮する必要がないことからKAOSを選択した。

図 3-2に、KAOS による要求分析モデルを示す。図 3-2において、平行四辺形はニーズあるいは要求を表し、太枠の平行四辺形は、KAOS ゴールモデルにおける最終的な要求を表す。雲形は、後述する品質特性及び品質副特性を表す。長方形は、KAOS 責任モデルにおけるエージェントを表し、本提案手法ではアーキテクチャを構成するモジュールの候補を表すものとした。

#### 3.2.1. 品質特性群と品質シナリオ

要求分析モデルにおいて、品質要求を扱う為

に、品質特性群と呼ばれる品質要求を扱う基盤を用意した。品質特性群は、ISO9126 に記載されている6つの品質特性及び、品質特性に付随した品質副特性を利用して構築される。図 3-2では、右上部に品質特性群を配置し、各品質特性を雲形の図形で表現した。これらの品質特性群を最上位ゴールとして、品質シナリオをマッピングし、ゴール分割を行った。

### 3.3. モジュール群の洗練化

KAOS による要求分析モデルを観察し、アスペクト指向型アーキテクチャの抽出を行う技法を提案する。図 3-4に、KAOS モデル上で、アスペクト指向型のアーキテクチャがどのように表現されるかを示したモデルを示す。

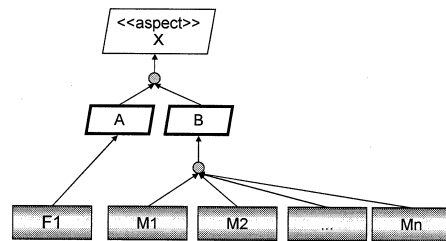


図 3-4 アスペクト指向型パターンモデル

図 3-4において、X はアスペクトとして抽出される要求(アスペクト要求)である。Xは、AとBという2つの要求に分解される。A は X を実現する為の



機能を提供する。B は A の機能に情報を引き渡す要求が該当する。図 3-2において、X は「ログを出力する」が相当し、A は「ログの出力を実行する」が相当し、B は「ログデータを提供する」が相当する。さらに、B は、複数のモジュール(では M1~Mn)に対して責任を割り当てる。図 3-2では、「ログデータを提供する」要求は、以下の4つモジュールに責任を割り当てる。

- リダイレクト情報ファイル読みモジュール
- PC/携帯判定モジュール
- URL パラメータ判定モジュール
- リダイレクト実行モジュール

これら4つモジュールはアスペクトが埋め込まれるモジュールとなる。X は、A に依存し、B は、複数のモジュールに依存する。

本稿では、X をアスペクト要求として、ステレオタイプ<<aspect>>を付与して KAOS 要求分析モデル内で明示的に取り扱った。

#### 3.4. パターンマイニング

ADD によるアーキテクチャ設計の際に、入力として利用されるアーキテクチャパターンを発掘するプロセスとして、パターンマイニングによるパターン発掘を工程として取り込んだ。これにより、経験や知恵といった暗黙知を顕在化し、アーキテクチャ分析・設計に入力可能なアーキテクチャパターンを発掘することができる。

#### 3.5. アーキテクチャ設計及び評価

本稿では、事例として簡単な WEB システムを扱い、手法間の整合性を考慮することに主眼を置いた。よって、一般的なアーキテクチャ設計手法を適応するという観点から、アーキテクチャ設計手法として品質特性駆動型設計(ADD)を適応した。また、アーキテクチャ評価手法として SAAM (Scenario-based Analysis of Software Architecture Method)[6]を適用した。

本提案では、品質要求を要求分析の段階で、品質シナリオとして具体的なストーリーで記述することにより、明示的に扱った。この品質シナリオを、ADD に投入し、アーキテクチャ設計を実施した。

アーキテクチャ設計を行うにあたり、アーキテクチャを決定する際の主軸となるアーキテクチャドライバを選択する必要がある。本稿では、図 3-2の品質特性群の内、変更性、試験性アーキテクチャドライバとして選択した。これらの品質特性は、短納期といった制約にも結び付く。ここで選択したア

ーキテクチャドライバを中心に捉え、KAOS の要求分析モデルで抽出・洗練化されたモジュールをアーキテクチャ構成要素としてアーキテクチャモデル上に配置していく。その際、よく知られているアーキテクチャパターン、パターンマイニングにより発掘されたアーキテクチャパターンを適用し、アーキテクチャを決定していく。

図 3-3に設計したアーキテクチャを示す。図 3-3において、Boundary-Control-Entity パターンを基盤アーキテクチャとして選択した。また、ログモジュールは、アーキテクチャ候補として、以下に挙げる2つを選択した。

- アスペクト型ロギングモジュール
- (非アスペクト型)ロギングモジュール

最終的には、変更容易性を考慮して、アスペクト型ロギングモジュールを選択した。

## 4. 評価

本稿では、1章において、アーキテクチャ分析・設計を行う際の問題点を3つ指摘した。

問題 1:品質要求の曖昧な定義

問題 2:属人的な分析・設計とトレーサビリティの欠如

問題 3:各手法間の整合性の必要

問題 1 は、品質要求を要求分析の段階で明示的に取り扱うことで対処した。その際、品質シナリオを作成して、要求分析の入力とした。

問題 2 は、ソフトウェア工学の手法を適応することにより対処した。具体的には、要求分析の段階ではゴール指向要求分析手法である KAOS を適応した。また、アーキテクチャ設計手法として、品質特性駆動型設計手法を適応した。

問題 3 は、手法間の整合性を考慮したアーキテクチャ分析・設計手法を提案することにより、対処した。

## 5. 関連研究

提案手法に関連して、要求分析の段階で品質要求を明示的に取り上げて分析し、トレーサビリティを確保してアーキテクチャ設計・評価へとつなげる取り組みがある。

例えば Tropos プロジェクト[10]や Grau ら[11]は、i\*を活用して要求分析からアーキテクチャ設計・選択(評価)までを一貫して実施している。これら

は一貫性に優れる一方で、各工程における他手法の取り込みの方法や可能性について幾らか不明な点がある。Janiらは、提案手法と同じくKAOSを用いて要求を分析し、続いて既知のパターンを併用してアーキテクチャへと変換することを試みている[12]。この試みではアーキテクチャ候補群に対する選択・評価について幾らか不明な点がある。今後、提案手法との比較評価を通じて、これらの関連手法を含めたそれぞれの特性と整合性確保の可能性を明らかにしたい。

また、Yuらは関係情報などが詳細化されたゴールモデル(要求分析モデル)上で構造等に基づいて将来のアーキテクチャや実装におけるアスペクトを特定する手法を提案している[13]。この手法と提案手法では扱うモデルの詳細さに相違があり、今後比較評価を通じて特性や併用可能性を検討したい。

## 6. おわりに

本稿では、簡単なWEBシステムの開発を事例として、複数の手法を用いた品質要求駆動型のアーキテクチャ分析・設計手法を提案した。具体的には、要求分析手法であるKAOS、アーキテクチャ設計手法である品質特性駆動型設計の整合性を考慮し、独自拡張を加えたアーキテクチャ分析・設計手法を提案した。さらに、要求分析の段階で、アスペクト指向型アーキテクチャを抽出する技法を提案した。

本稿では、アーキテクチャ設計手法として、品質特性駆動型設計を扱ったが、実問題では、システムの規模や要求等により最適な手法を選択する必要がある。同時に、手法間の整合を考慮する必要がある。本稿で扱ったアーキテクチャ設計技法以外に、アーキテクチャ決定上のトレードオフやリスクを明示的に扱うATAM (Architecture Tradeoff Analysis Method) [5]、費用や便益を考慮したCBAM(Cost Benefit Analysis Method) [9]等も、アーキテクチャ設計・評価手法の代替として検討可能である。これら手法の適応については、本研究の今後の課題とする。

## 謝辞

本研究にあたり、ソフトウェア工学に関する有用な知見を得る場を与えてくださったトップエスイープロジェクトに感謝いたします。また、国立情報学研究所の本位田教授、並びに、トップエスイープロジェクトの諸先生に心よりお礼申し上げます。

## 参考文献

- [1] 本位田真一(監修), 田原康之(著): サイエンスによる知的ものづくり1 要求分析, 近代科学社, 2007
- [2] 深澤良彰(監修), 鷺崎弘宜, 丸山勝久, 山本里枝子, 久保淳人(著): サイエンスによる知的ものづくり1 ソフトウェアパターン, 近代科学社, 2007
- [3] Len Bass, Paul Clements and Rick Kazman (著), 前田卓雄 他(訳): 実践ソフトウェアアーキテクチャ, 日刊工業新聞, 2005
- [4] Rob Wojcik, Felix Bachmann, Len Bass, Paul Clements, Paulo Merson, Robert Nord and Bill Wood: Attribute-Driven Design (ADD), Version 2.0, CMU/SEI-2006-TR-023, 2006
- [5] Paul Clements, Rick Kazman, and Mark Klein: Evaluating Software Architectures: Methods and Case Studies, Addison-Wesley, 2002,
- [6] Rick Kazman, Gregory Abowd, Len Bass and Paul Clements: Scenario-Based Analysis of Software Architecture, IEEE Software, November 1996, 47-55.
- [7] A. van Lamsweerde, A.Dardenne, Zb.Delcourt and F.Dubisy: The KAOS Project: Knowledge Acquisition in Automated Specification of Software. Proc. AAAI Spring Symposium Series, Stanford University, American Association for Artificial Intelligence, March 1991, pp.69-62
- [8] E.Yu and J.Mylopoulos: Using Goals, Rules, and methods To Support Reasoning in Business Process Reengineering, Proc. 27<sup>th</sup> Hawaii Int. Conf. System Science, Maui, Hawaii, Jan. 4-7, 1994, vol. IV.pp.234-243
- [9] Robert L. Nord, Mario R. Barbacci, Paul Clements, Rick Kazman, Mark Klein, Liam O'Brien and James E. Tomayko, Integrating the Architecture Tradeoff Analysis Method (ATAM) with the Cost Benefit Analysis Method (CBAM), CMU/SEI-2003-TN-038, 2003
- [10] Jaelson Castro, Manuel Kolp and John Mylopoulos: Towards Requirements-Driven Information Systems Engineering: The Tropos Project, Information Systems, Vol.27, No.6, 2002.
- [11] Gemma Grau and Xavier Franch: A Goal-Oriented Approach for the Generation

and Evaluation of Alternative Architectures,  
Proc. First European Conference on Software  
Architecture (ECSA 2007), 2007

- [12] Divya Jani, Damien Vanderveken and  
Dewayne Perry: Deriving Architecture  
Specifications from KAOS Specifications: A  
Research Case Study, 2nd European  
Workshop on Software Architecture (EWSA  
2005), 2005
- [13] Y. Yu, J.C.S.P. Leite and J. Mylopoulos:  
From goals to aspects: discovering aspects  
from requirements goal models, Proc. 12th  
IEEE International Requirements Engineering