

## モデリングは教育できるか？

妻木俊彦<sup>†1,4</sup>, 海谷治彦<sup>†1,3</sup>, 田原康之<sup>†1</sup>, 吉岡信和<sup>†1</sup>, 田口研治<sup>†1</sup>, 本位田真一<sup>†1,2</sup>

<sup>†1</sup>国立情報学研究所, <sup>†2</sup>東京大学, <sup>†3</sup>信州大学, <sup>†4</sup>日本ユニシス (株)

### 要旨

要求工学における重要な目標は、ステークホルダーの意図や期待を正確に理解し、厳密に定義することである。しかし、ステークホルダーが、自身の意図を明確に表現することは、極めて困難な作業である。問題領域のモデル化は、対象を客観的に捉える為の有効な技術と考えられ、多くの手法の中で採用されてきた。それにもかかわらず、実際のプロジェクトでのモデル化は、要求工学プロセスにおける最も困難な作業の1つである。その原因の1つが、モデリングにおける中心的作業である抽象化が、個人の認知スタイルに、大きく依存しているためであると考えられる。モデリングすなわち抽象化能力を教育によって高めることができるかどうかは、SW工学教育においても大きな課題である。われわれは、初期フェーズ要求工学手法であるi\*法の教育を通して、学習者によって作成されたモデルの問題を認識し、その原因を解明することによって、要求工学教育におけるモデリング教育の方法について提案する。

### Is it possible to teach the modeling technique?

Toshihiko Tsumaki<sup>1,4</sup>, Haruhiko Kaiya<sup>1,2</sup>, Yasuyuki Tahara<sup>1</sup>, Nobukazu Yoshioka<sup>1</sup>,  
Kenji Taguchi<sup>1</sup>, Shinichi Honiden<sup>1,3</sup>

Information Systems Architecture Research Division, National Institute of Informatics<sup>1</sup>  
Department of Information Engineering, Shinshu University<sup>2</sup>  
Graduate School of Information Science and Technology, The University of Tokyo<sup>3</sup>  
Nihon Unisys Ltd<sup>4</sup>

### Abstract

*Modeling a target domain is one of most significant but difficult works in the requirements engineering. Because, main activity of the modeling is abstraction and the abstraction depend on cognition style of each engineer. It is a big issue whether increasing of modeling ability is possible or not. In NII, we are trying to teach the modeling ability by various methods. In this paper, we report the typical problems of models presented by i\* method education and discuss about the modeling education.*

#### 1. はじめに

良いモデルの作成は、1つ要求工学に限らず、ソフトウェア工学全体の中心的テーマの1つである。その理由は、図式を使用するモデルが、絵の

持つ直観的な理解性と、視覚言語としての厳密性を併せ持っているため、以下のような性質に依存している。

- ・ 対象の全体構造を、簡単に表現できる。
- ・ 対象の構造的な評価が容易である。
- ・ 問題の本質を、容易に把握できる。

- ・ 基本的な間違いを、簡単に発見できる。
- ・ 他者との共通認識を得ることが容易である。
- ・ 他のモデルへのモデル変換が可能である。

同じモデルでも、要求モデルと設計モデルでは、モデルの作成の仕方に違いがある。要求モデルでは、問題領域の抽象化し、問題の解決方法を提示することが中心テーマであるのに対し、設計モデルでは、要求された機能のアルゴリズムとソフトウェア・モジュールの適切な構造を作成することが中心課題となる。本稿で検討対象とするのは、要求モデルである。

このように、モデリング技術の向上は、ソフトウェア工学における重要な課題の1つであるにも関わらず、プログラミング技術などと異なり、その技術力の向上は困難を極めているのが現状である。特に、要求モデルの作成には、豊富な業務知識とシステム開発経験が必要と考えられ、実際の開発現場での OJT が、要員育成の最も一般的な教育となっているのが現状である。しかし、開発現場での OJT では、経験の蓄積やノウハウの伝授は可能であっても、伝統的技術への固執により、新しい技術の導入に障害が伴うことが多い。その意味で、講義形式の教育に課せられる課題の1つは、新たな技術教育である。

要求モデリングに必要とされる抽象化と問題解決のための能力には、多分に先天的な要素が大きく作用し、教育によってその能力を高めることには限界があると考えられている [5]。能力の違いによって技術者の進むべき方向を示唆することができるのも確かに OJT の良さ1つではある。しかし、先天的な能力だけに依存しては、モデリング作業はいつまで経っても技能の域を脱することが出来ないだろう。モデリングが技能から技術に脱皮するには、講義形式での能力の向上が可能とならなければならないと考えられる。

本稿では、NII のトップエスイー・プロジェクトでの教育経験をもとに、モデリング技術のための教育について提言する。

本稿では、第2章で、われわれが行っている i\*法の教育を紹介し、第3章で、その教育を通して作成された具体的なモデルについて検討する。第4章では、検討結果をもとに、現実世界の抽象化作業における問題を議論し、第5章で、今後の課題を提言する。

## 2. NII トップエスイーにおける要求モデリング教育

### 1. トップエスイー・プロジェクトの概要

国立情報学研究所 (NII) のトップエスイー・プロジェクト [7] では、'07年度、表1に示したようなソフトウェア工学教育を実施している。トップエスイーが目指す教育の基本姿勢は、モデリング能力の養成である。

トップエスイー・プロジェクトでの要求モデリング教育は、現在、要求分析及び安全要求分析講座で実施されている [4]。

要求分析講座では、ゴール駆動型要求分析に焦点を当て、KAOS 法と i\*法という著名な手法の教育を実施している。本講座は、90分×12回の講義で構成されており、KAOS 法 [1] と i\*法に関する教育はそれぞれ5回の講義を割り当て、最後に、総合演習のために2回の講義時間をとっている。今年度の受講生の数は17名で、企業人と大学院生から構成されている。本稿では、この中の i\*法の教育での例を題材に要求モデリング教育について述べる。

表1 NII TopSE の教育コース

Series	Modules
Requirements Analysis	Requirements Analysis
	Security Requirements Analysis
System Architecture	Component-based Development
	Software Patterns
	Aspect-Oriented Development
Formal Specifications	Formal Specifications (Foundation)
	Formal Specifications (Advanced)
	Formal Specifications (Security)
Model Checking	Design Verification (Foundation)
	Design Verification (Advanced)
	Software Model Checking
	Real-time Model Checking
Implementation	Modelling and Verifying Concurrent Systems
	Testing
	Program Analysis

### 2. i\*法による要求モデリング教育目標

i\*法を使用したモデリング教育の目標は、  
抽象化技術力の向上  
問題解決スキルの向上

である。情報システムの役割が、人間が行っていた作業の生産性を高める為だけのものから、ビジネスプロセスの改革を期待されるようになり、要求技術者の資質として、問題化解決能力が問われるようになってきている。

i\*法は、トロント大学の E.Yu らによって提案された初期フェーズ要求工学用手法で、その商用版は Toropos としてリリースされている。初期フェーズとは、これまでの要求獲得プロセスに先立つという程の意味であり、これまでに提案されて

きた要求獲得手法の多くが、正確かつ完全な要求の獲得を目指していたのに対し、この手法が焦点を当てるは、個別の要求の前提となる個々のステークホルダーの意図であり、問題領域全体を、組織目標と実現するためのネットワークとしてモデル化しようとする。研究用の手法である *i\** には、多くの版があるが、本教育でベースとしたのは文献 [8] である。

*i\**法の特徴の第1は、問題領域を多様なモデル構成要素によって表現しようとしている点にある。*i\**モデルには、ゴール駆動型モデルの要素であるゴールやソフトゴールの他に、アクターや依存関係といった静的モデルの要素や、タスクやリソースといった動的モデル要素から構成されている。複雑な問題領域を正確に表現するには、当然、モデル構成要素の種類も多くならざるを得ない。

しかし、多すぎるモデル構成要素は、モデルそのものの複雑性を増加させ、見やすさを損なうことになる。*i\**法では、アクターの外部構造と内部構造という2つの局面からモデルを作成使用する。それらのモデルは、それぞれ、戦略依存モデル (Strategic Dependency model: SD モデル) と戦略論理モデル (Strategic Rationale model: SR モデル) と呼ばれている。SD モデルは、意図的アクターと、アクター間の3種類の依存関係 (ゴール依存、タスク依存、リソース依存) から構成される。SR モデルは、SD モデルのアクター同士のゴール依存関係から、ゴール分解 (Means-Ends 分解) やタス分解によって、それぞれのアクター内部のゴール、サブゴール、タスク、リソースと、それらの関係が導出される。こうした多様なモデル構成要素の使用は、対象の抽象化と整理という抽象化技術の向上に役立つと考えられる。

*i\**法の特徴の第2は、問題領域が抱えている問題に注目し、IT による解法を提示しようとすることである。言い換えれば、問題領域の基本的な構造を把握することによって、将来発生する可能性のある要求変更を先取りすることに主眼が置かれていると言ってよい。このことから、*i\**法では、問題を抱えた現状モデル (without computer model) と、問題が解決された将来モデル (with computer model) という2つのモデルが描かれる。同様のアプローチは、所謂、ビジネスモデリングなどでも採用されているが、ビジネスモデリングでは経営やマーケティングに関する知識が必要であるのに対し、*i\**法では、現場で活動している個々のアクターの潜在的意図に着目し、ソフトゴールとしてモデル化するため、ビジネス活動に関する知識以外の知識を必要とはしない。このように、*i\**法に

よるモデリングでは、現術世界の単なる抽象化ではなく、問題解決能力を必要とする。

### 3. 事例研究

ここでは、2つの事例について紹介し、それについて検討を行う。なお、紹介するモデルは、使用しているモデル記述ツール ST Tool [3, 6] は、この時点で日本語の入力ができなかったため、すべて英語である。

なお、受講生に与えたテーマは、「自分自身の身の回りにある問題」というだけである。

ちなみに、ゴール指向分析の方法については、この前の実施される KAOS 法のコースでカバーされている [2]。

#### 3. 事例1

テーマ： マンションやアパートといった集合住宅では、外部の飲食店から取り寄せた店屋物の空き食器が、廊下やエレベータホールといった共有場所に長い間放置され、住環境の美観を損ねているという問題がしばしば発生している。

プロセス： 最初に提出された現状モデル (図1) では、テナヤモノをオーダーした customer と出前業者である *delivere* との間に4つのソフトゴール “cleanly”, “unquickly pickup”, “easy to find empty dishes”, “pickup any time” が設定され、さらに、共同住宅の居住者間に “invisibly”, 業者と他の居住者の間に “quickly” というソフトゴールが設定されており、この問題領域で解決すべき課題が明確にされている。しかし、テナヤモノのオーダーに関し、顧客と業者の間に “order” “delivery” “return” 3個の依存ゴールが設定されており、これがモデルを複雑化していた。ここで、講師は、解決したい問題に焦点を当てるようにというアドバイスを送った。その結果、依存ゴールは “order delivery” という1個に絞られることになった (図2)。この新しいゴールでは、達成すべき状態は、必ずしも明確ではなかったが、取りあえず、この現状モデルをもとに、将来モデルを作成することにした。しかし、将来モデルを作成する過程において、このゴールのままでは、「受け取り」と「返却」の両方の仕組みが必要となり、解決すべき問題があいまいになる恐れが生じた。そこで、講師は、解決すべき課題を記述したソフトゴールを検討させた。受講生は、それら

がすべて「返却」に関するものであり、「受け取り」時には障害が発生していないことに気が付き、ゴールをさらに絞り込む必要があることに気がついた。最終的に，“Return empty dish”をゴールとしたモデルが作り直され(図3)，このモデルに基づき，Return box というコンピュータシステムを導入した将来モデル(図4)が作成された。このモデルでは，最初に設定したソフトゴールがReturn box によって解決される見通しがたった。

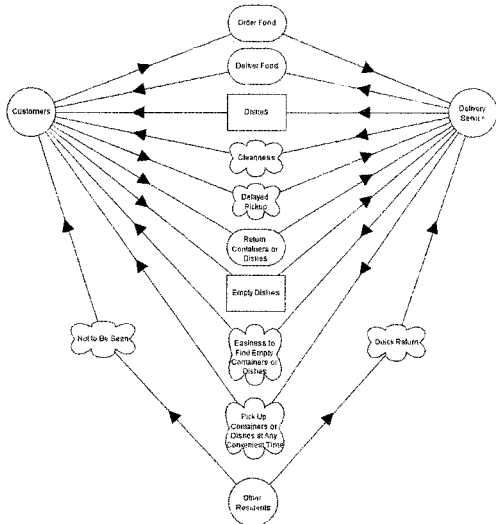


図1 現状 SDモデル 1-1

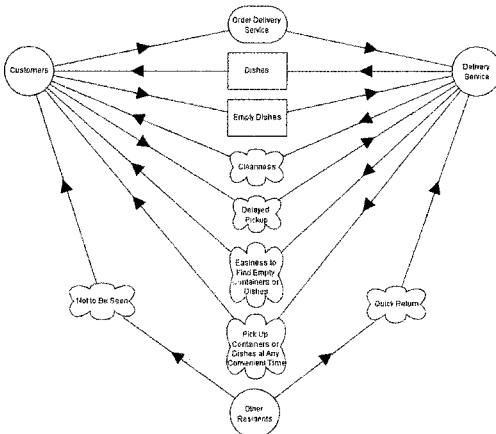


図2 現状 SDモデル 1-2

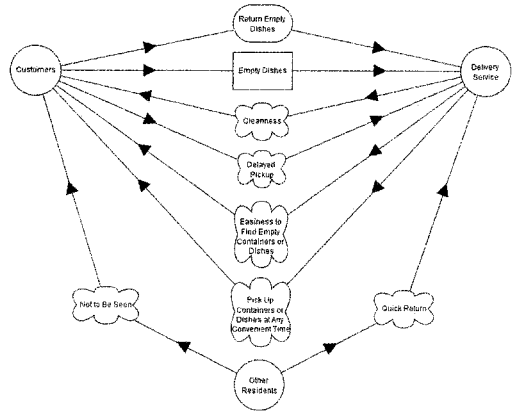


図3 現状 SDモデル 1-3

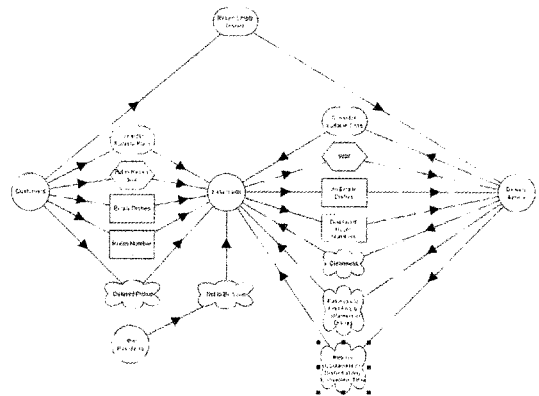


図4 将来 SDモデル 1-4

考察： 問題領域のモデル化に際して，適切なモデル化範囲を設定するのは，高度な作業である。モデル化の範囲が必要以上に大きくなれば，不要な情報によって検討すべき問題があいまいになってしまう。\*のように，豊富なモデル構成要素によって構成されるモデルでは，その傾向が，顕著に表れる。一方，それを必要以上に小さくすれば，問題の本質がモデルから外れてしまい，解が見つからないという状況に陥ってしまうことになる。本事例のように，事実を正確に把握しようとするめ，多すぎる情報を抱えて，それを整理し，抽象化することが困難になる場合には，重要な情報と重要なでない情報を区別するために，問題の本質の再確認をアドバイスすることが有効であると考えられる。

本事例では、対象範囲を絞り込む方向で、モデルの作り直しを行ったが、それは、最初のモデルで、解決すべき問題が、ソフトゴールとして明確に定義されていたからである。その結果、講師のアドバイスは、受講生自身が作成したモデルの再検討だけで十分効果を上げることができた。モデル化技術の向上にとって、講師からの予期せぬアイデアの提示より、自身のモデルの修正によって納得のゆくモデルを完成するという教育法は、受講生の自信を高めることにつながるだろう。この事例から得られた教訓は、問題領域のモデル化に際しては、解決すべき問題を明確にするということである。i\*法は、それをソフトゴールとして明示的に表現することができる。オブジェクトモデルやプロセスモデルのように、解決すべき問題を明示的に表現しないモデル化技法でも、他の方法で、それを定義することが必要であろう。

#### 4. 事例 2

テーマ： 本事例のテーマは、組織の強化についてのモデルの作成に挑戦した。或る組織では、メンバーの出入りが激しく、組織としての情報や知識の共有がされ難いことが問題となっていた。この問題は、組織の問題として取り上げられ、情報データベースが構築された。しかし、新たな組織メンバーには、この情報データベースが役に立つことはなく、常に、必要な情報の不足になやまされ続けることになった。こうした現状から、新たな組織の将来像が描かれている。

プロセス： 最初に描かれた現状モデルは、組織の情報環境であった(図5)。このモデルで、最初に問題となったのは、アクターMeであった。Meが、何故、アクターとして不適切であるかについて、講師は、当初、クラスとインスタンスの例をもとに抽象化の説明をしようとしたが、十分な理解を得るに至らなかった。そこで、講師は、組織内の役割に焦点を当てて、アクターを特定するように指導した。その結果、知識の提供者と受領者という役割に注目することによって、新たなモデルを作成しすることができた(図5)。このモデルの Old member は、年寄りということではなく、その組織に長くいるという意味である。長く在籍していれば、知識や情報をたくさん持っているとい概に言うことはできないが、モデリングの演習ということで、リソース依存とタスク依存の違いを指摘し、モデル化を先に進めることにした。それ以上の指摘は、本人を混乱に陥れるだけ

になるので、依存方向については、無視することにした。次に提出された現状 SR モデルでは、データベースとの情報交換は省略され、メンバー間での情報の共有も、製品情報だけに絞られた(図6)。さらに、将来モデルでは、映像によって情報を伝える仕組みが提案された(図7)。しかし、現状モデルにおける課題がソフトゴールとして明示的に定義されていないために、将来モデルによって何がどの程度解決されるかを判断することが出来なかった。

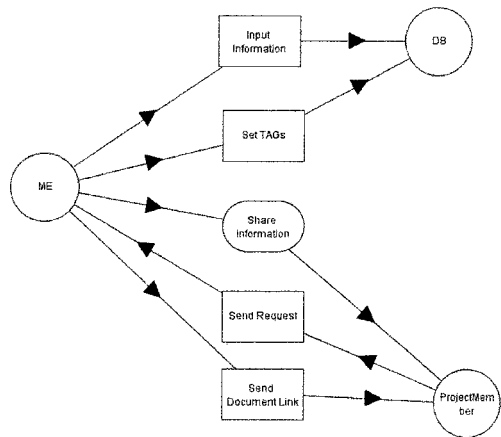


図5 現状 SD モデル 2-1

考察： このモデル化プロセスでは、対象の定義が常に具体化、特定化してゆく傾向が見られた。モデリングは、抽象化操作であるといわれている。しかし、実際のモデルの作成では、抽象化の方向と、抽象レベルの決定が必要になる。そして、これらの決定は、多分に、モデラーの発見的知識と認知スタイルに依存する。実際、オブジェクトモデリングにおいても静的モデルの作成が得意なモデラーと、不得手なモデラーがいることは、よく知られている。オブジェクト指向技術に熟達していることが、クラス図を描けるということと同じではない。モデリングは、個人の認知スタイルの特性に大きく依存している所以である。発見的知識は、教育や経験が有効であるが、認知スタイルは、それが個人のアイデンティティの重要な位置を占めることを思えば、教育によってそれを強制的に変更することには問題があるとも考えられる。教育による改造の可能性は発見的知識のレベルに限定すべきであるかもしれない。認知スタイルに

は違いがあるだけで、善し悪しがあるわけではない。また、同じ認知スタイルであっても、個々の認知プロセスには無限の可能性があるのだから、できるだけ多くの認知プロセスを開拓してやるという方法はある。ただし、それは、時間のかかる作業だろう。個人の認知スタイルの特性を踏まえた上で、モデリングにおける抽象化操作を教育するということは、今後の大きな課題である。

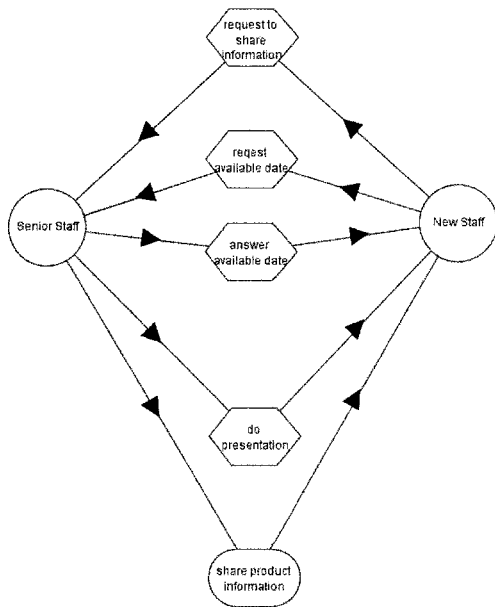


図6 現状 SD モデル 2-2

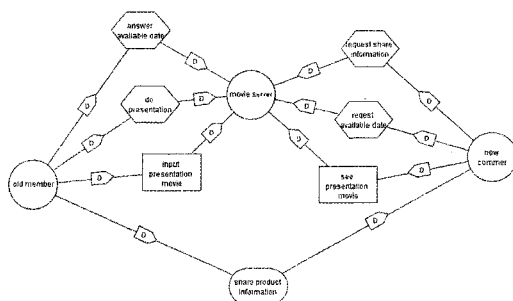


図5 将来 SD モデル 2-3

#### 4. 議論

以下、本演習を通して得られたその他の知見について議論する。

**抽象化：** 自分自身をアクターにした例が、多数見受けられた。受講生の多くが優れた SE であることを考えれば、これは驚くべきことである。通常システム開発では、モデル化の対象が、製品や組織のように、ビジネス規則が確立しているビジネス世界である場合には、既に概念化、すなわち、抽象化されている対象を扱うことが圧倒的に多い。その結果、モデラーの抽象能力に関わりなく、それなりのモデルが出来上がることになる。今回の演習のように、身近な問題、すなわち、概念化が行われていない世界を相手にしたときに、その弱点が如実に現れてくると考えられる。今後、情報化の対象世界が、未知の世界へと広がってゆこうとしているおり、この問題は、極めて大きいと考えざるを得ない。簡単な問題でお茶を濁しているような教育では、新たな情報システム開発の担い手は生まれにくいに違いない。

**問題の設定：** 今回の演習では、多くの受講者が、モデル化を始める前に、問題をどのように設定すればよいか悩んでいた。問題は、それを見る人の視点によって異なった姿を現す。抽象化と同じように、未知の問題に遭遇した時、それをどのようにこなせば良いかも、モデリングにおける重要な能力である。多くのモデル化手法は、問題そのものが、ある程度、整理された世界を扱うのには適している。言い換えれば、システム化対象の問題領域の多くは、モデラーが認識する前に、標準的な認識が定着した世界であるということかもしれない。i\*法のように、ゴールを導入する手法は、組織目標を明確にする糸口を与えてくれる。しかし、そのことは、多くのゴール分解法が抱える多様性の犠牲という問題をも抱え込むことになる。教育者は、個々の手法の限界を理解した上での教育を実施しないと、現実世界の豊饒さに対応する技術者は生まれまいだろう。

**問題の解決：** 受講生の多くは、はじめのうち、現状モデルのゴールやタスクの粒度をどの程度に設定すべきか戸惑っていたが、将来モデルの作成を通して、対象の粒度は、解決すべき問題の設定によって異なることを理解していった。ドメインモデリングという、問題領域内の対象を与えられた記法に従って表現するだけという考えを持

った技術者が多いが、意味のあるモデルを作成できるようにするには、問題領域が抱える問題を速やかに見抜き、適切な解を得ることのようなモデル化技術を身につける必要がある。一方、管理情報のコンピュータ化という方法で、問題のかなりの部分が解決することが知られている。しかし、益々複雑化してゆく現実世界の中で、それは問題解決の1部に過ぎないことを、学習者に知ってもらうことも重要である。できるだけ、異なったタイプの、解決するのが困難な問題を解く技術教育が、モデル化にとって重要であると再認識している。

手法の相性： i\*法は、ゴール以外にソフトゴールやタスクやリソースといったモデル構成要素により、現実世界を忠実な、リッチなモデルを作成することが可能である。そのことは、一方で、モデル構成要素が多すぎて、うまく抽象化しないとモデルが大きくなりすぎ、問題がぼやけるといふ欠点につながる。反対に、構成要素の少ないモデル化手法は、問題を尖鋭的に表現できるが、その分、多くの情報が欠落する。実際、オブジェクト指向モデリングは、静的モデルと動的モデルの分離を図って成功したが、i\*法は、再び、ゴールとタスクとリソースの統合を目指そうとしている。どのような手法が適正なのかは、問題の性質と技術者の思考スタイルに依存するかもしれない。我々は、トップレベルのSE教育の中で、問題に応じて、モデル化手法を選択できるようにと、様々な技法の教育を実施しているが、技術者の性質と手法の相性の研究は、今後のテーマである。

## 5. まとめと今後の課題

今回は、トップSEに対する5日間のi\*の研修を通して得られた知見について、報告を行った。

演習は、初期フェーズ要求獲得というより、問題領域のモデリングという点に焦点があたった。初期フェーズ要求獲得手法としては、将来モデルのさらなる洗練を行い、仕様化できる状態まで詳細化する必要があるが、時間的な問題から、そこまでは手が回らなかった。したがって、本報告は、ドメインモデリングという観点での報告となっている。

多くのモデリング演習では、解があらかじめ分かっている問題が与えられることが多いが、そうした問題は、モデリングがしやすいように作られている。しかし、本当のモデリング技術を身につけるには、例えどんなに小さくとも、未知の問題

に挑戦することが有効であることが証明されたと考えている。それは、本稿で報告したように、受講者自身による問題領域の設定と抽象化操作に対する挑戦でなければならない。その他の問題については、もう少し多くのデータを収集しなければならないと考えている。

今回の教育を通して得られた結論は、モデリング作業での課題であり、それを系統的に教育するための方法は、今後の課題である。

問題領域のモデリングにしろ、問題解決にしろ、それらの能力を向上させるためには、問題領域が抱えている問題そのものの把握と設定が重要であり、解が事前に用意されている、所謂、玩具問題では、本当のスキルを向上は困難であるとの考えられる。本コースでは、それぞれの受講生に、自分自身の身の回りの問題を独自に設定させることにした。

i\*法のモデルは、複数の目標と、それを実現するための多くの構成要素からなる、複雑なモデルを作成し、問題領域が抱えている問題を解決しなければならない。この特徴は、複雑な問題領域をモデル化するための道具として有効であると考えられる。

## 謝辞

トップエスイープロジェクトは、平成16年度文部科学省科学技術振興調整費の支援を受けたプロジェクトである。

## 参考文献

- [1] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-Directed Requirements Acquisition. *Science of Computer Programming*, 20:3–50, 1993.
- [2] R. Darimont and A. van Lamsweerde. Formal refinement patterns for goal-driven requirements elaboration. In *Proc. of FSE'96*, pages 179–190, 1996.
- [3] P. Giorgini, F. Massacci, J. Mylopoulos, A. Siena, and N. Zanneno. ST-Tool: A CASE Tool for Modeling and Analysing Trust Requirements. In *Proceedings of the Third International Conference on Trust Management (iTrust 2005)*, LNCS, pages 415–419. Springer-Verlag, 2005.
- [4] S. Honiden, Y. Tahara, N. Yoshioka, K. Taguchi, and H. Washizaki. Top SE: Educating Superarchitects Who Can Apply Software Engineering Tools to Practical Development in Japan. In *Proceedings of International Conference*

- on Software Engineering (ICSE 2007), pages 708–718. IEEE, 2007.
- [5] J. Kramer. Is Abstraction The Key to Computing? Communications of the ACM, 50(4):37–42, 2007.
- [6] ST-Tool Web Site.  
<http://sesa.dit.unitn.it/sttool/index.php>.
- [7] Top SE web site. <http://www.topse.jp/>.
- [8] E. Yu. Towards Modeling and Reasoning Support for Early Requirements Engineering. In Proceedings of International Symposium on Requirements Engineering (RE 1997), pages 226–235. IEEE, 1997.