



Title	情報システム計画における事例活用ならびに解釈に関する研究
Author(s)	平松, 綾子
Citation	大阪大学, 1998, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3144015
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

情報システム計画における事例活用
ならびに解釈に関する研究

1997年 12月

平松綾子

内容梗概

本論文は、著者が1994年から現在までに大阪大学工学部情報システム工学科、大阪大学大学院工学研究科博士前期課程ならびに後期課程在学中に行った情報システム計画における事例の活用、解釈に関する研究の成果をまとめたものである。

近年、情報技術の進展は目覚しく、情報システムの適用範囲が拡大している。従来、情報システムは省力化、コスト低減など、既存事業の枠組みの中で企業体質を強化するための手段であったのに対し、これまで蓄積してきたデータ資源と通信を基盤として情報システムを再構築することにより競争戦略の武器として重要な役割を果たすようになった。このように、情報システムに求められる役割が、業務レベルの処理から戦略的な経営レベルのものへと変遷するに伴い、システム開発の最も上流工程である情報システム計画工程の重要性が増してきている。しかしながら、このような工程を受け持つシステムエンジニア (SE) に対しては、業務知識とシステム計画のための問題発見能力や、提案能力が必要とされるため、経験の少ない初級者には困難な工程である。そのため、初級 SE でも熟練 SE と同様に業務遂行を行うための支援と共に、熟練 SE の育成が望まれている。

このような課題を解決するため、本論文では熟練者の知識の代わりに過去に作成したシステム計画事例を活用することにより、クライアントサーバシステム (CSS) の計画支援方式と、定性的な構造モデルを用いたシステム導入効果の予測方式を提案する。

CSS の提案において、顧客からの要求情報入手作業は容易ではない。さらに、技術の進歩が急速であり、新技術に対応するための知識も要求される。本論文では情報システム計画業務中の各プロセスを個別に支援するの手法を提案するのではなく、初級者でも手法の学習の必要がなく業務遂行が可能となるように、要求分析から設計までの業務全体を統合的に支援する CSS 構成設計支援手法の提案を行う。本方式では、支援システムを要求分析部、設計案作成部、設計案分析説明部、知識学習部から構成する。要求分析部において、顧客からのシステムに対する曖昧な要求から具体的な要求を推論し、その出力をもとに設計案作成部においてさまざまな制約や、要求に含まれている競合問題を考慮し、提案内容である CSS 構成を作成する。設計案分析説明部では、作成された CSS 構成が複雑に対応づけされた要求を満たしているかどうかを検討し、ユーザに分かりやすいように提示するための説明を作成する。また、知識学習部では構成設計案の作成の際に必要な構成部品決定上の制約条件を見出す。このように、CSS 開発における情報システム計画業務全般を自動的に行うことで、経験のない初級者の SE に対して CSS 提案業務を支援する。

また、新システム導入に対する効果の予測は、企業内の因果関係を把握してなければならず、特に定性的な効果の予測は熟練者の経験によることが多い。システム導入に対する効果予測のために、企業内の因果関係を表現した構造モデルを用いた定性シミュレーションがある。この定性シミュレーションでは、結果理解のためにはモデル自身の構造を理解する必要がある。構造モデルが複雑化すると、モデルの理解は困難となり、モデルの作成者である熟練 SE によって解釈されなければならない。そのため、モデルの作成者でない初級 SE は、シミュレーション結果に対しての理解が困難であったり、間違った因果関係を理解してしまうといった問題がある。そこで、本研究では定性シミュレーション結果を分析することで、実質的な因果関係を抽出し、結果に対する説明としてシナリオを作成する。

本論文は全 7 章から構成される。第 1 章に序論を述べる。第 2 章にクライアントサーバシステム向け構成設計支援システムである CIDAC (Case based Integrated Design Assist system for CSS) の構想を述べた後、以下の第 3,4,5 章で CIDAC 中の対話型要求分析部、事例ベース型設計案作成部、設計案分析説明部について述べる。第 3 章では対話型要求分析部における業務ツリー作成方式について述べ、第 4 章では事例ベース型設計案作成部における事例修正のためのクライアントサーバシステム構成記述方式を示し、第 5 章では設計案分析説明部における説明文作成方式について述べる。また、第 6 章では情報システム導入における期待効果を予測する定性シミュレーションシステムにおいて挙動の類似性に着目した因果関係の抽出による出力結果のシナリオ化について述べ、第 7 章に本論文の結論を述べる。以下に各章の概要を示す。

第 2 章では、CSS 構成設計業務を 3 つのフェーズに分け、それぞれの業務内容および問題点を明確にし、支援システムにおいてどのような機能が必要となるか分析する。その分析結果に基づき、CSS 向け構成設計支援システム CIDAC(Case based Integrated Design Assist system for CSS) の構成を考案する。

第 3 章では、CIDAC の入力のための機能である対話型要求分析部に焦点をあてる。まず、CIDAC における要求分析部の役割と共に本機能の出力となる要求仕様について述べる。次に、要求仕様の一つである業務ツリーを定義し、作成上の問題点を列挙した後、事例を用いた業務ツリー作成方法を示す。また、SE 経験の無い被験者による本手法を用いた業務ツリー作成と事例参照による業務ツリー作成との比較実験について述べる。

第 4 章では、CIDAC の中心機能の一つである事例ベース型設計案作成部に焦点をあてる。まず、CSS 構成設計案作成方式のための入力条件を定義し、事例修正による CSS 構成設計案作成手法について述べる。ついで、事例修正の際に用いられるデータの表現形式を考案し、修正例を示す。

第 5 章では、CIDAC の出力解釈のための機能である設計案分析説明部に焦点をあてる。まず、設計案分析説明部の CIDAC における役割について述べ、説明文作成手法の入出力

を定義し、入力である要求分析過程情報、設計案作成過程情報の説明を行う。ついで、どのような設計案分析説明が必要となるか考察した上で、説明文作成方式と共に説明作成例を示す。

第6章では、定性シミュレーション結果の説明文作成方式について述べる。まず、定性シミュレーションに用いる構造モデルについて説明し、シミュレーション手法について述べる。ついで、シミュレーション結果の説明であるシナリオ作成のための本質的な因果関係抽出方法について述べた後、自然言語の説明文として出力するための文章化について述べる。また、作成されるシナリオの因果関係の妥当さを評価するために、本手法による因果関係の抽出と、一定の因果関係の利用との比較実験を示す。

第7章では、本研究で得られた成果を要約した後、今後に残された課題について述べる。

研究業績

A. 学術論文誌論文

- (1) 平松綾子, 畑慎也, 大川剛直, 薦田憲久: “挙動の類似性に着目した因果関係の抽出による定性シミュレーション結果のシナリオ化方式,” 電気学会C部門論文誌, Vol.116, No.3, pp.381-388 (1996.3) .
- (2) 平松綾子, 一階良知, 大川剛直, 薦田憲久: “有意性判定によるクライアントサーバシステム設計案説明文作成方式,” 電気学会C部門論文誌, Vol.117, No.5, pp.618-624 (1997.5) .
- (2)’ Ayako Hiramatsu, Yoshitomo Ikkai, Takenao Ohkawa, and Norihisa Komoda: “An Explanation Composition Method for Client Server System Configuration Design Based on Significant Judgement,” Electrical Engineering in Japan, John Wiley & Sons, Inc (to appear).
- (3) Ayako Hiramatsu, Yoshitomo Ikkai, Hiroshi Morihisa, Takenao Ohkawa, and Norihisa Komoda ; “Requirement Analysis for Client Server System Design based on Case base,” Studies in Informatic and Control, Vol.6, No.3, pp.281-293 (1997. 9) .
- (4) 平松綾子, 一階良知, 森久博, 大川剛直, 薦田憲久: “クライアントサーバシステム構成設計向け事例修正方式,” 電気学会C部門論文誌, Vol.118, No.4(1998. 4)(掲載決定).

B. 国際会議

- (1) Ayako Hiramatsu, Shinya Hata, Takenao Ohkawa, and Norihisa Komoda; “Scenario Generator for Qualitative Simulation System,” in Proc. of 1995 IEEE Int. Conf. on Systems, Man and Cybernetics(SMC’95), pp.143-148 (1995.10.22-25, Vancouver, Canada).
- (2) Ayako Hiramatsu, Yoshitomo Ikkai, Takenao Ohkawa, and Norihisa Komoda; “Requirement Analysis for Client Server System Design based on Case base,” in Proc. of 15th IMACS World Congress - Scientific Computation Modelling and Applied Mathematics - Vol.4, pp.31-36 (1997.8.24-29, Berlin, Germany).

- (3) Ayako Hiramatsu, Atushi Naito, Yoshitomo Ikkai, Takenao Ohkawa, and Norihisa Komoda; "Case Based Function Tree Generator for Client-Server Systems Configuration Design," in Proc. of 1997 IEEE Int. Conf. on Systems, Man and Cybernetics (SMC'97), Vol.3, pp.2069-2074(1997.10.12-15, Orlando, Florida, U.S.A.).

C. 学会講演

- (1) 平松綾子, 大川剛直, 薦田憲久: "因果関係に基づく定性シミュレーション結果のシナリオ化," 平成7年電気関係学会関西支部連合大会, G3-33, p.G100 (1995.11.11-12, 京都) .
- (2) 平松綾子, 一階良知, 大川剛直, 薦田憲久; "事例修正によるクライアントサーバシステム構成設計支援システムの構想," 平成8年電気学会電子・情報・システム部門大会, B12 - 6, pp.585-588 (1996.9.5-6, 京都府田辺町) .
- (3) 平松綾子, 大川剛直, 薦田憲久: "事例ベースを用いたクライアントサーバシステム構成設計向け要求分析方式," 平成8年電気関係学会関西支部連合大会, G3-28, p.G100 (1996.11.23-24, 草津) .
- (4) 平松綾子, 小野田仙一, 大川剛直, 薦田憲久: "事例ベースを用いたクライアントサーバシステム構成設計手法," 第7回A I シンポジウム (人工知能学会研究会), pp.57-62 (1996.12.4-6, 東京) .
- (5) 平松綾子, 一階良知, 森久 博, 大川剛直, 薦田憲久: "クライアントサーバシステム構成設計向け事例修正方式," 平成9年電気関係学会関西支部連合大会, G14-3, p.G309 (1997.11.23-24, 吹田) .
- (6) 本田真吾, 平松綾子, 山松宏子, 一階良知: "クライアントサーバシステム演習用システムの開発," 平成9年電気関係学会関西支部連合大会, G15-9, p.G342 (1997.11.23-24, 吹田) .

目次

1	序論	1
1.1	はじめに	1
1.2	従来研究	3
1.3	本研究の方針	4
1.4	本論文の構成	5
2	クライアントサーバシステム構成設計支援システム CIDAC	7
2.1	まえがき	7
2.2	CSS 構成設計	8
2.3	CSS 構成設計業務の分析	9
2.3.1	要求分析	10
2.3.2	設計案作成	11
2.3.3	設計案の分析	11
2.4	支援アプローチ	12
2.5	CIDAC の構成	13
2.6	まとめ	16
3	要求分析部における業務ツリー作成	17
3.1	まえがき	17
3.2	業務ツリーの概要	18
3.2.1	要求仕様の記述	18
3.2.2	業務ツリーの定義	20
3.2.3	業務ツリー作成における問題	20
3.3	業務ツリー作成方式	21
3.3.1	作成アプローチ	21
3.3.2	事例ベース	22
3.3.3	事例の結合	26
3.4	事例結合例	27

3.5	比較実験	31
3.6	まとめ	33
4	クライアントサーバシステム構成設計向け事例修正方式	35
4.1	まえがき	35
4.2	事例修正による CSS 構成設計の概要	36
4.2.1	設計案作成部の概略機能	36
4.2.2	事例修正方式	36
4.2.3	事例修正方式における問題点	38
4.3	CSS 構成の表現形式	38
4.3.1	CSS 構成データ	38
4.3.2	部品データ	40
4.3.3	相性知識	42
4.4	プロトタイプシステム	42
4.5	事例修正例	43
4.6	まとめ	46
5	設計案説明文作成方式	49
5.1	まえがき	49
5.2	説明文作成におけるシステム構成	50
5.2.1	要求分析過程情報	50
5.2.2	設計案作成過程情報	51
5.3	設計案分析説明の作成	52
5.3.1	設計案説明文の種類	52
5.3.2	説明内容の有意性	53
5.3.3	センテンステンプレート	55
5.4	説明作成適用例	59
5.5	まとめ	62
6	定性シミュレーション結果のシナリオ化方式	63
6.1	まえがき	63
6.2	構造モデル	66
6.2.1	因果関係の種類	66
6.2.2	定性的情報	67
6.3	類型挙動パターン帰着型定性シミュレーション	69
6.3.1	シミュレーションの概要	69
6.3.2	シミュレーションの動作と例	69

6.4	シナリオ生成機能	71
6.4.1	シナリオ生成機能の概要	71
6.4.2	因果の流れの特定	72
6.4.3	重要ノードの選出によるシナリオ構成の生成	75
6.4.4	シナリオ構成の文章化	76
6.5	シミュレーション及びシナリオ生成例	79
6.5.1	実行例	79
6.5.2	検討	86
6.6	まとめ	88
7	結論	89
	謝辞	93
	参考文献	95

第 1 章

序論

1.1 はじめに

近年、情報技術の進展は目覚しく、情報システムの適用範囲が拡大している。従来、情報システムは省力化、コスト低減など、既存事業の枠組みの中で企業体質を強化するための手段であったのに対し、これまで蓄積してきたデータ資源と通信を基盤として情報システムを再構築することにより競争戦略の武器として重要な役割を果たすようになって来ている。このように、情報システムに求められる役割が、業務レベルの処理から戦略的な経営レベルのものへと変遷するに伴い、どのような情報システムが必要とされているのかの提案を行う、システム開発の最も上流工程である情報システム計画工程の重要性が増してきている。

代表的な情報システム開発プロセスであるウォーターフォールモデルは、「計画」「設計」「製作」「テスト」「運用・保守」からなる。計画フェーズでは対象とするシステム構築の意志決定と共に、業務システム変革の考え方と構想、業務処理の基本となる諸原則の確定、コンピュータシステムの分担範囲などのシステム構想を行い、以降に行われる設計作業に対する枠組を与える。

このような工程を受け持つシステムエンジニア (SE) に対しては、情報処理技術以上に、業務知識とシステム計画のための問題発見能力や、提案能力が必要とされる。情報システム計画工程の遂行に必要とされる上記のような能力の獲得は、方法論やガイドラインとして存在する知識の学習以外に、システム開発を実体験することにより得られる経験的な知識によるところが大きい。このため、情報システム計画工程を担当する SE は、システムアナリスト等の熟練した SE とされている。しかしながら、情報システム開発件数は急増しており、現在、計画工程を担当する熟練 SE の数が不足している。そのため、初級 SE でも熟練 SE と同様に業務遂行を行うための支援と共に、熟練 SE の早期育成が望まれている。

熟練者の知識を体系化することは、多くの熟練者の協力が必要となる上に、経験的に得

ていることが多いために困難である。過去の熟練者が開発した結果は事例として存在し、その中に熟練者の経験的な知識が含まれていると考えられる。従って、過去の事例を用いて初級者が学習したり、業務に利用したりするのは有効である。しかしながら、事例は形式化されていたり、共有されていることは稀であり、有効な活用方法が存在しない。さらに、事例活用するために熟練者が作成した事例の内容を理解することも、初級者にとっては困難である。

情報システム計画業務に対するツールや方法論として、BSP/CPS[1]、HIPLAN[2]、C-NAP[3]などが提案され実践の場で活用されている。このようなツールや方法論は、情報システムの品質向上、開発期間の短縮といった点において、多くの利益をもたらしている。しかしながら、システム開発経験の少ない初級SEに対する教育、提案能力や業務知識のサポートを行っている訳でない。従って、初級SEは予め方法論の修得や、業務知識の理解をした後、ようやく実践のシステム開発を経験することになる。また、これらの方法論では、情報システム計画業務をさらに細かなフェーズに分け、それぞれのフェーズにおいて手法やツールが考案されている [4][5][6][7][8]。SEはそれぞれのフェーズにおいて、それぞれの手法を利用している。そのため、初級SEが実務に携わるまでに、手法を学習のために多くの時間を要するだけでなく、熟練者にとっても初級SE指導のため本務のさまたげとなり大きな負担となる。さらに、新システム導入に対する効果の予測は、企業内の因果関係を把握してなければならず、特に定性的な効果の予測は熟練者の経験によることが多い。そこで、初級者が様々な手法の学習の手間が少なく、情報システム計画業務の統合的な支援、また熟練者に対して教育のための負担をかけない支援方法が必要とされる。

このような背景をもとに、本研究では情報システム計画において、クライアントサーバシステムの提案とシステム導入による効果予測を取り上げ、熟練者の知識の代わりに過去に作成したシステム計画事例を活用することにより、初級SEに対して教育と業務遂行を同時に行える様に、知識の提供、理解の補助という立場より計算機支援を実現する。まず、提案する方式では、熟練者が作成した事例をベースとして、システム概要提案および効果予測を行い、それらに対する解釈を添えることで初級SEに対する支援を行う。本研究ではまず、近年需要が増しているクライアントサーバシステムの構成提案を、過去の提案書事例を用いて、要求の具体化から、自動生成された構成案の分析および解釈までを包括的に支援する方式を提案する。ついで、モデル化された企業内構造事例を基にシミュレーションすることで、情報システム導入による効果予測を行い、それに対する解釈としてシミュレーション結果の説明を生成する方式を提案する。

1.2 従来研究

近年、SE に対しクライアントサーバシステム (CSS) の設計要求が急増しており、CSS の設計において SE 業務をサポートする様々なシステム開発手法および支援ツールが開発されている。CSS 開発標準手順の一つである HI-CIRCLE[49] では、「業務プロセスから見直してシステムのライトサイジングを行う」「CSS を早期に開発する」という 2 つの階層を定義し、各階層を反復拡大型プロセスで実現する。また別の標準手順 OKI Framework[50] は、エンドユーザ思考のソリューションを提供するシステム構築体系であり、階層構造の枠組に CSS の構成要素を規定している。また、分散システム構築と運用に必要な諸機能及び手法を提供し、ホストと CSS を連携したオンライン分散システムに再構築するためのミドルウェアとして Dolphine[51] が開発されている。

これらの手法およびツールはシステム開発において効率向上に効果があると考えられるが、システム開発過程で最も重要とされるシステム設計において必要とされる経験的ノウハウの修得および新技術に対応するための知識獲得の支援を行っていない。そのため熟練 SE に対してのサポートは可能であるが、初級 SE に対してはサポートの効果が少ないと考えられる。初級 SE に対して有効なサポートを行うには、経験的なスキルやノウハウを必要とせずに利用できるサポート方法でなければならない。初級 SE に対する最も有効なサポートは、設計案そのものを計算機によって作成することであると考えられる。

次に、自動設計に関する従来研究について述べる。現在のところ、多種のシステム設計を支援するシステムが開発されている [54, 55]。VAX-11/780 計算機システムを構成する R1[56] は、問題解決理論に基づくプロダクションシステムとして実現されている。Dominic[57] は、与えられた問題に対して初期設計を作成し、それに評価と再設計を反復しながら、仕様を満たすまで改良を重ねて最終解を得ている。DSPL[58] は、定型的な設計過程を記述する言語として提案された。この DSPL とよく似た考えで複写機の紙送り機構を設計する PRIDE[59] が開発され実用化された。その他に、事例ベース推論を用いた設計エキスパートシステムが、誘導モータの機械設計に適用されている [60]。

このように、定型的な設計業務においては設計エキスパートシステムは実用化の段階に達している。ところが、CSS の設計においては入力である要求情報が曖昧で不完全であり、そのような曖昧な情報より計算機が設計を行うことは困難である。また、具体的な機能要求が明確になったとしても、その機能を満たすための実現方法は多種多様にわたり、一定の対処方法といったものが存在しない。そのため、要求機能と実現する設計案との対応が複雑である。以上のような理由により、CSS の設計は従来の定型的な設計とは異なり、計算機による実現が困難であるとされている。

本論文で取り上げたもう一方の課題であるシステム導入による効果予測には、諸問題と施策案との因果関係を構造モデル [11] として捉え、これを用いて、各施策案に対する挙動を把握することが有効である。因果関係に基づく挙動の推定を計算機を用いて行な

う試みとして、システムダイナミックス [13][14] に基づくシミュレーション手法が挙げられる。しかし、システムダイナミックスは定量的な数式でモデル化が可能な対象を扱うものであり、定量的に捉えることが困難な対象にはあまり適さない。人工知能の分野において、定性的なモデルに基づいた挙動推定や因果解析を目指す、定性推論に関する研究が盛んに行なわれている [24][25][26]。これらの研究は、本来、定量的に捉えることが可能な対象に対して、定量値を定性値に写像し、複雑な対象における大まかな挙動の把握、不完全な情報からの挙動推定などを実現するものである。このため、組織における諸問題と施策案との因果関係といった本質的に定性的な評価しかできない対象に対しては、定量的空間における特徴点を反映した境界標を設定できない。また、定性的情報を基にしたために発生した曖昧性に対し、物理的制約や定量的な情報などは利用できない。例えば、「消費者のイメージ」や「社員の能力」といった項目があった場合、これらの項目は、はっきりとした基準値がなかったり、段階的な設定が行いにくいなどの理由で、定量化ができず、適用が困難になってくる。

このため、組織における諸問題の因果関係といった、定性的にしか評価できない対象の挙動を知るために、構造モデルをもとに、組織改善案実施後の挙動の予測が可能な類型パターンを用いた定性シミュレーション手法 [35] に関する研究が行なわれている。この構造モデルに基づく定性シミュレーションシステム [36][37] において、結果の出力は、全体の構造モデルとともに、それぞれのノードにおける挙動を、時間経過を横軸に、定性的な状態値を縦軸にとった、グラフ表示でユーザに与えられている。グラフ表示は、ユーザが注目している事項のみの挙動を一目で知ることが可能である。つまり、局所的な結果の挙動を知るには有効であると考えられる。しかし、実施しようとしている施策案が、どのように注目している事柄に影響を及ぼしてくるか、といった因果関係のつながりを知るためには、全体の構造モデルをユーザ自身が解析を行なわなければならない、構造モデルが複雑になるとそのような解析は困難となってくる。

1.3 本研究の方針

本研究では、システム構想の一部であるシステム概要提案および効果予測において、熟練者の経験的な知識が含まれていると考えられる情報システム計画事例をもとにした業務支援および教育のための知識を提供支援方式を提案する。中でも、最も業務知識や提案能力が必要であり初級 SE には業務遂行が困難となる業務である、柔軟な構成が可能な CSS の提案および情報システム導入による企業内の定性的な効果の予測に焦点をあてる。

提案する第一の方式である CSS の提案においては、情報システム計画業務中の各プロセス毎の手法を提案するのではなく、初級者でも手法の学習の手間が少なく業務遂行が可能となるように、業務全体を統合的に支援した CSS 構成設計の自動作成を行う。本研究

では、図 1.1に示す枠組を用いる。要求分析部において、顧客からのシステムに対する曖昧な要求から具体的な要求を推論し、その出力をもとに設計案作成部においてさまざまな制約や、要求に含まれている競合問題を考慮し、提案内容である CSS 構成を作成する。設計案分析説明部では、作成された CSS 構成が複雑に対応づけされた要求を満たしているかどうかを検討し、ユーザに分かりやすいように提示するための説明を作成する。また、知識学習部では構成設計案の作成の際に必要な構成部品決定上の制約条件を見出す。このように、CSS 開発における情報システム計画業務全般を自動的に行うことで、経験のない初級者の SE に対して CSS 提案業務を支援する。

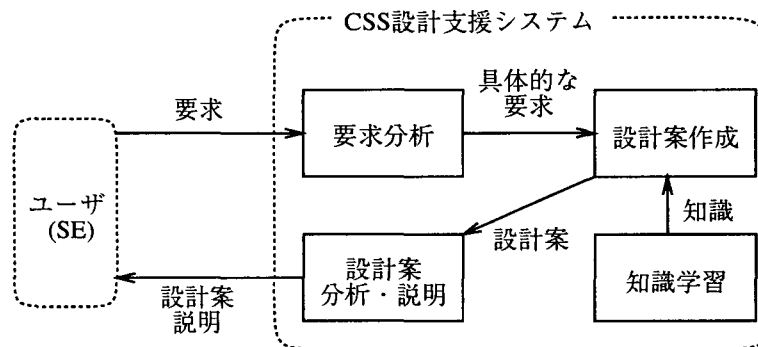


図 1.1: CSS 構成設計支援の枠組

次に、提案する第二の方式では、システム導入に対する効果予測のための、企業内の因果関係を表現した構造モデルを用いた定性シミュレーションを対象とする。この定性シミュレーションでは、結果理解のためにはモデル自身の構造を理解する必要がある。構造モデルが複雑化すると、モデルの理解は困難となり、モデルの作成者である熟練した SE によって解釈されなければならない。そのため、モデルの作成者でない初級者の SE は、シミュレーション結果に対しての理解が困難であったり、間違った因果関係を理解してしまうといった問題がある。そこで、本方式では定性シミュレーション結果を分析することで、実質的な因果関係を抽出し、結果に対する説明としてシナリオを作成する。

1.4 本論文の構成

本論文では、第 2 章以降を以下のように構成する。

第 2 章では、CSS 構成設計業務を 3 つのフェーズに分け、それぞれの業務内容および問題点を明確にし、支援システムにおいてどのような機能が必要となるか分析する。その分析結果に基づき、CSS 向け構成設計支援システム CIDAC(Case based Integrated Design Assist system for CSS) の構成を考案する。

第3章では、CIDACの入力のための機能である対話型要求分析部に焦点をあてる。まず、CIDACにおける要求分析部の役割と共に本機能の出力となる要求仕様について述べる。ついで、要求仕様の一つである業務ツリーを定義し、作成上の問題点を列挙した後、事例を用いた業務ツリー作成方法を示す。また、SE経験の無い被験者による本手法を用いた業務ツリー作成と事例参照による業務ツリー作成との比較実験について述べる。

第4章では、CIDAC中の中心機能である事例ベース型設計案作成部に焦点をあてる。まず、CSS構成設計案作成方式のための入力条件を定義し、事例修正によるCSS構成設計案作成手法について述べる。ついで、事例修正の際に用いられるデータの表現形式を考案し、修正例を示す。

第5章では、CIDACの出力解釈のための機能である設計案分析説明作成部に焦点をあてる。まず、設計案分析説明部のCIDACにおける役割について述べ、説明文作成手法の入出力を定義し、入力である要求分析過程情報、設計案作成過程情報の説明を行う。ついで、どのような設計案分析説明が必要となるか考察した上で、説明文作成方式と共に説明作成例を示す。

第6章では、定性シミュレーション結果の説明文作成方式について述べる。まず、定性シミュレーションに用いる構造モデルについて説明し、シミュレーション手法について述べる。ついで、シミュレーション結果の説明であるシナリオ作成のための本質的な因果関係抽出方法について述べた後、自然言語の説明文として出力するための文章化について述べる。また、作成されるシナリオの因果関係の妥当性を評価するために、本手法による因果関係の抽出と、一定の因果関係の利用との比較実験を示す。

第7章では、本研究で得られた成果を要約した後、今後に残された課題について述べる。

第 2 章

クライアントサーバシステム構成設計支援 システム CIDAC

2.1 まえがき

本章ではクライアントサーバシステム構成設計業務分析を行い、支援の枠組 [74][75][76][78] を提案する。

近年、システムエンジニア (SE) に対しクライアントサーバシステム (CSS) の設計要求が急増している。CSS の設計において、SE は経験的なスキルが通常のシステム設計以上に要求される。また、顧客からの要求情報入手作業は容易ではない。さらに、技術の進歩が急速であり、新技術に対応するための知識も要求される。以上のような問題点を含んでいるため、SE にとって CSS 設計の技術獲得は困難であり、計算機による支援が要求される。

現在、CSS の設計において SE 業務をサポートする様々なシステム開発手法および支援ツールが開発されている [49][50][51][52][53]。これらの手法およびツールはシステム開発において効率向上に効果があると考えられる。しかしながら、システム開発過程の第 1 段階であり最も重要とされるシステム設計 [1][9] において必要とされる経験的ノウハウの修得および新技術に対応するための知識獲得の支援を行っていない。初級 SE に対してサポートを行うためには、経験的なスキルやノウハウを必要とせずに利用できるサポート方法でなければならない。

初級者に対する有効なサポート方法として、計算機による CSS 構成の自動設計が挙げられる。計算機による自動設計を考えた場合、熟練者のノウハウを計算機上に実現しなければならない。しかしながら、熟練者が持つノウハウは、経験的に得られるものが多くルール化が困難である。そこで、設計に関する知識をルール化するのではなく、熟練者によって設計された過去の事例を用い、事例ベース推論により自動設計する。

本章では初級SEでも顧客のニーズをもとに、熟練SEに近いCSS構成の設計が行えるよう、過去の設計事例を用いたCSS構成設計の支援システム (CIDAC: Case based Integrated Design Assist system for CSS)[76] の構想を示す。

以下、第 2.2 節では CSS 構成設計について、第 2.3 節 CSS 構成設計業務の分析結果について説明する。第 2.4 節に支援アプローチを示した後、第 2.5 節で CIDAC の構成を示す。

2.2 CSS 構成設計

CSS は、ネットワーク上に分割されたクライアントとサーバが一体になって、全体で 1 つの処理を実行する分散処理システムである。CSS では、ハードウェアやソフトウェアを自由に組み合わせることができ柔軟なシステム構造が実現される。図 2.1 に CSS の構成例を示す。

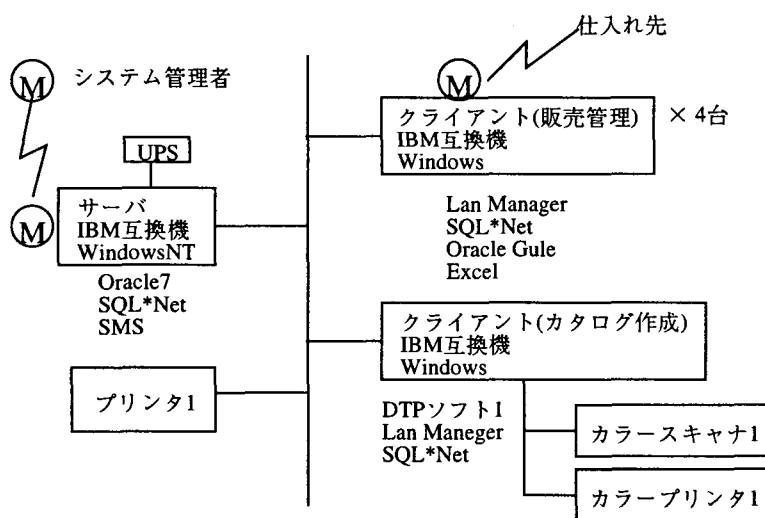


図 2.1: CSS 構成例

CSS 構成設計業務は図 2.2 に示すように、「要求分析」「設計案作成」「設計案の分析」の 3 つのフェーズからなる。「要求分析」では、顧客との対話により現状や要求を SE が把握し、要求仕様を作成する。「設計案作成」では、要求仕様をもとに実際のソフトウェアやハードウェアの構成を考え、「設計案の分析」において作成した構成が適当であるか分析を行い、顧客に説明するための提案書を作成する。この提案書をもとに再び顧客との意見交換が行われる。

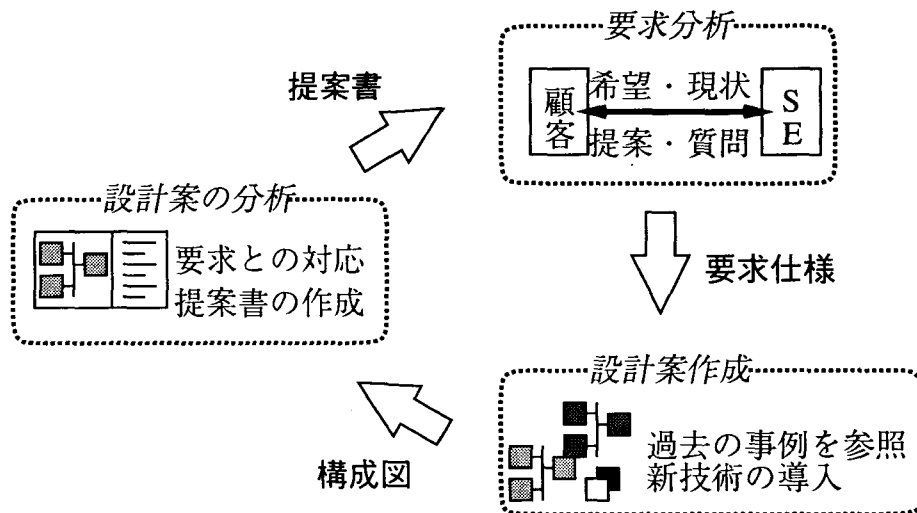


図 2.2: CSS 構成設計業務

2.3 CSS 構成設計業務の分析

実際の CSS 構成設計業務において、SE の活躍が期待される工程のアンケート結果として図 2.3 が報告されている。この結果では、CSS の設計を行うことよりも、コンサルティングやシステム化計画、システム分析に対する期待が大きいことがわかる。従って、顧客との接点となる要求分析および設計案の分析作業は、設計案の作成よりも重視されているといえる [27]。

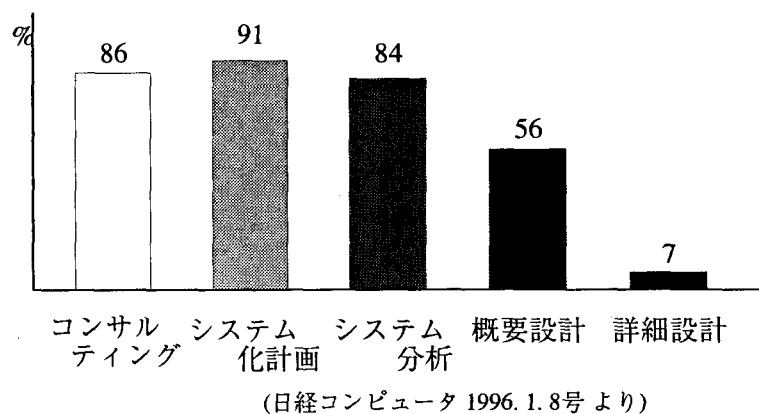


図 2.3: CSS 構築において SE の活躍が期待される工程のアンケート結果

また、要求分析や設計案の分析作業のほうがより経験的なノウハウが必要となるために、初級 SE にとって困難な作業となっており、より熟練者に頼っている [33]。そのため、CSS 構成の自動設計による SE 支援の実現には、設計案作成作業のみの支援ではなく、CSS 構

成設計業務の一連の作業を統合的に支援することが望ましい。

CSS 構成設計業務の統合的な支援を目指すために、一連の各作業に対しての分析結果として、以下に各工程作業の内容および問題点について述べる。

2.3.1 要求分析

顧客からのシステムへの要求を分析する作業は、システム構築の最初の段階である。正確に要求を認識しなければ、顧客のニーズにあったシステムの構築は不可能であり、要求分析作業はどのようなシステム設計の際においても重要となってくる [31, 32]。

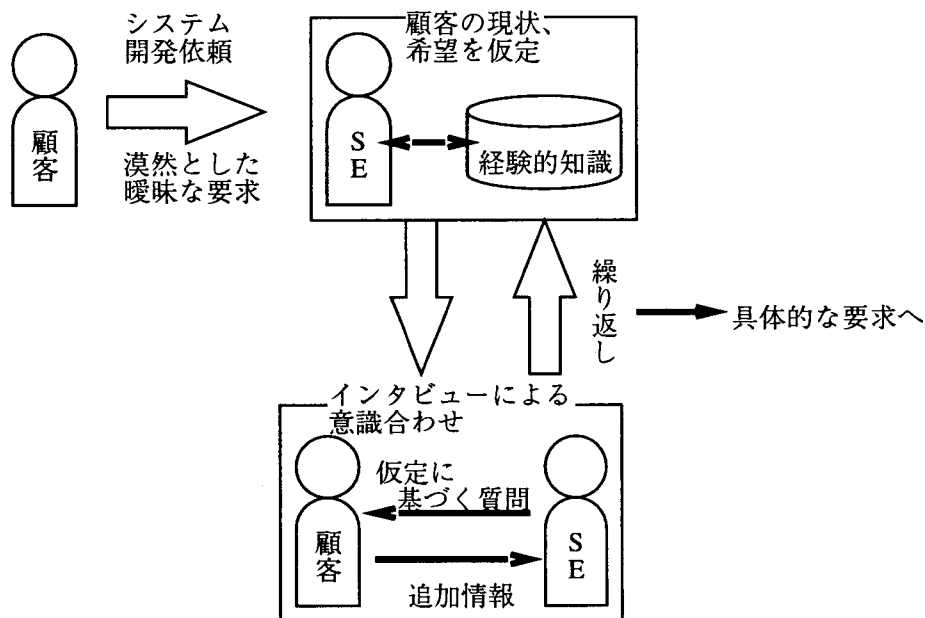


図 2.4: 要求分析の過程

最初に顧客からシステム構築の要求がなされる。しかしながら、最初に顧客から出される要求は漠然としたものであり、明確な目的や機能といったものは含まれていることは稀である。そのために、通常 SE は追加情報や明確な情報を得るために、顧客に対してインタビューを行う。SE は経験的なノウハウによって顧客の現状を予想し、顧客が答えやすい適切な質問を行い、必要な情報を入手する。得られた情報より顧客の業務や現状、希望を理解し、どの様な機能が必要となるかといった分析を行い、要求仕様へとまとめる [28]。

この作業における問題点をまとめると以下ようになる。

- 顧客から得られる情報は曖昧で不完全である
- 情報収集のためには適切な質問が必要である
- 不完全な情報に対して、適切な仮定を置き要求仕様を作成する必要がある

2.3.2 設計案作成

SE は作成した要求仕様をもとにシステム構成の設計案作成を行う。設計案の作成では、ソフトウェアやハードウェアの自由な組合せが可能であるため、過去の経験よりシステムをパターン化して捉える。パターンをもとに、要求仕様に沿う様に修正することで設計案を作成する。修正の際には技術動向に応じて修正を施すことで、新技術を取り入れる。しかし、事例を要求に応じて修正するだけでは、設計案全体の整合性がとれなくなる恐れがある。SE はこのような整合性が成立するかどうかを考慮して要求の緩和のような柔軟な対応をとっていると考えられる [50]。

この作業における問題点としては、以下の 3 点が挙げられる。第一に SE は個人ベースの活動が多いために、過去の有効な設計案や知識を個人レベルのみでしか利用できないことがある。殊に初級 SE には、参考にする過去の事例を多く持ち合わせていないために、適切な設計案作成が困難である。第二に技術進歩に対応するためには多大な努力が要求されることがある。近年技術の進歩が早く、次々と新たなソフトウェアやハードウェアが開発されている。SE は最新の技術動向を踏まえた設計案を作成するために、これら新製品を設計案に取り入れなければならない。そのためには、SE には新製品に関する知識を常に吸収していくことが要求される。このため、SE は通常の設計業務以外に、技術動向に関する知識を獲得する努力をしなければならない。しかしながら、新製品・技術の種類は膨大であり、全ての製品に関する知識を獲得することは多くの時間を要し、技術動向に的確に対応していくことは困難である。最後に、顧客からの要求には実現不可能であったり、コスト上の制約があるために、實際上すべての要求を満たすことは困難な場合がある。そのために、SE はこのような競合問題を顧客が納得のいく形で要求仕様を変更するなどして解消しなければならず、問題の解決には経験的なノウハウを必要とする [29]。

2.3.3 設計案の分析

作成した設計案を基に顧客と話し合い、新たな情報を得たり要求の訂正を行う。しかしながら、情報技術や CSS 構成に関する知識をほとんど持ち合わせていないと考えられる顧客に対し、システム構成を示した設計案のみを見せるだけでは、意見や同意を求めることは不可能である。そのため、設計案に関する説明として提案書を作成しなければならない。提案書は設計案の妥当性を示し、顧客に設計案を納得してもらうための資料である。この提案書には

- 設計案がどの様に要求を満たしているか
- どの要求が満たせなかったか (なぜ満たせないのか)
- 要求情報不足のために SE 側が用いた仮定について

- 他の設計案との比較による提案した設計案の妥当性

といった内容が記述される [30]。

この作業における問題点としては、以下の3点が挙げられる。まず、要求とその実現方法の対応が複雑さが挙げられる。CSS 構成では、要求される機能とその実現方法が1対1で対応しておらず、その対応が複雑であるため、説明が難しい。次に、SE がおいた仮定を見逃し易いことが挙げられる。SE は設計案作成の前段階で不完全な要求については仮定をおいたり、変更を加えることで情報を補っている。この仮定や変更は、SE の常識を基にしていることが多いが、CSS に関する技術的な知識が少ない顧客には理解できない場合、顧客の意図にそぐわない場合がある。そのために、SE が行った仮定や変更を明確にし、その理由を説明しなければならない。最後に、初級 SE に対しては構成中の各部分の意味を理解させる必要があることが挙げられる。

2.4 支援アプローチ

前節で述べたように、CSS 構成設計業務にはさまざまな問題点が含まれている。これらの問題点を踏まえた上で、支援システムどのような機能が必要であるかを図 2.5 のように考える。

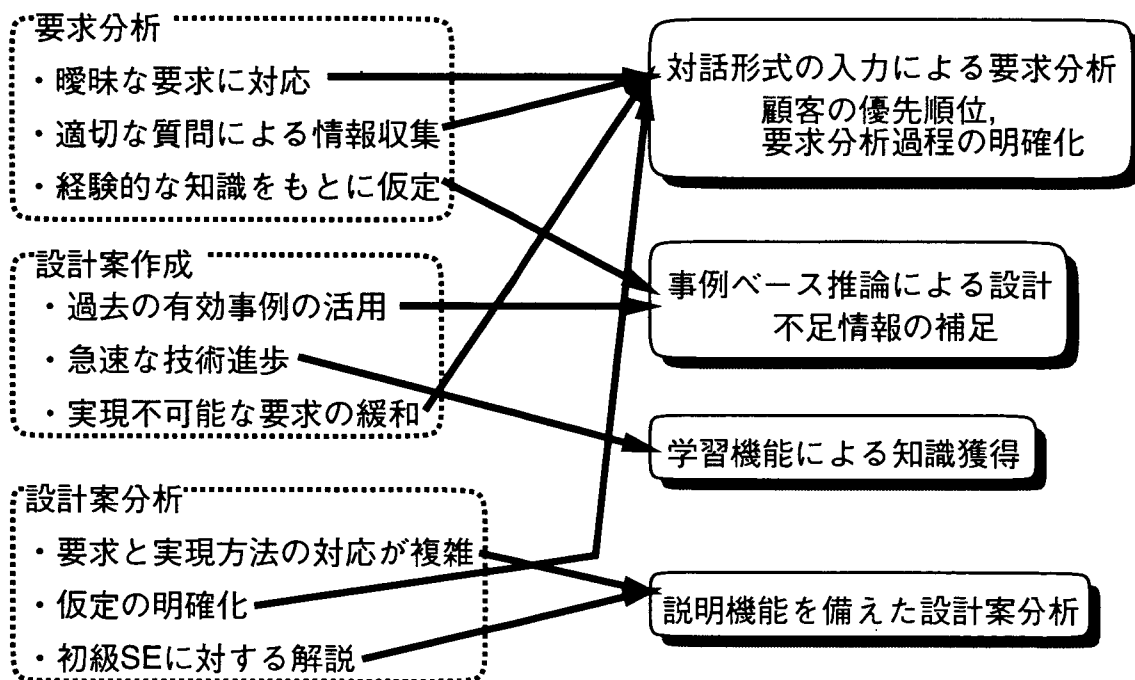


図 2.5: システムに求められる機能

要求分析において問題となる曖昧な要求への対応、適切な情報収集による情報収集に対応するため対話形式の入力による要求分析部を実現する。適切な質問を提示し対話形式の入力を行うことで、ユーザと共に曖昧な要求を具体化する。また、設計案作成において問題となる実現不可能な要求の緩和を行うために、要求分析において予め顧客の要求の優先順位を設定しておく。さらに、設計案分析の問題である仮定の明確化に対応するために、要求分析過程を明確に記述しておくことで、どのような仮定をおいたのか記録する。

次に、設計案作成時の問題である過去の有効事例の活用に対応するために、設計案作成部では事例ベース推論を用いる。事例ベースを用いることで、要求分析時の問題である経験的な知識に基づく仮定を実現する。

また、設計案作成時の問題である急速な技術進歩に対応するため、学習機能を設け、技術の傾向に関する知識の獲得を試み、設計案の作成に反映させる。

設計案分析時の問題である、要求と実現方法の対応が複雑であること、および初級 SE に対する解説が必要であることへの対応は、説明機能を備えた設計案分析部において行う。要求分析および設計案作成の過程情報を対応させ、要求とその実現方法の対応関係を分析する。

2.5 CIDAC の構成

前節の考えをもとに、図 2.6 のような統合的な CSS 構成設計業務の支援システム CIDAC の構成を提案する。

対話的要求分析部では、曖昧な要求に対応できるよう要求事例を用いて顧客の現状を推論し、適切な質問や不足情報に対する仮定の作成を試みる。さらに、対話的に情報を収集

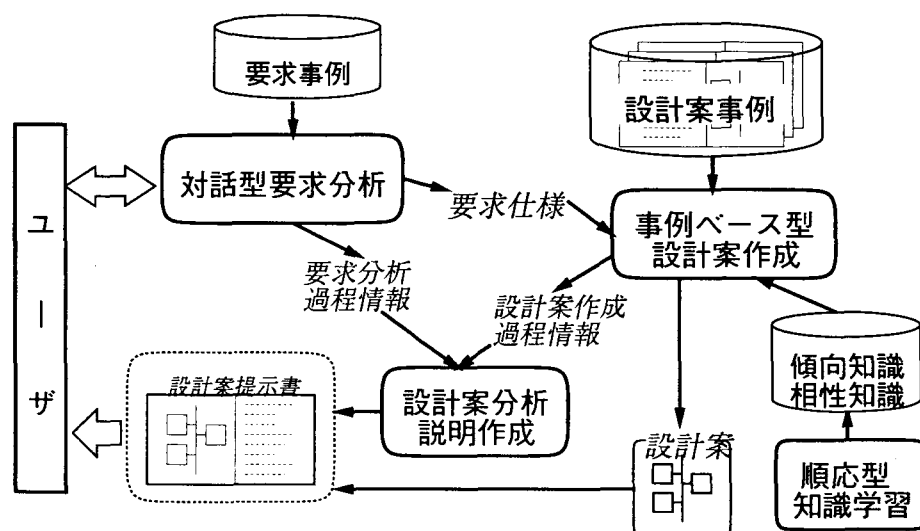


図 2.6: CIDAC の構成

することにより、顧客の各要求に対する優先順位を捉えることで、実現不可能な要求の緩和を試みる。この要求分析部では、設計案作成部に対して要求仕様を、設計案分析説明作成部に対して要求分析過程情報を出力する。要求仕様とは設計案作成に必要な情報をまとめたものであり、提案されるシステムがどのような機能を果たすか、どのような規模のものであるかといった内容が形式的な表やグラフとして記述される。具体的には図 2.7 に示す業務内容を示した業務ツリー、CSS の基本的な特徴として、処理パターンやどのようなサーバ、どのようなクライアント、どのような機能が必要であることを記述した要求特徴データ、その他の細かな情報を記述した詳細要求である。

要求分析過程情報は図 2.8 に示すように、複雑な要求分析の過程を明確な形式で捉えるための木構造のモデルであり、漠然とした要求から具体的な要求へと変換される過程が、変換のための要因と共に記述される。

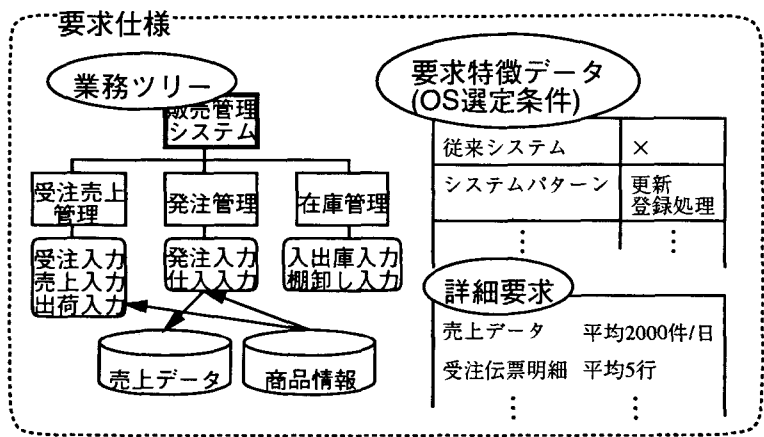


図 2.7: CIDAC における要求仕様

事例ベース型設計案作成部では、SE がもつ経験的な知識として事例ベースを利用する。企業内もしくは雑誌等に公表されている過去の有効な事例である CSS 構成設計案に対し、要求分析部の出力である要求仕様を基に、修正を加えて設計案を作成する。要求仕様中の業務ツリーおよび要求特徴データ中のシステムパターンを用いて事例を検索する。次に、初期修正として要求仕様の要求特徴データ中のサーバやクライアントの機能に関するデータに沿うように修正を施す。初期修正においては、機能のみの実現を考慮しているため、実際にはコスト面や性能面において実現が不可能な設計案が作成される。初期修正結果を基に、詳細要求を用いて顧客が何を重視しているかということに注目し、要求の条件緩和を行い実現可能な設計案へと修正し図 2.1 に示すような CSS 構成の設計案を作成する。各修正の際には、技術動向や実現可能性を高めるために傾向知識、ソフトウェアやハードウェアの組合せの相性知識、部品 DB より CSS を構成するソフトウェアやハードウェアといった部品の情報を用いる。また、この設計案作成過程の情報を設計案分析説明作成部

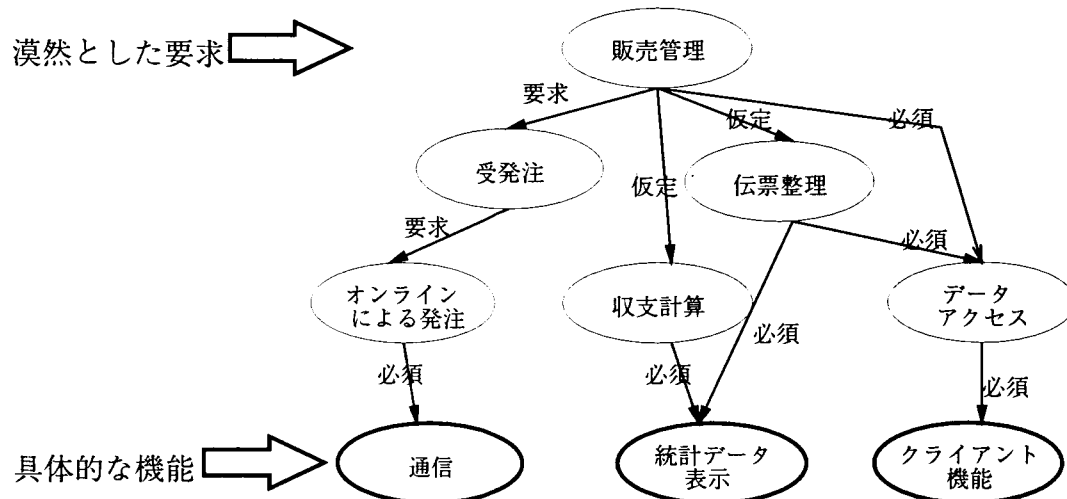


図 2.8: 要求分析過程情報の例

に渡す。設計案作成過程情報は図 2.9に示すような、木構造の構造モデルであり、各アー
クには部品決定理由が付属されており、設計案作成過程の履歴情報となる。

設計案分析説明作成部では、要求分析過程情報、設計案作成過程情報をもとに、顧客に
提示するための説明を作成すると同時に、必要に応じてシステムの利用者である初級 SE
に対して、「販売管理において、通信の実現のために、モデム：モデム 1、相手先：仕入
れ先が設計案に含まれています。」のような構成要素の説明を作成する。また、SE が学習
を可能とするために、設計案中の各部品が決定された根拠提示するために、「OS は、ソ
フト：Oracle Glue, Excel, Lan Maneger, SQL* Net のために Windows と決定しました。」
のような説明文も必要に応じて作成する。

順応型知識学習部では、設計案事例より最近の傾向やソフトウェアとハードウェア間の
相性を抽出し知識として蓄える。相性知識はソフト・ハード間の相性を保証する知識であ
る。ソフト・ハード間の相性は、実際に稼働しているかどうかによってのみ判定できるも
のである。そのため、過去の事例をもとに相性を割り出すほかはない。よって、相性知識
は、事例ベースをもとにし、1つの事例中に含まれていれば相性が良いものとされ、グ
ループとして獲得される。このグルーピングを複数の事例に跨って行うことで、より相性
のグループを明確な形式で抽出することができる。傾向知識は、設計案の技術的な傾向を
踏まえた修正を可能にするための知識である。この傾向知識には大きく分けて二通りあ
る。1つは、要求中に必要な情報が欠如していた場合、適切な仮定をおくための知識であ
る。もう1つは、要求中の内容とは違う場合でも、最近の技術動向から考えて適切である
と考えられる修正を行うための知識である。前者は、要求仕様中に含まれていない要件を
持つ事例を集め。その要件を如何に処理しているのかを分析することによって知識として
得ることが可能である。後者は、事例を時間によってソートし、処理方法の時間的変化の

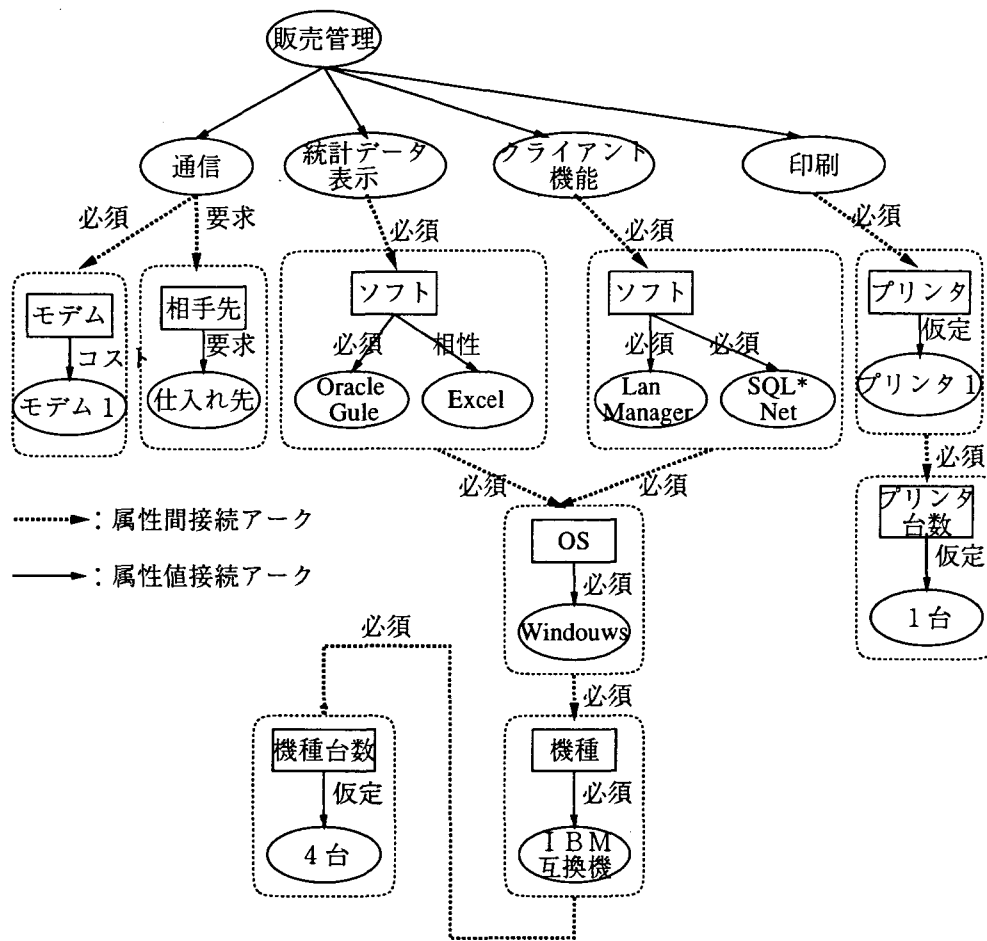


図 2.9: 設計案作成過程情報の例

有無を統計的に調べる。時間的変化が確認されれば最近の処理方法を、または、変化がなければ全体の処理方法を知識として得ることが可能であると考えている。

2.6 まとめ

本章では、CSS 構成設計業務における各作業を分析し、どのような機能が必要となるかを明確にした上で、CSS 構成設計作業の効率向上および初級 SE に対しての設計技術獲得支援を目的とした CSS 向け構成設計支援システム CIDAC の構想を示した。

CIDAC は、曖昧な要求を具体化する要求分析、事例ベース推論によって設計案を作成する設計案作成、出力結果の設計案を SE が理解しやすいように説明を作成する設計案分析説明作成、技術の進歩に対応するための知識を獲得する知識学習の 4 つの部分から構成することで、初級 SE に対する CSS 構成設計支援を目指す。

第 3 章

要求分析部における業務ツリー作成

3.1 まえがき

本章では前章で説明した CSS 構成設計支援システム CIDAC の要求分析部の機能の一つである業務ツリー作成 [79][80] について述べる。

要求分析部では、SE が顧客から収集した要求情報を要求仕様へとまとめる。要求仕様とは、構成案を考える際に用いる情報を記述したものであり、構成案の設計方法や SE の好みにより一般に記述方法は様々である [31] [32]。CIDAC で用いる要求仕様の一つである「業務ツリー」とは、システム導入対象の業務内容をツリー形式で表現したものである。これによって設計するシステムのサーバの能力の割り振りや各マシンにおけるデータのやり取り状況などの設計における大枠を決定する重要な要求情報である。

業務ツリーは、顧客が計算機上で実行したい業務の内容をツリー形式で表現したものであり、業務ツリー作成のためには顧客の業務内容を理解しなければならない。また、業務ツリーは同種の業務であれば類似しているが、SE は顧客毎の業務体系の違いによる差異に気付く必要がある。そのため、顧客からの情報収集は欠くことができず、業務に対する基礎知識と共に顧客との対話のスキルも要求される。つまり、業務ツリー作成では、一般的に要求分析と同様に、情報収集能力や顧客毎の柔軟な対応が求められる [27][28][29]。以上のような業務ツリー作成上での問題点があるため、計算機上での作成を考えた場合、熟練した SE の経験的な知識を一定形式の知識として蓄えることは困難である。しかし、対象業務が同じであれば業務ツリーが類似しているため、過去の事例を参照することが有効である。そこで、本章においては事例ベースを用いた業務ツリーの作成を行う。

知識化が困難な対象の問題解決を目指した事例ベース推論では、ある問題に対して、過去の類似した成功事例もしくは失敗事例を修正することにより解を導き出している。この事例ベース推論を用いた研究は、現在までに広く為されている [60][61][62][63][64][65][66][67][68]。このような事例ベース推論においては、事例に対する修正方法が重要な要素の一つとなる。対話型事例ベース推論 [69] では、事例修正等の際に必要な知識が不完全である場

合に、ユーザとの対話によって知識を補完することにより、解を導き出している。また、一つの事例に対して修正を施すのではなく、断片的に得られた複数の事例を組み合わせることにより一つの解を見出すシステムとして CELIA[70] がある。この組み合わせには、事例間のつながり情報を用いている。各事例において、どの事例と繋がりがあるのかという情報を持たせておくことで、事例を繋ぎ合わせ一つの解決案を見出している。

ところが、今回対象としている業務ツリーでは、ユーザによって臨機応変な対応をしなければならないため、一定の修正知識を持つことは不可能である。また、知識を用いず複数の事例より解を見出す方針を考えた場合、事例間の関連情報が必要となる。しかしながら、個々の業務ツリーの事例は独立して得られるものであり、どの事例と繋がりがあるのかといった情報は得られない。つまり、事例間にある一定の関係を位置付けることができないため、事例ベース中に関連情報を付加させることは不可能である。そのため、事例を利用するのみで知識を用いず、さらに事例間の関連情報を必要としない修正方法が必要となる。

本章では、業務ツリーのための情報を格納した事例ベースを基に対話形式により修正を行うことで特定の知識を必要としない業務ツリー作成方式を提案する。キーワードをもとに検索された複数の事例を用い、ユーザに的確・不的確の指摘を委ね、その情報をもとに事例の接続を行う。接続には業務ツリー中のノード内容のマッチングをとり、重複を防ぎつつ繋ぎ合せていく。また、マッチングにより接続が不可能な場合には、ユーザからさらなる情報を得るために、システムが接続情報を得るためのみに作成した業務ツリー（候補事例）をユーザに提示する。本方式による業務ツリーの作成では、ユーザの意見に委ねられる部分が多いが、ユーザの操作は的確・不的確の指摘のみという簡単な操作である。また、初級 SE にとって的確・不的確が不明である場合にも、顧客に対して二者選択といった答えやすい質問が可能である。

以下、第 3.2 節では要求仕様の記述について、第 3.3 節業務ツリー作成方式について説明する。第 3.4 節に事例の結合例を示した後、第 3.5 節で比較実験を示す。

3.2 業務ツリーの概要

3.2.1 要求仕様の記述

要求分析部では顧客から対話的に要求情報を収集し、設計案作成部に渡すための要求仕様を作成する。

顧客からの要求は、図 3.1 に示すように、業務システム要求、CSS のネットワーク要求、詳細要求の 3 つに分類できる。業務システム要求は顧客の業務プロセスに関係する。CSS のネットワーク要求は CSS の基本的な構成を決定条件とする要求であり、データトラフィックやセキュリティといった CSS 構成部品に直接関係する要求である。詳細要求

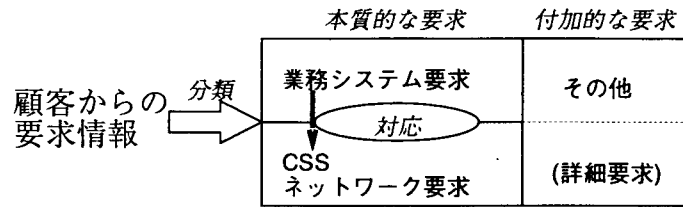


図 3.1: 要求情報の分類

はある部品を加えることにより満足できるような要求である。

業務システム要求は直接 CSS 構成に反映されない。そのため、業務システムと CSS ネットワークとの対応させるためのデータが必要となる。このデータを表現するために、業務システム要求として業務構造をシステムティックに表現した業務ツリーを用意する。CSS ネットワーク要求は曖昧であったり情報不足であっては設計案の作成ができない。そのため、このデータは一定項目に対する定量的もしくは定性的な値を記述した表として用意する。詳細要求は一定の項目を持たない柔軟な表として表現する。以上より、要求仕様として図 3.2 に示す 3 種類のデータを用いる。業務システム要求を表現した業務ツリー、CSS ネットワーク要求を表現した要求特徴データおよび詳細要求を用いる。

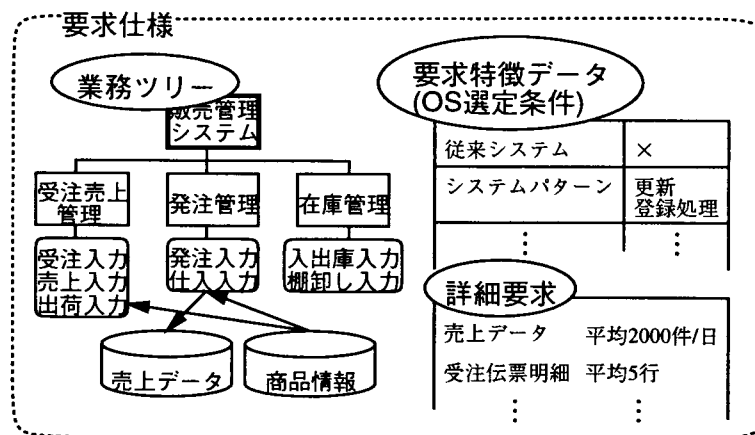


図 3.2: CIDAC における要求仕様

これらの要求仕様は次機能の設計案作成部において、以下のように用いられる。まず、設計案事例の検索には業務内容を示した業務ツリー、および要求特徴データ中のシステムパターンが用いられ、それらに対する初期修正として要求特徴データ中のサーバやクライアントに必要な機能に関する内容が用いられる。詳細要求は、初期修正後の再修正時に用いられる。また、要求分析部では設計案分析説明作成部に渡すための要求分析過程情報も作成する。これは、要求仕様中の要求特徴データおよび詳細要求を作成する際の、ユーザ

との対話の過程を木構造で表現したデータである。

本章では要求情報収集の対話の手がかりとして業務内容を把握するため、第一に作成される CIDAC の要求仕様である業務ツリーの作成について述べる。

3.2.2 業務ツリーの定義

業務ツリーとは、図 3.3 に示すようなシステム導入対象の業務内容をツリー形式で表現したものである。これによって設計するシステムのサーバの能力の割り振りや各マシンにおけるデータのやり取り状況などの設計における大枠を決定する重要な要求情報である。

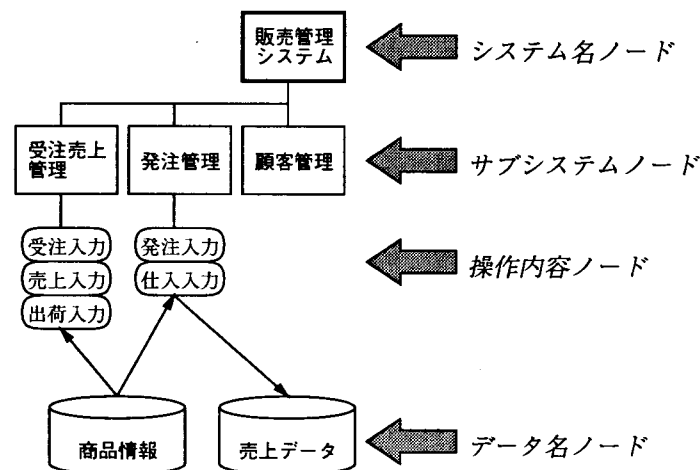


図 3.3: 業務ツリーの例

業務ツリーは、以下に示す 4 種類のノードおよびデータのやりとりを示す有向アーク、機能の具体化を示す無向アークからなる。

システム名ノード：まとまった一つの役割を示す（基本的に一つのサーバとなるもの）

サブシステムノード：システム内にどのような機能を持たせるかを示したノード

操作内容ノード：一つの機能において、ユーザがどのような操作を行うかを示したノード

データ名ノード：取り扱うデータを示したノード

3.2.3 業務ツリー作成における問題

業務ツリー作成においては以下の 4 点の問題点がある。

システム導入対象業務をツリー構造で表現した業務ツリー作成では、顧客の業務内容をよく理解しなければそれらを表現できない。そのため、経験の少ない初級 SE には、様々な業務に対する理解を深めるために多くの努力が必要となる。

次に、通常業務名等が同じであればその業務内容は同じであると考えられる。しかしながら、それらをシステム上に実装するためには、内容のみでなくどのような構成で業務が行われているかといった業務体系までを理解しなければならない。同様の業務であっても、顧客毎によってその業務体系が違っており、それらの差を理解しなければならない。

3つ目に、顧客から得られる業務についての情報は、対話的に会話で得られることが多い。それらの情報を構造的に捉えられなければ、業務ツリーとして表現する事ができない。つまりは、顧客からの話を聞き、それらを理解し全体を考えた上で、自分が持つ知識を活かさなければならない。これらの作業は初級 SE には困難である。

最後に、業務ツリーを作成し記述することで導入するシステムが何を行うか、どのように処理を分散させるかを決定する。つまりは、導入する新システムがどのようなものかという提案となる。このため、顧客にはどのようなシステムが必要となるかを、顧客からの情報により考え出さなければならない。

以上を示した理由により、経験的な知識を要するため初級 SE には業務ツリーの作成が困難であり、CIDAC への入力として直接この形式は望めない。そのため、初級 SE にも操作可能な形式により業務ツリーの自動作成を行う必要がある。

3.3 業務ツリー作成方式

3.3.1 作成アプローチ

同じ業務に関する業務ツリーは類似しているので、新しい業務ツリーを作成する際に事例として過去の業務ツリーを参考にする。

過去の類似した成功もしくは失敗事例を修正することで問題を解く事例ベース推論は、一定のルールを形成することが難しい問題を対象したときに有効であるとされている。事例ベース推論において、事例の修正方法は重要となる。とりわけ事例を修正するために必要なルールやデータが無い場合、対話型的事例ベース推論ではユーザから修正に関する情報を得ることで、ルールやデータの代替を行う。今回対象としている業務ツリー作成においても修正のための十分なルールが存在しないので、対話による事例修正を試みる。ただし、本システムのユーザは初級 SE であるので、修正の際に用いた事例以上の情報を加えることは難しい。

この問題に対応するために、本方式では図 3.4 に示すように、最類似事例のみを用いるのではなく、複数の類似事例に対し、ユーザからの意見を対話的に求めることで、適用可能な断片的な事例を作成し、それらを結合させる方針をとる。また、対話形式の修正では容易に修正情報の入力を行えるよう、事例中の各部分に対して的確・不的確といった 2 者選択形式による入力を考える。2 者選択形式であれば、ユーザ自身が不明な場合でも、顧客に対して容易に質問ができるため、初級 SE でも情報の入手が容易である。

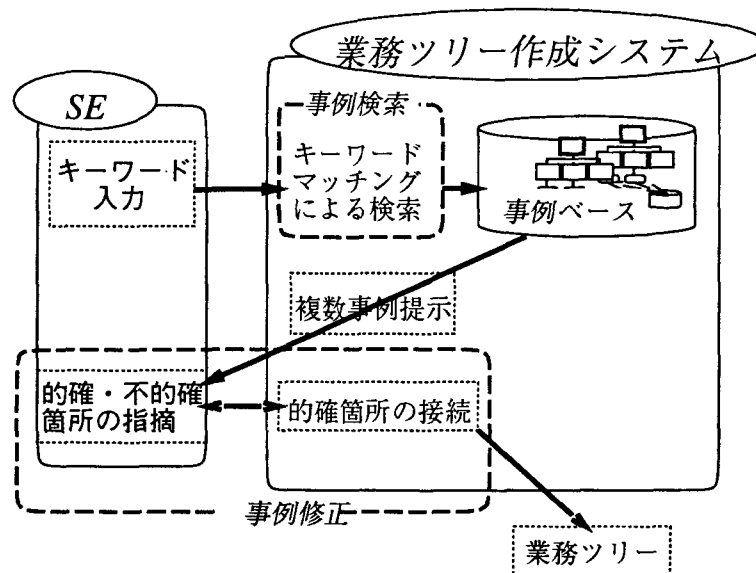


図 3.4: 業務ツリー作成概要

複数事例を基に得られた断片的な事例を結合させる方針をとった場合、その事例の結合方法が重要となる。事例の結合は、ノード中の語句のマッチングにより実現する。しかしながら、語句を単なる文字列として扱いマッチングをとることはできない。業務ツリー中のシステム名ノード、サブシステムノードおよび操作ノードには依存関係があり、上位ノードは下位ノードに影響を与える。つまり、文字列としての語句が同じであっても、上位ノードの影響を受ける場合、同意であるとは言い切れない。また、ノードの語句がマッチングされない事例に対しては、上位ノードとの依存関係が不明であり、部分グラフとして他の事例と繋ぎあわせることは不適當であり、結合箇所が特定できない。

そこで、本方式ではノード間の依存関係を踏まえたマッチングを行うため、最も的確箇所の多い事例（基本事例）に対し、他の断片的な事例（部分事例）をそれぞれ上位ノードからマッチングをとる。上位ノードにおいて、マッチングがとられなければ、その下位ノードについてはマッチングをとる対象としない。また、結合箇所が不明な事例に対しては、結合可能な箇所の選択肢をユーザに提示することにより、依存関係に関する情報を入手し、結合箇所を決定する。以上のようにして、事例を結合することで、ノード間の依存関係を考慮した業務ツリーの作成を行う。以下に、本方式で用いる事例ベースについて、また事例の結合方法について述べる。

3.3.2 事例ベース

事例とは過去に作成された業務ツリーである。事例ベースにはこれらの過去の事例を格納する。事例の内容および格納形式を考えた場合、その検索方法を考慮する必要がある。

そこで、最初に検索方法について述べた後、事例の格納方法について述べる。

業務ツリーにおける類似性とは、ノードの内容が一致するかどうかで判断する。業務ツリー中のノードの内容が一致しておれば、2つのツリーは類似していると考え、業務名等のキーワードのマッチングにより検索を行う。そのため、事例ベース中に検索のためのキーワードとして、業務名や業種名をツリー以外に付加しておく必要がある。以上より、事例として格納しておくデータは、過去の作成した業務ツリーと、その業務ツリーが対象としている業務名、業種名等のキーワードである。

表 3.1: 業務ツリーの格納データ形式

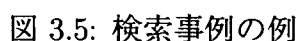
導入部署名	
業種名	
⋮	
ヘッダ終了記号	
データ名 1	識別コード
データ名 2	識別コード
⋮	⋮
データ名終了記号	
参照ノード識別コード	被参照ノード識別コード
⋮	⋮
アーク指定終了記号	
システム名 1	
サブシステム名 1	識別コード
操作内容 1	
操作内容 2	
⋮	
サブシステム終了記号	
サブシステム名 2	識別コード
⋮	
システム終了記号	
システム名 2	
⋮	
業務ツリー終了記号	

図 3.3に示すような業務ツリーをデータとして格納するには、ツリーをノードデータおよびノード間を繋ぐアークデータからなる一つのテーブルとして扱う必要がある。ノードデータは、そのノードの種類(システム名ノード等)と、実際にノード内に記述される内容、ノードの識別番号を格納する。アークデータは、接続しているノードの識別番号を格納する。これらのノードデータおよびアークデータに、キーワードデータを加えたものが1つの事例である。これらのことから、事例の格納形式は表 3.1のような形式とする。図 3.3の業務ツリーの例を事例ベース中に格納した形式で表現すると、表 3.2のようになる。

表 3.2: 業務ツリーのデータ構造例

営業	
KEY END	
商品情報	1
売上データ	2
DATA END	
1	10
1	11
11	2
ARC END	
販売管理	
受注売上管理	10
受注入力	
売上入力	
出荷入力	
SUBSYSTEM END	
発注管理	11
発注入力	
仕入入力	
SUBSYSTEM END	
顧客管理	12
SUBSYSTEM END	
SYSTEM END	
TREE END	

現在 CIDAC の事例ベース中には営業部門の 21 件の事例が格納されている。この事例ベースに対して、「営業」というキーワード入力によって検索を行うと、図 3.5 に示す 5 件が検索された。



3.3.3 事例の結合

検索された複数の事例に対して、顧客の要求を元に、それぞれの的確箇所、不的確箇所をSEに指定してもらう。それらの情報をもとに事例を結合させることで業務ツリーを作成する。

まず、的確・不的確の指定がされた事例を、結合に用いるための基本事例および部分事例へと整形する。基本事例、部分事例の定義は以下の通りである。

基本事例： 最も的確箇所の多い事例の不的確箇所を除去したツリー

部分事例： 基本事例となった事例以外の事例中の的確箇所のみを残したツリー

基本事例に対して、それぞれの部分事例を順次結合させる。結合には、ノード内容のマッチングをとり同意と見なされるノードを重ねる。この際に、操作内容ノードはサブシステムノードに、サブシステムノードはシステム名ノードに依存関係があるため、部分事例は上位ノードから順次マッチングをとる。つまり、図 3.6 の例に示すように、ある部分事例において操作内容ノードにおいてマッチングが可能であっても、その上位ノードであるサブシステム名ノードが存在した場合、上位ノード同士にマッチングがとられなければ同意とみなすことができない。

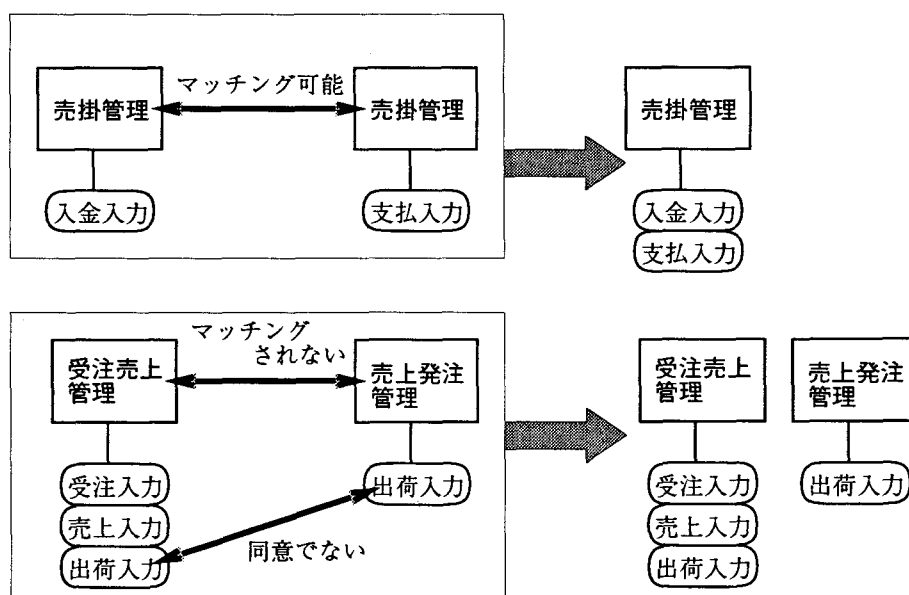


図 3.6: ノードの比較

このように基本事例と部分事例の結合を試みると、複数箇所においてマッチングが行われたり、マッチングが行われない場合があり、結合処理ができない事例が存在する。そのため、再びSEからの情報収集を試みる。このSEからの情報収集においても、SEにとっ

て容易な入力方法でなければならない。そこで、具体的にどのような選択肢があるのか、実際の例を示して意見を求める。つまり、結合可能なすべての箇所に対して結合させたツリー（候補事例）を提示し、再び的確・不的確の入力を求める。そのようにして得られた情報をもとに、再びマッチングを行い結合させる。また、結合の完了した業務ツリーにおいても、必要な業務内容や、サブシステムノードとデータ名ノード間でのデータのやりとりを表す有効アークが不足している場合を考え、最終的な修正を SE に行ってもらおう。

3.4 事例結合例

図 3.5 の検索結果を例に挙げて説明する。まず、図 3.5 の検索結果に対して、図 3.7 のような的確箇所・不的確箇所の指定がされたとする。これらを基に基本事例および部分事例を作成すると図 3.8 となる。

1. システム名ノードのマッチング

システム名ノードの「販売管理システム」がマッチングされるため、部分事例 2 が基本事例と結合される。次に、部分事例 1 ではシステム名ノードが存在するが、マッチングがとられないため、先ほどのデータ名ノードを介しての結合のみとされる。

2. サブシステムノードと操作内容ノードのマッチング

部分事例 1 については、システム名ノードでマッチングされていないので対象外となる。部分事例 2 では、システム名ノードでマッチングされているので、「受注売上管理」「発注管理」でマッチングがとられ、基本事例に重ねられる。部分事例 2 中の「在庫管理」についてはマッチングがとられないので、付属している操作内容ノードとともに基本事例に継ぎ足される。部分事例 3 では、「顧客管理」でマッチングがとられる。しかしながら、この「顧客管理」ノードにはシステム名ノードと接続されていないため、「販売管理システム」との依存関係が明確でない。そのため、結合はされない。部分事例 4 の「電子メール」ではマッチングがとられないので、結合されない。

3. データ名ノードのマッチング

上位ノードの影響を受けないデータ名ノードよりマッチングをとると、部分事例 1 の「商品情報」「売上データ」がマッチングされるので、部分事例 1 が基本事例にデータ名ノードを介して結合される。部分事例 5 では、「商品情報」がマッチングされるので基本事例に結合される。

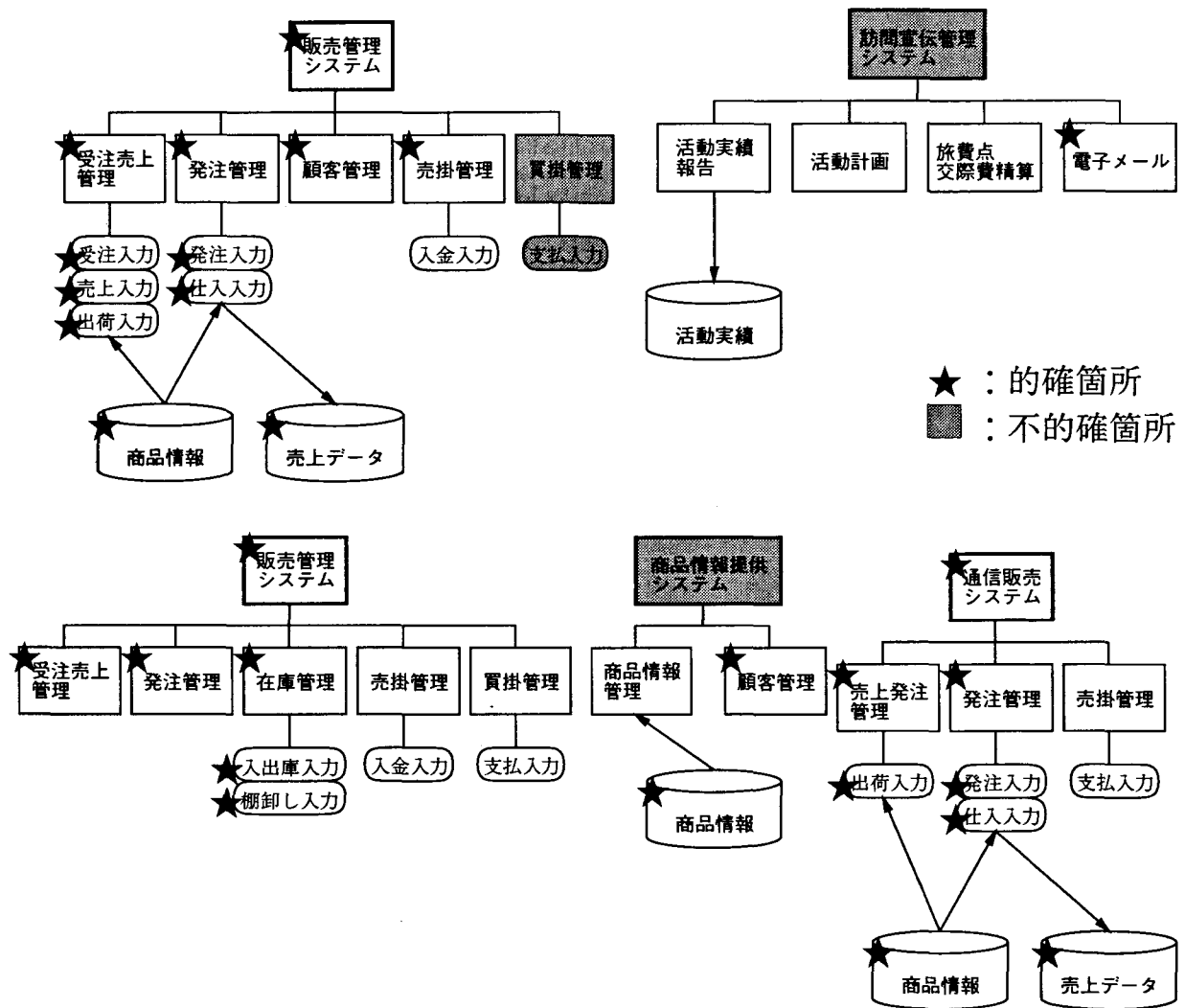


図 3.7: 的確・不的確の指定

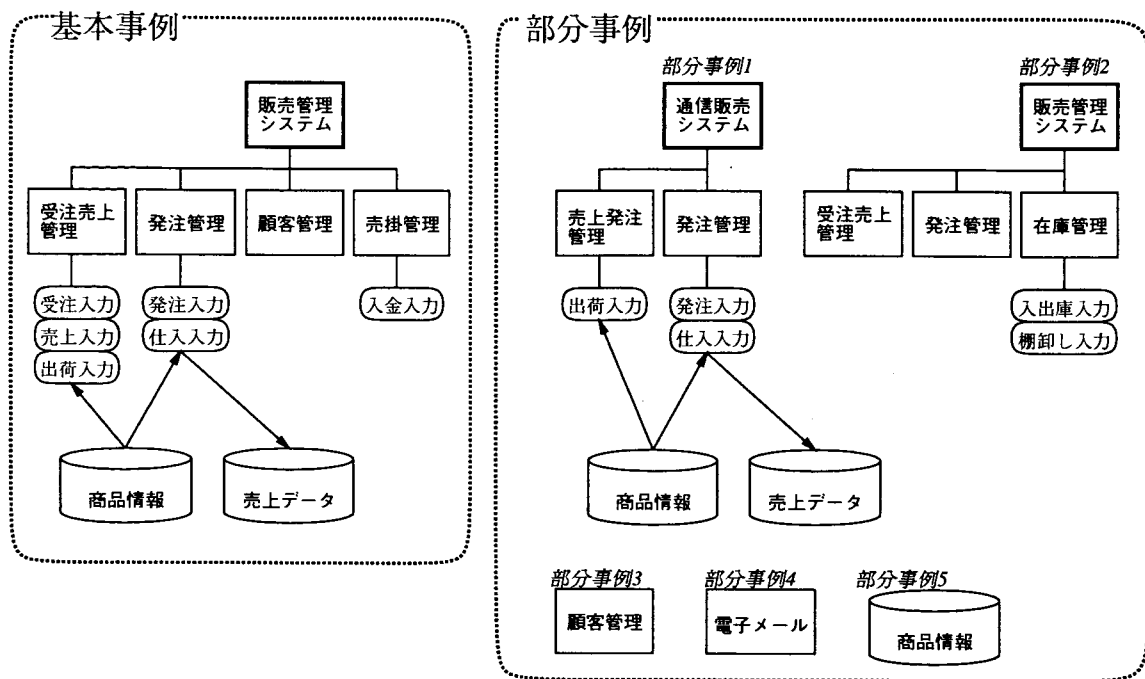


図 3.8: 基本事例と部分事例

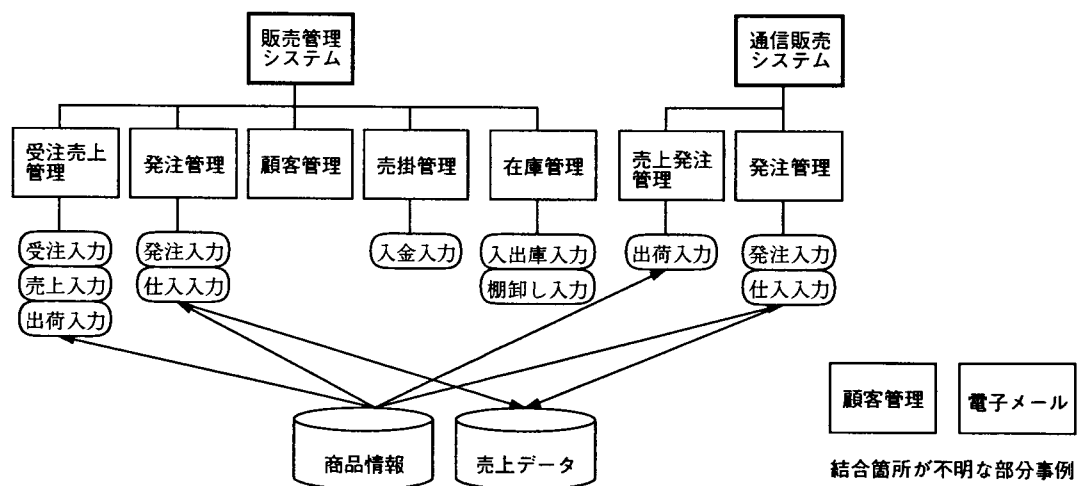


図 3.9: 結合結果

結合結果を図 3.9を示す。この結合では結合されなかった部分事例「顧客管理」「電子メール」が存在する。「顧客管理」に対しては、「販売管理システム」のサブシステムノードと重複、「通信販売システム」のサブシステムノードとして新たに加える、または独立した「顧客管理システム」として存在させるという結合候補がある。「電子メール」に対しては、「販売管理システム」または「通信販売システム」のサブシステムノードとして加

えるか、独立した「電子メールシステム」として存在させるという結合候補がある。従って、基本事例への結合候補が図 3.10 のように示される。これらを再びユーザに提示し、的確・不的確の入力を行ってもらう。

このようにして、事例は繰り返し結合される。最終結合結果として図 3.11 のツリーが得られる。

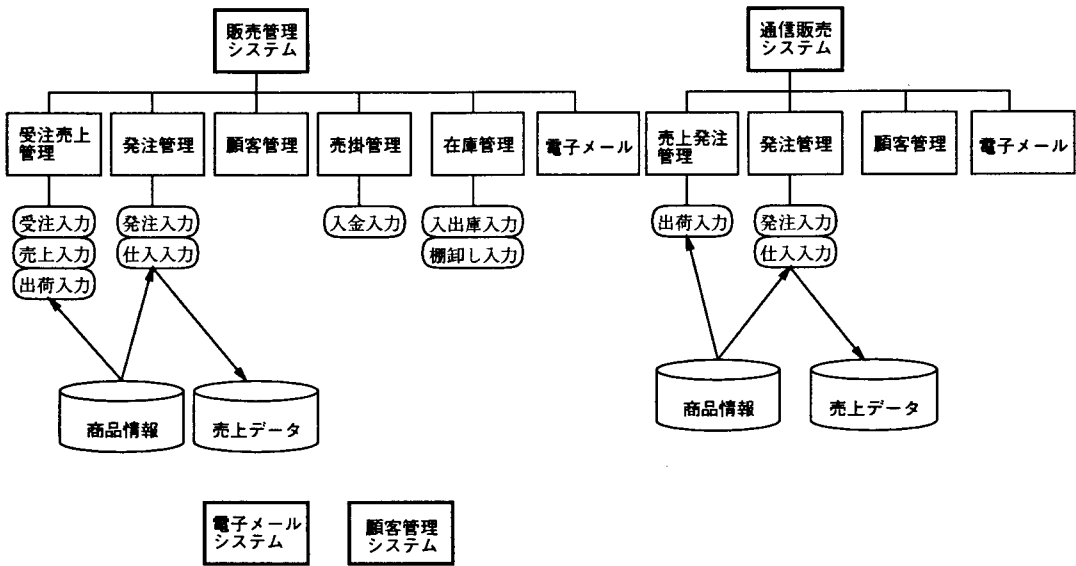


図 3.10: 候補事例

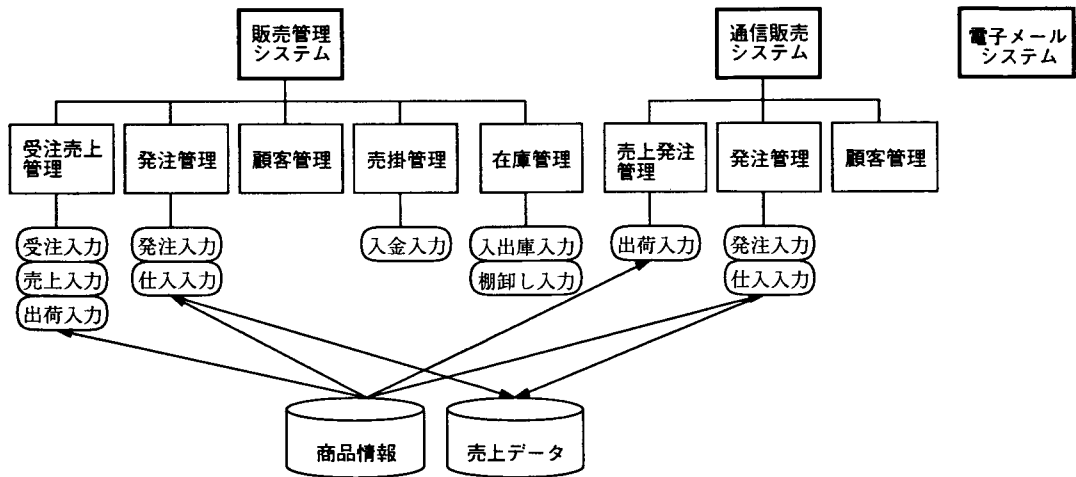


図 3.11: 最終結合結果

3.5 比較実験

複数事例の結合によって作成される業務ツリーがどの程度有効なものであるか検討するために、比較実験を行った。実験方法を以下に示す。

1. SE 経験のない被験者 8 人に対して、対象の業務の内容に関する A4 用紙 1 枚程度の資料を与えた。実験に用いた事例は、図 3.12 の 6 件である。

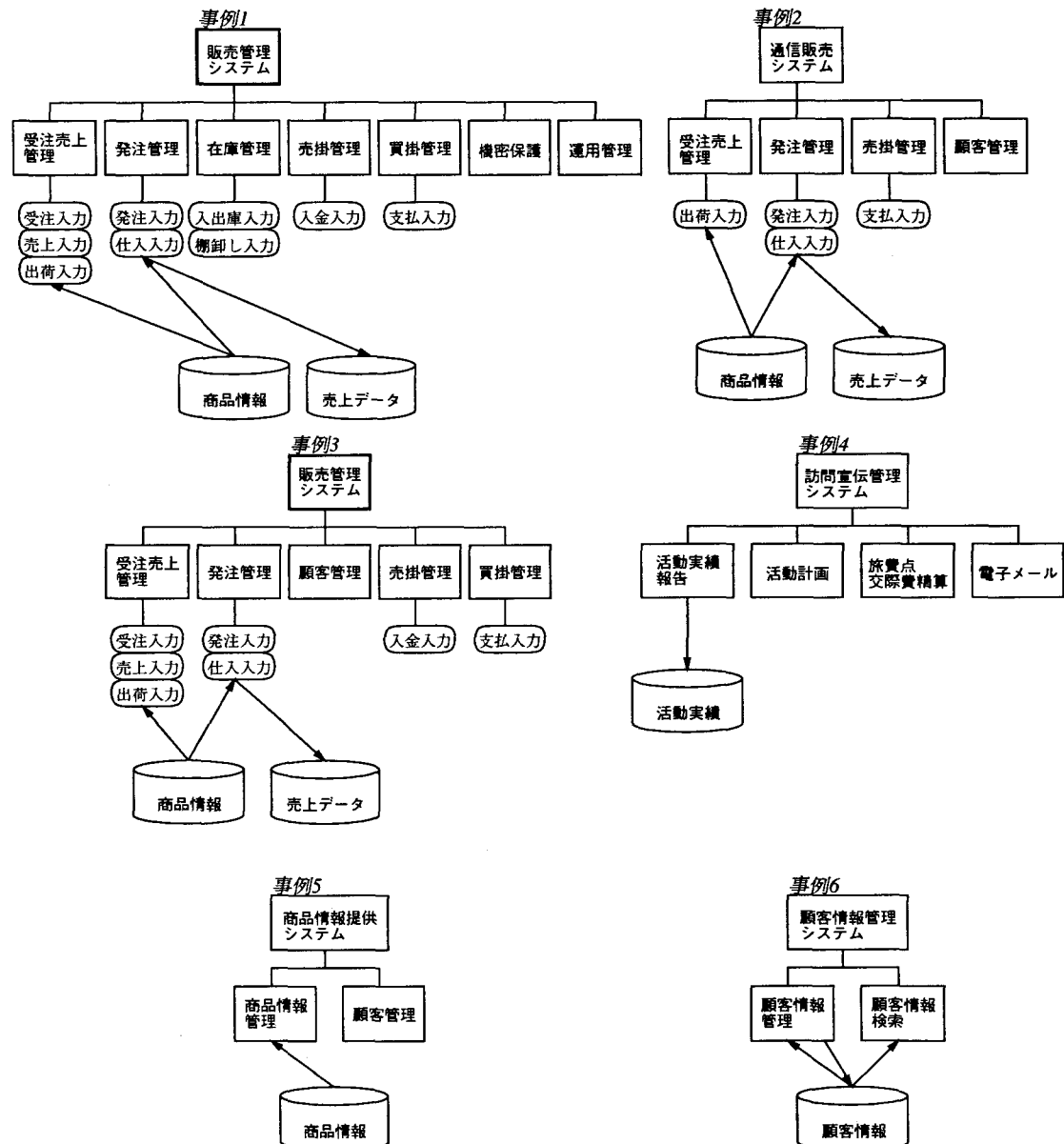


図 3.12: 比較実験に用いた事例

2. 資料を基に、4 人の被験者に複数事例に対する的確・不的確の指摘を行ってもらい、その結果を結合することにより業務ツリーを作成した。他の 4 人には類似事例 1 件

(図 3.12の事例 1) を参考にし業務ツリーを作成してもらった。

3. 被験者により作成された業務ツリーと、同様の資料をもとに熟練者によって作成された業務ツリー (図 3.13) とのノード比較を行った。

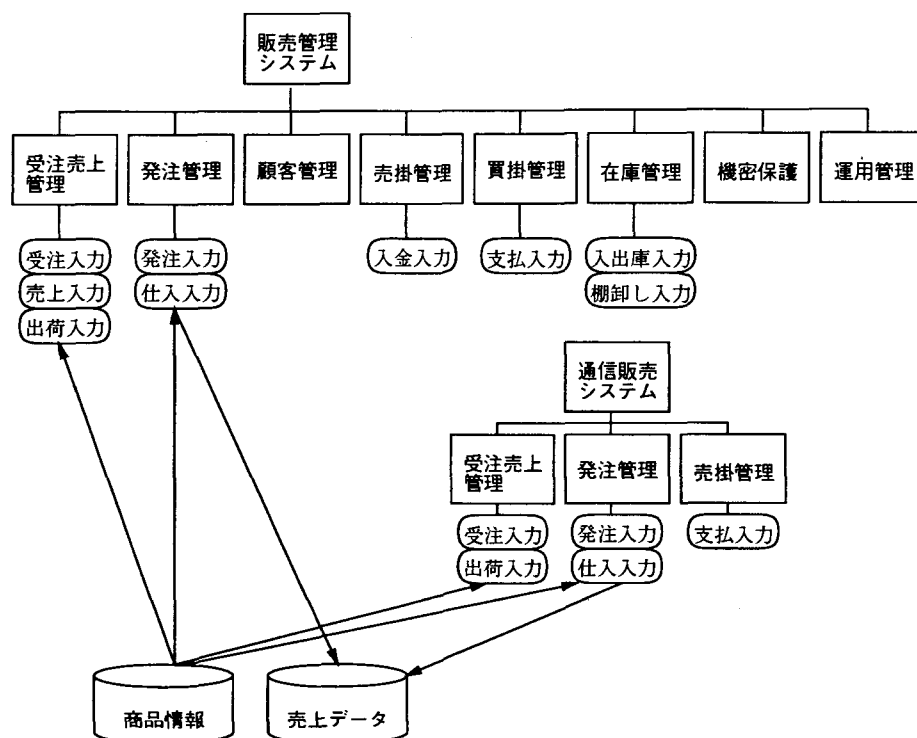


図 3.13: 熟練者が作成した業務ツリー

表 3.3: 比較実験結果 (平均ノード数)

	一致ノード	不足ノード	余分ノード
複数事例結合	26.25	2.75	4.50
1 事例参照	16.25	12.75	2.75

結果として表 3.3を得た。表中の「一致ノード」とは被験者のツリー中に存在しかつ熟練者のツリー中に存在するノードである。「不足ノード」とは熟練者のツリー中に存在するが被験者のツリーには存在しないノード、「余分ノード」とは熟練者のツリーには存在しないが被験者のツリーには存在するノードである。このように、複数事例結合によって作成した業務ツリーは、類似事例参照による業務ツリーに比べ、より熟練者に近いツリーが作成されており、業務ツリーの作成には、複数事例の結合が有効であることがわかる。

3.6 まとめ

本章では CIDAC 中の要求分析部における業務ツリー作成方式について述べた。過去に作成された業務ツリーを事例ベースとして用い、複数事例の結合により対象システムの業務ツリーを作成する。複数の事例を用い、業務ツリーに存在する階層的な因果関係を考慮しながら結合することで、目的とする業務ツリーの不足箇所が軽減され、初級 SE でもより熟練者に近い業務ツリーが作成される。ユーザは事例中の各ノードに対して的確であるか、不的確であるかという二者選択という容易な入力のみにより、対象システムに関する業務ツリーの作成が可能である。

第 4 章

クライアントサーバシステム構成設計向け事例修正方式

4.1 まえがき

本章では CIDAC 中の事例を用いた設計案作成 [81][82] について述べる。CIDAC では設計案作成時に起こる問題として、機能を満たすソフトウェアやハードウェアといった構成部品結合の整合性を考慮する。しかし、レスポンスタイムについては次フェーズである詳細設計の際に考慮されるべき内容であるので、本システムでは対象外とした。

設計案作成部では、ユーザの要求を入力とし、その要求を満たすような CSS の構成を出力する。CSS 構成の設計においては、要求機能と設計案上でその機能を実現する部分との対応が複雑であるため、設計案作成ルールを構成することが難しい。そこで、CSS 構成設計支援システムでは事例ベース推論 [60]～[69] を適用した設計案作成を行う。

事例修正により設計案を構成するには、CSS 構成案を、機能とそれを実現するために必要なソフトウェアやハードウェア (以下部品と呼ぶ) とを対応できるように表現しなければならない。そこで、本研究では、CSS 構成案を各部品の機能や関係を明記できるような階層的な木構造で表現する。修正に用いる部品は、機能実現のために必要となるソフトウェアなどの組合せを表現した木と、それらを使用するために必要となる他の部品の組合せを表現した木を用いる。

また、CSS を構成する際にシステム構成上の実現可能性が重要となる。実際に稼働する組合せかどうかという部品間の相性を考慮するために、部品の相性を記述した相性知識を用いて、設計した CSS が実際に動作するかどうかをチェックする方法を提案する。

以下、第 4.2 節では事例修正方式について、第 4.3 節 CSS 構成の表現形式について説明する。次に、第 4.4 節に事例修正の例を示した後、提案方式に基いたプロトタイプシステムの画面例を示す。

4.2 事例修正による CSS 構成設計の概要

4.2.1 設計案作成部の概略機能

設計案作成部では、要求分析部において作成された要求仕様中の業務ツリーおよび要求特徴データ中のシステムのパターンに関する情報を用いて設計案事例の検索を行う。CSS 構成設計案は、検索された事例に対して、要求仕様をもとに修正を加えることによって作成される。まず、要求特徴データ中の機能に着目した内容を用いて初期修正が行われた後、詳細要求を用いて再修正が行われる。修正過程は設計案作成過程情報として設計案分析説明部に渡される。以下、初期修正について述べる。

初期修正に用いられる要求仕様は、要求特徴データの一部であり、ユーザが CSS に求めている機能が以下のように、一つのサーバおよびクライアント毎に示される。まず、最初にクライアントであるかサーバであるかを記述し、次に全体システムにおける役割として目的を記述し、その後ろに必要となる機能を列挙する。

サーバ 目的 (機能 1, ..., 機能 m)

⋮

クライアント 目的 (機能 1, ..., 機能 n)

⋮

検索される事例は過去の CSS 構成設計案であり、CSS を構成しているハードウェアやソフトウェア (部品) の集合と、それぞれのサーバもしくはクライアントの全体システムにおける役割が記述される。この CSS 構成では各部品において、部品の種類に関する制約、部品の製品選定における制約の 2 種類の制約が存在する。前者は、OS を持たなければソフトウェアは動かないし、OS を動かすためにはマシンが必要となるといった、部品の種類レベルでの制約であり、構造的制約と呼ぶ。後者は、あるソフトウェアを使用するためには、OS の選定は限定されるといった個々の製品レベルの使用条件に関する制約であり、性質的制約と呼ぶ。

要求特徴データ中の機能要求情報を全て満たすようなハードウェアやソフトウェア (部品) を設計案中に含むように検索事例を修正し、図 4.1 に示すような構成案を作成する。また、作成過程情報として決定条件に関する情報を構成案中の部品に付属させ、設計案分析説明部に出力する。

4.2.2 事例修正方式

一般に事例ベース推論における事例修正過程は、事例の修正箇所の特定、それらに対する修正操作、修正結果の確認というステップからなる。本節での事例修正箇所の特定とは、事例である設計案中の不必要な部分と、事例では不足する機能を特定することが対応する。また修正操作とは、不必要な部分の削除と不足した機能の補充であり、修正結果の

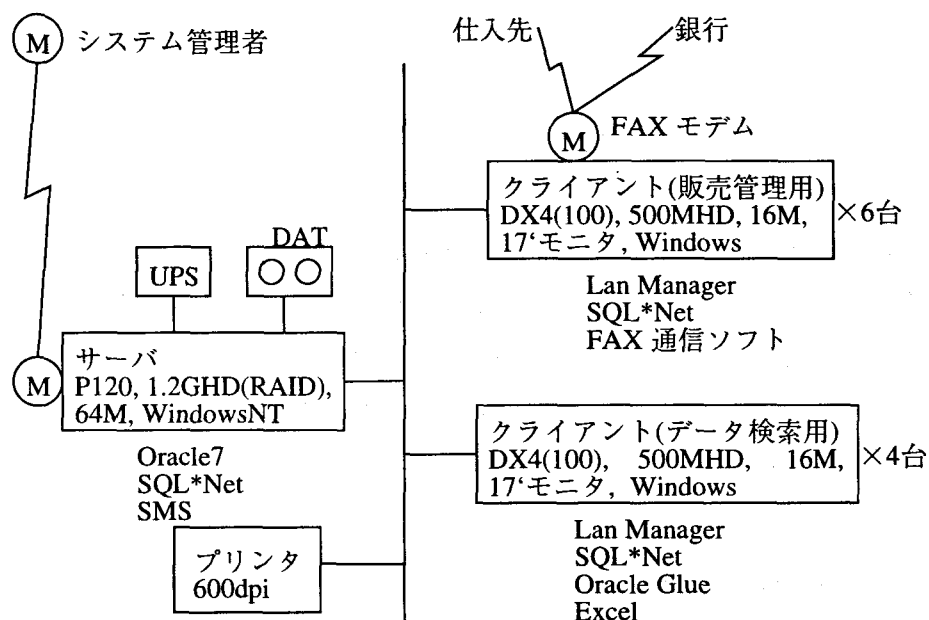


図 4.1: CSS 構成設計案の例

確認とは、修正結果の設計案が稼働可能であるかという確認である。事例修正を図 4.2 に示す手順によって行う。

「機能比較」は、事例中で要求を満たしている部分、要求中で満たされていない部分をそれぞれ明確にするために行う。要求仕様中の各クライアントやサーバに必要な機能と、事例の設計案上に実現されている機能との対応をとる。

次に、「不要部品削除」を行う。「機能比較」において事例中の不要とみなされる機能を果たす部品、又は個々のクライアントやサーバ単位での部品集合を削除して、修正事例とする。

「必要部品選択」では必要となる機能を果たす部品を部品データベースより選択する。

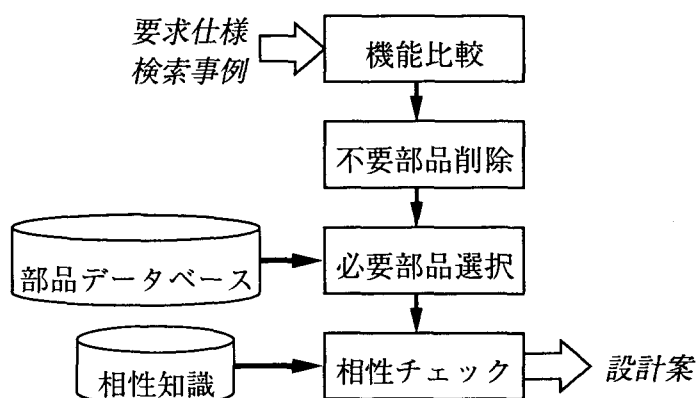


図 4.2: 事例修正方式の概要

部品データは、不足する機能を補うために用いられ、一つの機能に対して必要となる部品の組合せが記述する。この組合せには性質的な制約が存在する。次に、相性の良い部品の組合せを記述した相性知識を用いて、新たに加える部品に対して、修正事例を構成している部品との「相性チェック」を行う。ここでいう相性とは、部品の組合せによって設計案が稼働するかどうかである。

それぞれの機能毎に新たに加える部品候補の組み合わせを網羅する。次に、それらの組合せの相性が良いものであるかどうかを確認し、相性の良い組合せの部品を構造的制約に基づいて修正事例に加えることにより設計案を完成させ、図 4.1 のように部品をサーバもしくはクライアント毎にまとめた形式で出力する。相性知識の条件を満たした部品の組合せ数だけ設計案は出力される。

4.2.3 事例修正方式における問題点

上述の事例修正方式では、事例、部品データ、相性知識の 3 種類のデータを用いる。計算機上で本方式を実現させるためには、これら 3 種類のデータを計算機処理できる形式で表現しなければならない。以下にそれぞれのデータ表現における問題点を述べる。

CSS 構成事例データを計算機で処理するには、「各構成部品がどのような機能を果たしているのか」を明確にしなければならない。そこで、事例表現形式では一つ一つの機能がどのような部品によって満たされているかを明確にする。次に、構造的な制約関係が明確であることが要求される。しかしながら、通常よく提示される図 4.1 に示した様な CSS 構成図を事例表現として用いると、各部品の役割は不明であり、部品間の関連も不明であるため、修正箇所を特定することはできない。

また、部品データの記述においては、機能を満たすための部品を単なるグループとして扱った場合、性質的な制約関係を表現することができない。

相性知識の表現では、一つ一つの部品間の相性が必要である訳ではない。例えば稼働可能な部品の組合せとして (部品 A, 部品 B)(部品 B, 部品 C) の 2 つがあった場合、必ずしも (部品 A, 部品 B, 部品 C) という組合せが稼働可能であるとは限らない。そのため、相性知識では全体的な相性関係を表現しなければならない。

4.3 CSS 構成の表現形式

4.3.1 CSS 構成データ

事例ベースには CSS 構成データが格納される。1 事例中に記述される項目は、CSS 構成の要素となる対象業務、システム化の目的、果たす役割、用いたソフトウェア、OS、機種、周辺機器であり、個々の事例間には関連を持たない。

要求仕様に沿って置換操作するため、大きな枠組として、後述する個々のクライアントもしくはサーバの部品集合 (部品セット) の表現方法と統一させるために、事例となる CSS 構成を図 4.3 のような木構造で表現する。

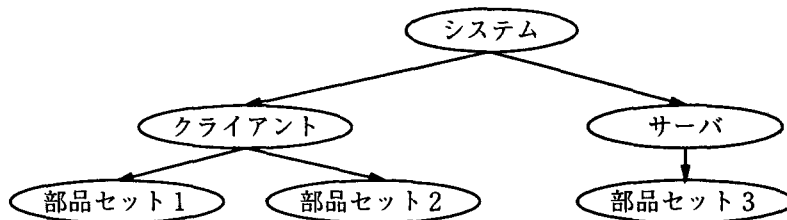


図 4.3: CSS 構成の表現

システムは全体としてクライアントとサーバに分別し、クライアント・サーバはそれぞれ数種類の部品セットを持つ。部品セットとは、一つのラベル (使用目的名) を持ったクライアントまたはサーバを実現するためのソフトウェアおよびハードウェア (部品) の集合である。具体的には図 4.1 に示した構成例中に記述されている四角で囲まれたハードウェアとその下に記述されたソフトウェアの集まりである。部品セットノードには、その部品セットが果たす役割や目的を記述される。

部品セットを構成する部品には、「通信」「ソフトウェア」「周辺機器」「OS」「マシン」といった種類がある。部品間の構造的制約を表現するために、部品の種類に関する情報を付加させると共に、種類間の制約関係を表現する。このような種類間の制約関係を表現するためには、単なる部品の集合として扱うことはできない。また、制約関係を踏まえた各部品をノードとした柔軟なネットワークとして表現した場合、修正箇所の決定が複雑化する。そこで、制約条件 (接続関係) を記述し、方向性を持たせることができる表現として木構造を採用する。木構造として表現するためには、制約関係に一定の規則性が必要となる。部品セットにおける各部品の制約関係を分析した結果、部品セット内での接続関係は図 4.4 のように示される。ただし、記号の定義は以下の通りである。

a	: 部品セットノード
B	: 機能ノード (実現される機能を記述) の集合
c_{type}	: ソフトウェア, 通信, 周辺機器のいずれかの種類を示すノード
c_{os}	: 種類が OS であることを示すノード
$c_{machine}$	: 種類がマシンであることを示すノード
c_{name}	: 部品の名前を示す部品名ノード
C_a	: $(c_{type} : c_{name})$ の組の集合
C_b	: $(c_{os} : c_{name})$ の組の集合
C_c	: $(c_{machine} : c_{name})$ の組の集合

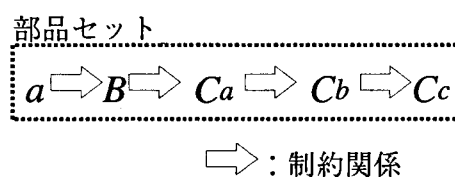


図 4.4: 部品セット内の接続関係

まず 1 サーバ (クライアント) の全体システム中での役割 (ノード a) があり、それらをもとに果たすべき機能 (ノード B) が決定される。機能を果たすために必要となる部品 (ノード C_a) としてソフトウェアや通信や周辺機器があり、それらを稼働させるためには OS (ノード C_b) が必要となる。また、OS の稼働させるためにはマシン (ノード C_c) が必要となることを表現している。

具体的には、各部品セットを図 4.5 のような階層構造で表現する。

以上のように CSS 構成を表現することで、各機能レベルでの比較を実現すると同時に各部品間の構造的制約の記述を可能にする。

4.3.2 部品データ

部品データは以下の 2 種類を用意する。

- ある機能に対して必要となるソフトウェア、通信、周辺機器を決定するデータ
- 前者の決定結果をもとに、OS やマシンを決定するデータ

前者では 3 種類の部品の組合せを、図 4.6 のような木構造で表現する。

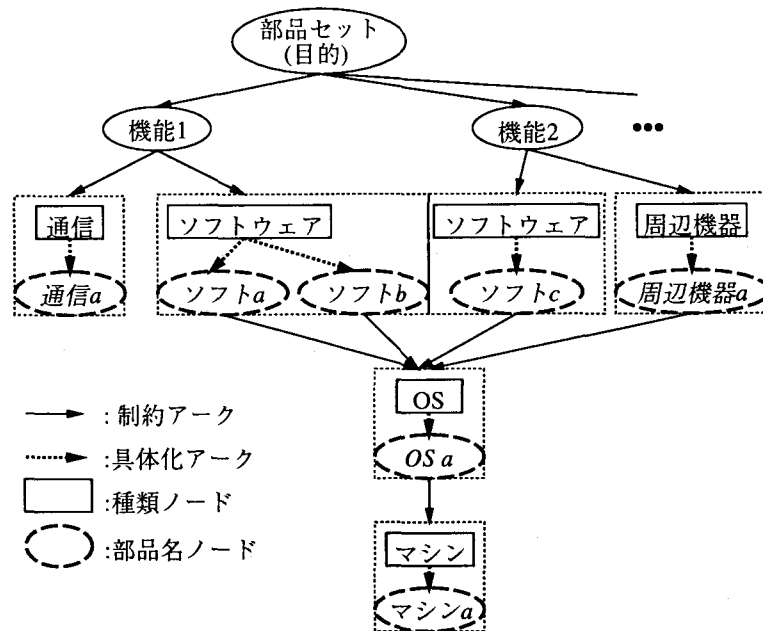


図 4.5: 部品セットの表現

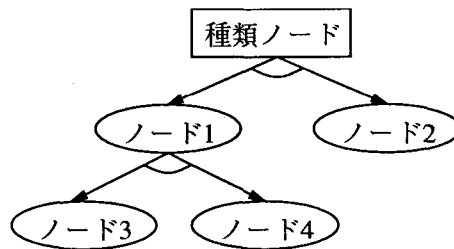


図 4.6: 各要素の組み合わせ

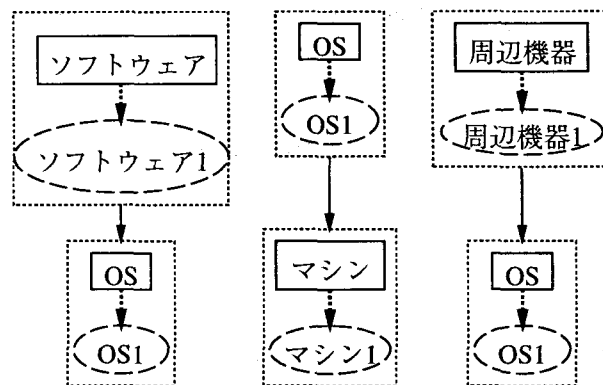


図 4.7: 部品間の性質的な制約

親ノードである種類ノードは、ソフトウェア、通信、周辺機器のいずれかが記述される。他のノードには部品名が記述される。木の横のつながりが OR、縦のつながりが AND となっており、図 4.6 の木で種類ノードがソフトウェアであるとき機能を実現するために必要となるソフトウェアは、(ノード 1, ノード 3)(ノード 1, ノード 4)(ノード 2) のいずれかの組み合わせである。

次にマシンや OS を決定するために、後者のデータでは、ソフトウェア等からの制約条件を満たす組合せを図 4.7 のように表現する。この表現では、性質的な制約を表現すると同時に、構造的な制約を考慮に入れるため、CSS 構成の表現の一部と同様の形式である。

4.3.3 相性知識

CSS 構成部品の全体的な整合性確認するための知識であり、部品選定の際の条件となる。相性の良い稼働可能な組み合わせを 1 つの集合 (グループ) として扱い、グループに含まれている組合せとなるように部品を選ぶ。集合を記述する場合、集合の要素をなす部品の列挙することが考えられるが、今回の場合は各集合の要素が重複していることが多いため、部品の集まりとして記述するのではなく、各部品がどのグループに所属可能であることを記述する。具体的には、図 4.8 のように表現する。図中の各部品を示したノードがグループマネージャによって 1 つのグループを形成している。グループマネージャとは、各相性の良い部品間を接続し、1 つのグループマネージャから出されるアークによってつながれている部品間は相性が良いことを示している。このグループマネージャによって繋がれた部品のグループのサブセットであれば、選定された部品は稼働可能である。例えば図中のグループマネージャ 1 ではマシン A、マシン B、ソフト B、ソフト C の組合せが相性が良いことを表わしており、これらの中の組合せであれば稼働可能である。

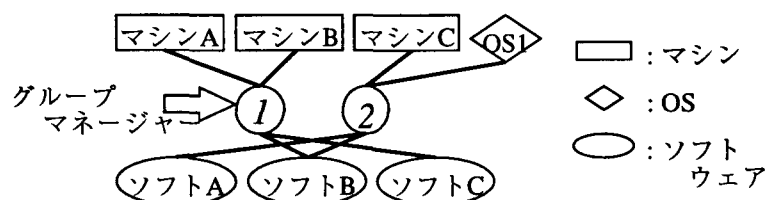


図 4.8: 相性知識

4.4 プロトタイプシステム

前述の手法をもとにシステムのプロトタイプとして事例修正による CSS 構成設計システムを開発した。

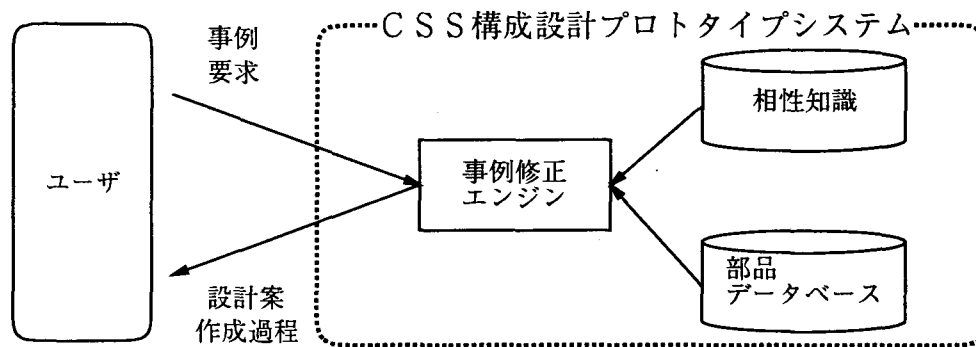


図 4.9: プロトタイプの構成

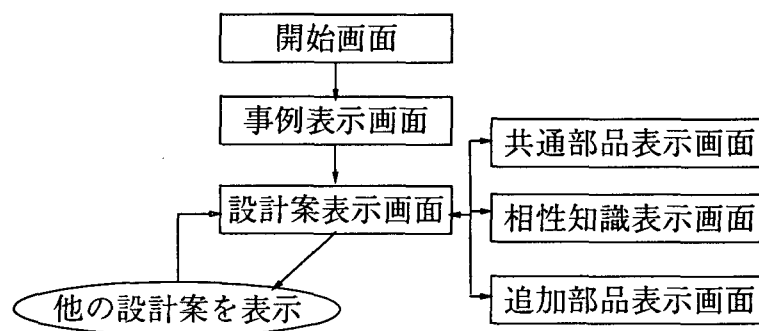


図 4.10: 画面の流れ

このシステムは図 4.9 のような構成をとり、要求と事例を与えることで設計案を作成する。このシステムではまず事例の表示を行ない、次に修正結果である設計案を表示する。さらに、この設計案の修正過程を説明するために事例と要求との共通部品、使用した相性知識、新たに追加された部品の表示を可能とした。図 4.10 に画面の流れを示す。

出力画面の例として、修正結果として得られた設計案表示画面を図 4.11 に示す。

4.5 事例修正例

以下に事例修正結果の例を示す。例で用いる要求仕様は

クライアント 販売管理用 (販売管理,DTP)

クライアント 統計データ (統計データ表示)

サーバ データベースサーバ ()

である。

修正に用いた最類似事例は図 4.12 である。この際事例の修正に用いた部品データは、DTP、統計データ表示機能のための組合せツリー 2 個であり、性質的な制約データは OS

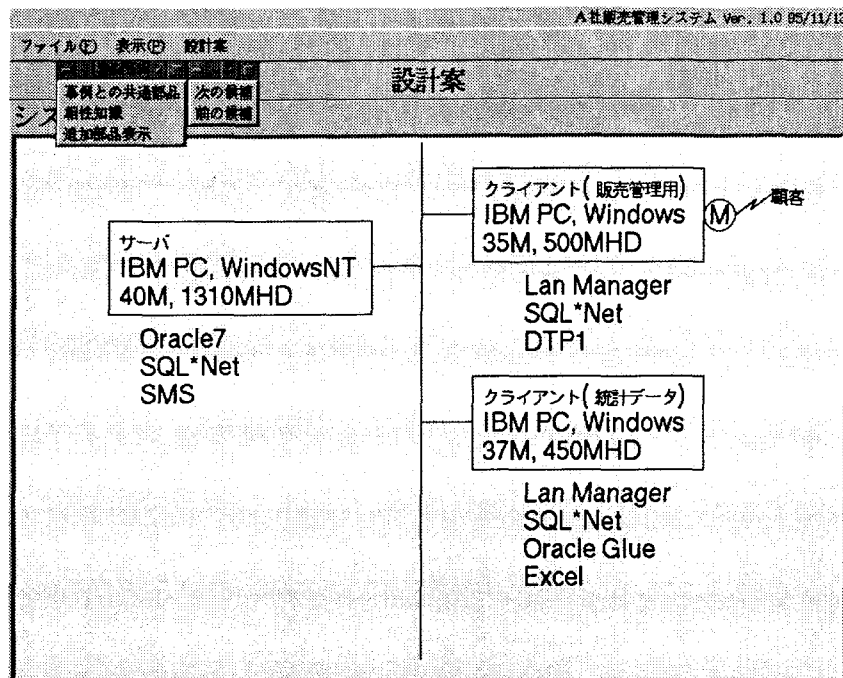


図 4.11: 設計案表示画面

が Windows であるという制約を含んだ 4 種類 6 個を使用した。相性知識としては 1 個のグループマネージャーが使用された。

要求仕様の機能に対応する事例中の販売管理用クライアントとサーバがそのまま使用され、含まれていないデータ検索用クライアントは削除される。補充する機能として、DTP と統計データ表示が必要となる。これらの機能を満たすため部品データベースよりソフトウェアの組合せが (Oracle Glue, Excel, DTP1), (Oracle Glue, Excel, DTP2), (Oracle Glue, Lotus, DTP1), (Oracle Glue, Lotus, DTP2) の 4 通りが選出された。これら 4 通りのソフトウェアの組合せと修正中の事例に含まれている OS およびマシンとの全体的な相性を考えるため、それぞれの場合について修正中の事例にある部品 (IBM PC, Windows, SQL*Net, Oracle7, SMS, Lan Manager) を含む図 4.13 に示す相性知識を用いて確認を行った。その結果、図 4.13 中には先述の修正中の事例の部品以外に、(WindowsNT, Oracle Glue, Excel, DTP1) が記述されており、これらの 4 つの部品を修正中の事例に同時に加えても問題がないことがわかる。そのため、(Oracle Glue, Excel, DTP1) の組合せの採用が決定された。そこで、使用目的が統計データであるクライアントの統計データ表示機能に対するソフトウェアとして Oracle Glue, Excel が付加され、使用目的が販売管理であるクライアントの DTP 機能に対するソフトウェアとして DTP1 が付加され、図 4.14 に示す修正結果が得られ、出力結果として各クライアントもしくはサーバの部品セット毎に部品を列挙した形式である図 4.11 に示した構成案として得られる。

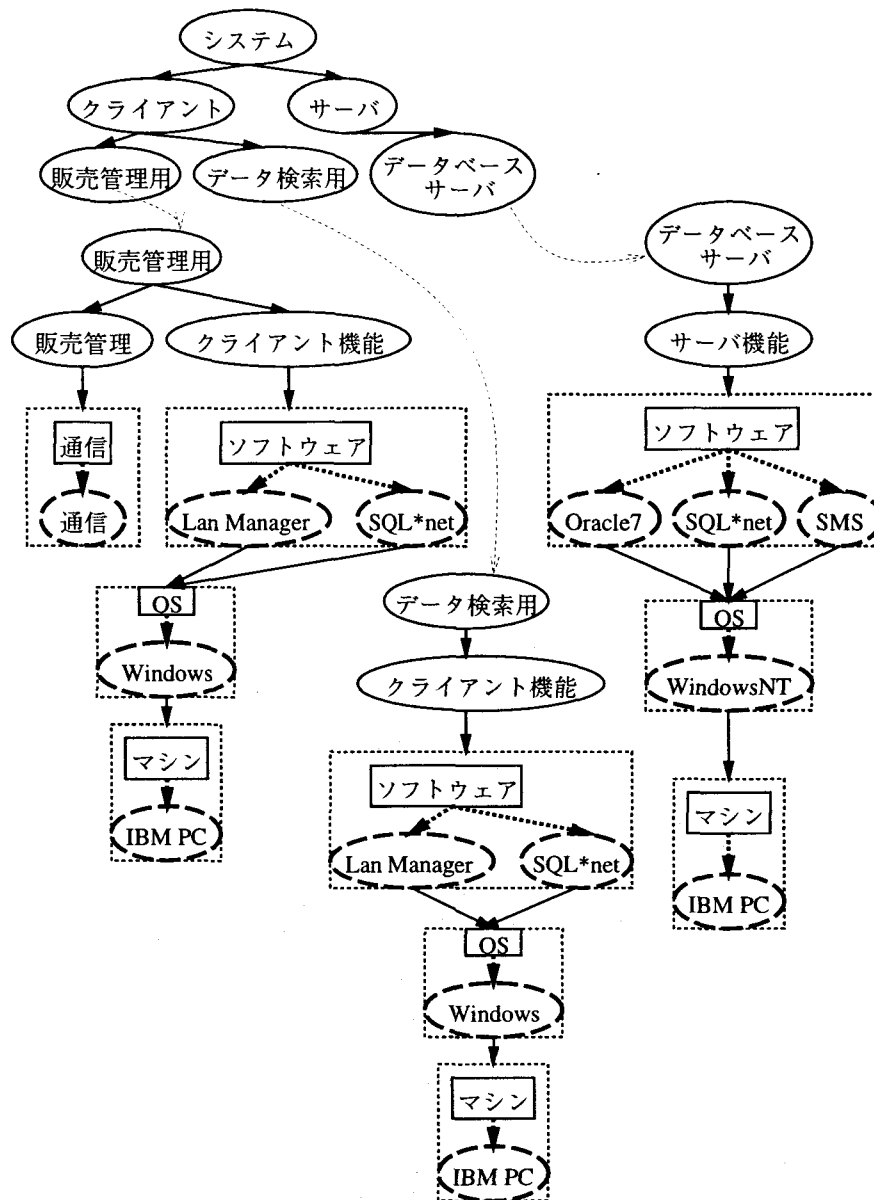


図 4.12: 事例

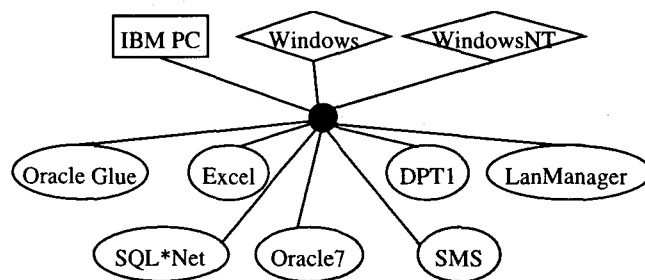


図 4.13: 用いた相性知識

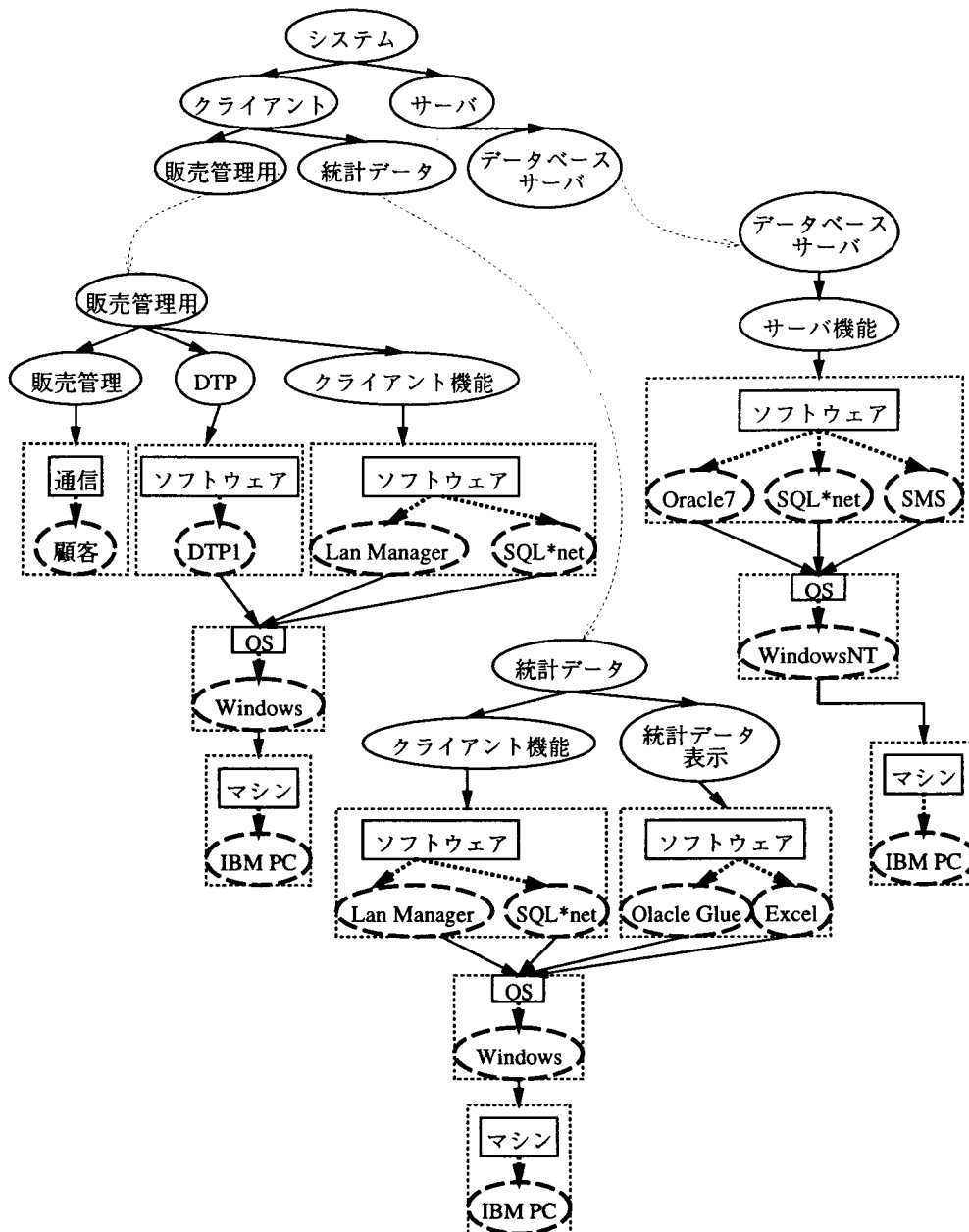


図 4.14: 事例修正結果

この出力結果に対して、専門家(熟練SE)に意見を求めたところ、「要求を満たした構成が導出できている。事例や部品データベース、相性知識を充実させることで、システム構成上の実現可能性の保証による効果も期待できる」との評を得ている。

4.6 まとめ

本章では、CIDAC の設計案作成プロセスにおける事例修正方式を考案した。本方式では、要求仕様との機能比較を可能とするよう CSS 構成事例を階層的な木構造で表現する

ことを提案した。また、修正に用いる部品データを、機能とソフトウェア、ソフトウェアと OS などをそれぞれの対応に即した木構造で表現することを提案した。さらに、ソフトウェアやハードウェアの相性を記述した相性知識を用いることにより、CSS を構成する部品間の相性のチェックを行うことで、実際に動作可能な CSS 構成案を設計することができる。

本システムの利用によって期待できる第 1 の効果は、設計案作成時間の短縮である。初級 SE が設計案を作成するのに、調査等を行うため 1 ヶ月程度必要としている。これを本システム利用することで、調査や資料作成の手間が削減されるので熟練 SE と同様の 1 週間程度でシステム提案が可能になると予測される。第 2 の期待効果は、初級 SE が提案を行えるまでの教育期間の短縮である。本システムを利用することで、教育と同時に実務が行えるので、初級 SE が提案を行えるまでにかかる年数が半減できると想像される。

なお、本システムでは事例ベース中に適当な事例がない場合、部品データにより事例をすべて置き換えることにより次善の回答を作成する。しかし、部品データが存在しない場合には、完全な解が求めることができない。今後、データ不足の際の緩和解を求める機能が必要となる。また、作成した複数の CSS 構成案の評価を行うため、コストなどの要素を用いた比較検討機能が必要となると考えられる。

第 5 章

設計案説明文作成方式

5.1 まえがき

本章では CIDAC における設計案分析説明部 [77] について述べる。

設計案分析説明部を構成する設計案分析およびその説明作成作業は、ユーザである SE に対する出力結果を作成する部分であり、全体システムを考える上で重要な部分となる。実際に作成した設計案を顧客に提示する際には、必ず各部の設計根拠を説明しなければならず、設計案の説明作成機能なしでは初級 SE が熟練した SE に近い役割を果たすことはできない。また、説明文は全体システムの出力結果となるものであり、説明文がどのような形式ならば計算機上で作成することが可能であるかという検討をしなければ、設計案の作成や要求分析の実現方法は考えられない。

設計案分析説明は、業務支援と共に学習を目的としている。そこで、SE の理解を容易にするため設計案説明を簡潔な自然言語で提示する。システムが行った思考過程や出力結果を自然言語による説明機能はいろいろなシステムで行われている [39][40][42][43][72]。これらの説明では、使用した知識やルールを提示したり、ルール使用の履歴を検索することで説明文が作成されている。しかしながら、本システムでは、一定のルールを用いていない。さらに、説明すべき内容がシステム動作の履歴のみでは不十分であるため、従来の考え方では説明文を作成できない。

本説明機能において説明すべき内容を明確にするために、実際の設計提案書の記述内容を分析した結果、計算機が仮定した事項の確認を促すもの、設計案が要求仕様をどのように満たしているか説明するものという 2 種類に分類される。さらに、初級 SE に対する学習のため、どのように設計を行ったかという設計案作成過程を説明が必要となる。そこで、本説明機能では上述の 3 種類の説明を提供する。要求分析や設計案の作成過程を、構造的なモデルとして表現し、そのモデル中において有効である説明箇所を特定することで、SE にとって必要な部分のみを簡潔に自然言語形式の文章で提示する。

以下、第 5.2 節では本機能の構成と要求分析および設計過程の表現モデルを提案する。

第 5.3 節において設計案分析説明文の種類、説明すべき事柄の選択基準について、第 5.4 節では説明文作成の実行例について述べる。

5.2 説明文作成におけるシステム構成

設計案分析説明作成部では、要求分析過程情報と設計案作成過程情報をもとに、SE に対して設計案の作成過程の説明文を自然言語形式で作成する。

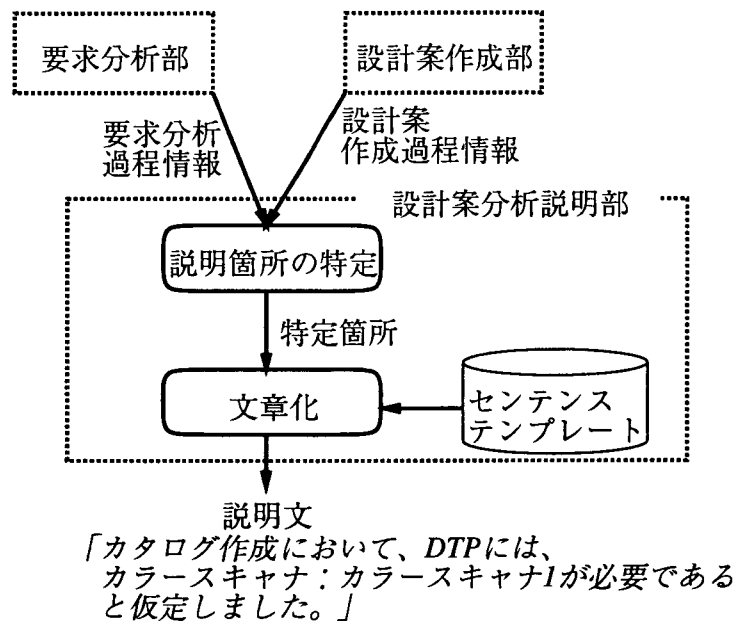


図 5.1: 設計案分析説明部システム構成

説明文の作成は、説明箇所の特定、文章化の2つのフェーズに分かれる。設計案分析説明部のシステム構成を図 5.1に示す。与えられたモデルにおいて、後で説明する有意性を基に説明箇所の特定を行う。特定された箇所を文章の枠組であるセンテンステンプレートの空欄に埋め込んでいくことで、自然言語形式の説明文を生成し出力する。このセンテンステンプレートは、説明内容によって種類を設けることで、説明内容に適した表現を実現する。

以下に入力となる要求分析過程情報、設計案作成過程情報について述べる。

5.2.1 要求分析過程情報

要求分析過程情報とは図 5.2に示すような、複雑な要求分析の過程を明確な形式で捉えるための木構造のモデルである。モデル中の各ノードには、CSS 構成に対する要求が記述される。アークは具体化の継りを表しており、アークの始点のノードに記述された要求

の具体化の結果がアークの終点のノードに記述される。このアークには、上述の具体化の種類によってそれぞれ「仮定」「要求」「必須」のいずれかの意味が付加される。アーク x の意味が「仮定」である場合、 $M(x)$ = “仮定” と表現する。このモデルの最上部のノードは、最初に入力された最も抽象的な要求が示されており、最下部のノードには、設計案に直ちに反映させることができる具体的な要求が記述される。モデルの最上部ノード (初期入力要求ノード) から最下部ノード (具体機能ノード) までをたどることで、要求の具体化の過程を知ることができる。

図 5.2 の例では「販売管理」という漠然とした要求から、「受発注」、「オンラインによる発注」、「通信」機能とたどれる具体化の過程が示されている。

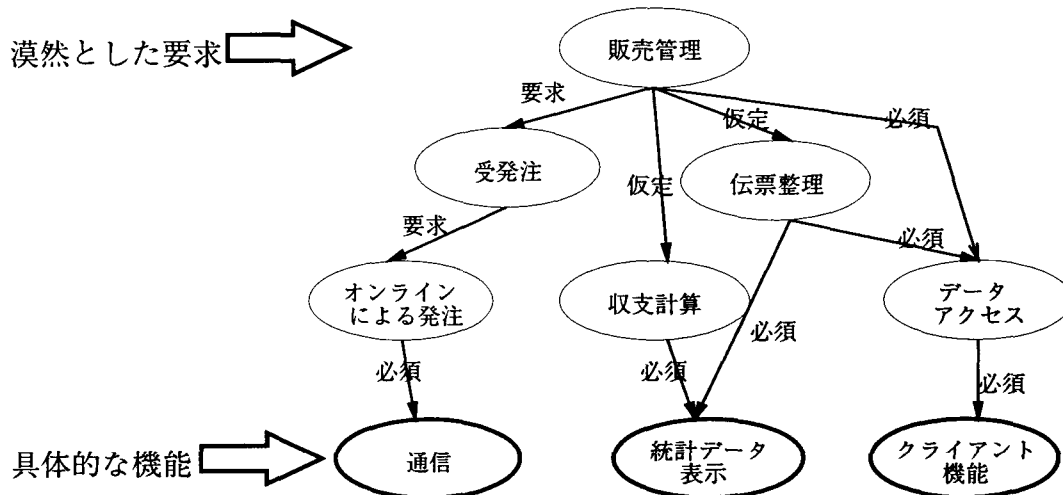


図 5.2: 要求分析過程情報の例

5.2.2 設計案作成過程情報

設計案作成過程情報とは、図 5.3 に示すような設計案作成部における CSS 構成の記述方式において部品決定の過程情報を含んだ構造モデルである。モデル全体で CSS 構成における 1 つのクライアントもしくはサーバの実現方法を表している。モデルの最上部のノード (目的ノード) には、全体の CSS 構成中でこのクライアント (またはサーバ) が果たす役割や目的を記述している。目的ノードから接続されているノード (機能ノード) に、要求分析過程情報中の末端に示された具体的に必要となる機能が記述される。つまり、要求分析過程情報中の最上部と末端部のノードが設計案作成過程情報中の上部に示されている。機能ノードより、実際に機能を実現させるための構成部品が接続されている。構成部品は「属性：属性値」という形式で表現される。ノード間の関係を示すアークには、構成部品の属性同士を接続するアーク (属性間接続アーク) と、属性から属性値を指定するアーク (属性値接続アーク) の 2 種類がある。また、それぞれの決定原因を表現するために、「仮

定」「必須」「要求」「コスト」「相性」「トラフィック」の6つの意味のいずれかがアークに付加される。要求分析過程情報中のアークと同様に、アーク x の意味が「仮定」である場合、 $M(x)$ = “仮定” と表現する。

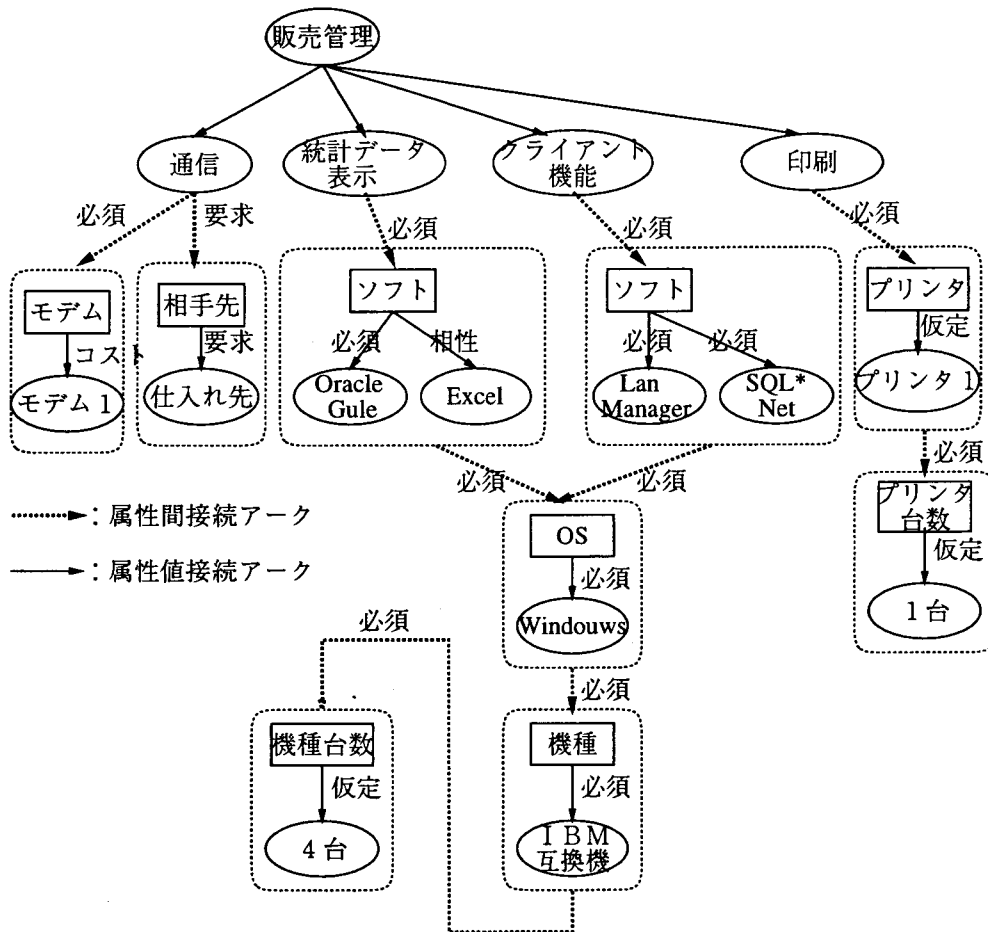


図 5.3: 設計案作成過程情報の例

図 5.3では、クライアントの1つである「販売管理」用のクライアントに関する設計案を示している。「販売管理」には「通信」「統計データ表示」「クライアント機能」「印刷」が必要であることが示されている。そして、それぞれの機能を満足させるための設計部品が以下に接続されている。

5.3 設計案分析説明の作成

5.3.1 設計案説明文の種類

作成すべき説明文はどのような種類があるか、実際の事例にある説明を参考に考察する。

通常、SE は作成した設計案がどのように顧客からの要求を満たしているかといった説明をし、顧客の理解を得なければシステム開発に入れない。そこで、作成する説明の内容として、作成した設計案がどのように要求を満たしているかという説明は必要不可欠となる。

また、設計案は理想的な場合、要求仕様を全て満たしているはずである。しかし、要求を全て満たした設計案を作成することが困難である場合が多い。また、CSS の設計に対しては漠然とした要求も多く、SE は設計の際にそれらがある程度仮定する必要がある。そのため、条件緩和や仮定がどこであり、どのような要素からそのような仮定を行ったかといった説明が必要となる。

さらに、SE の設計技術学習のため、作成した設計案のそれぞれの部品がどのような役割があるか、どのようにして決定されたものであるかといった説明が必要となる。

以上のように説明文は、入力した要求がどのように設計案に生かされているかという説明である「設計案・要求対応説明」、計算機がおいた仮定についての確認を促す「確認必要項目」、SE への教材として利用する設計案の各部の役割や設計根拠を説明した「設計根拠説明」という 3 種類に分類される。

それぞれの説明において、要求分析過程情報を対象としているのか、設計案モデルを対象としているのかによって伝えるべき情報が違ってくる。そこで、説明対象モデルによって表現を変え、適切な表現にするため、要求分析過程情報と設計案作成過程情報によって説明文を分類する。ただし、設計案根拠説明は設計案の作成過程の説明であるので、対象モデルは設計案作成過程情報のみである。さらに、設計案作成過程情報中には属性間接続アークと属性値接続アークの 2 種類がある。これらの違いに対応した表現をするためにもアークの種類によって説明文を分類する。

以上のように、作成する説明文の種類は図 5.4 のように分けられ、各種類によって伝えるべき内容が異なっているので、各説明の種類によって表現を変化させる必要がある。反面、情報を完全に伝えるために大量の文章を表示すると、ユーザである SE にとって読みづらく、冗長な情報を与え、実際に必要となる情報を逃す恐れがある。そこで、最小限伝えるべき情報のみを表示するようにする。

5.3.2 説明内容の有意性

必要最小限の説明文を作成するために、説明箇所の特定制が必要である。そのための基準として、説明内容の有意性を定義する。この有意性とは、作成する説明文に情報量があるか、つまり、内容が重複しているものが無いこと、矛盾した記述が無いことを示している。3 種類の説明の内、作成された設計案がどのような根拠を基に作成されたかという設計根拠説明は、説明箇所を特定する必要が無いので、説明箇所の特定制を行うのは前者の 2 種類である。これら 2 種類の説明はそれぞれ伝えたい情報の性質が異なるため、有意性の基

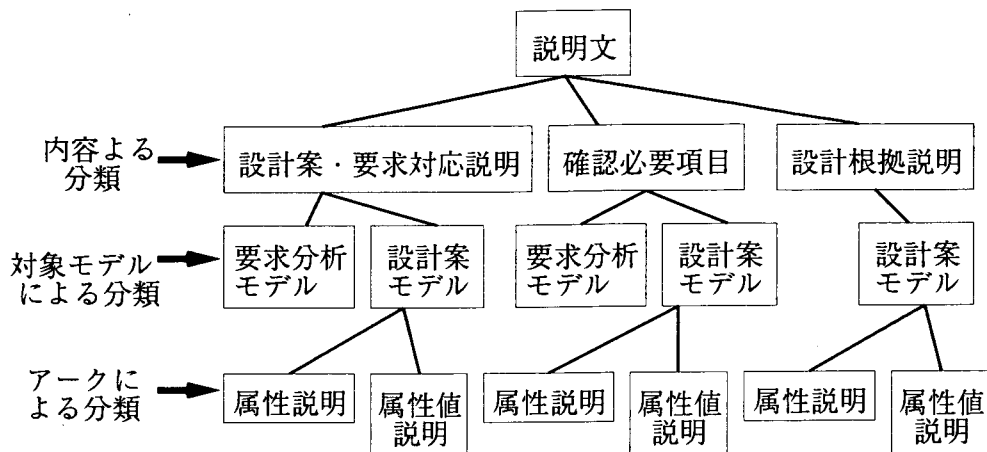


図 5.4: 説明文の種類

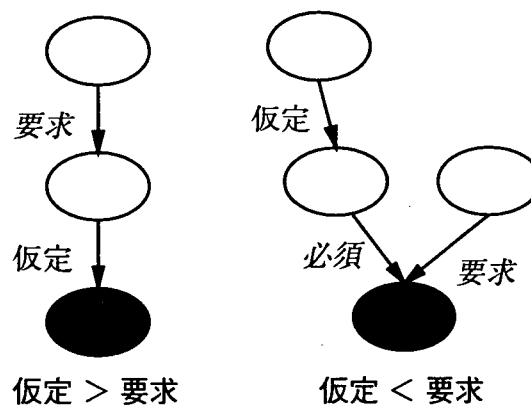


図 5.5: 有意性の判断

準も異なってくる。また、構造的な違いより対象とするモデルによって、有意性の判断基準が異なる。以下にそれぞれの有意性の判断基準について述べる。

「設計案・要求対応説明」で特定すべき内容は、顧客からの要求と、それが直接反映されている設計案の部分との対応である。そこで、意味が「要求」であるアークに注目し、出された要求が仮定を用いずに直接設計案に反映されているかという有意性を判断する。設計案作成過程情報が対象となる場合は、どのようなアークにおいても入力された要求が直接設計案に反映されているため、有意性は全て成立する。要求分析過程情報が対象となる場合は、道筋中のアークにおいて、「要求」の後に「仮定」が存在せずに具体機能ノードに到達していれば有意性があると判断できる。図 5.5の左側では、要求アークに対して具体機能ノードは有意性が無いと判断し、右側では有意性があると判断できる。

「確認必要項目」では、意味が「仮定」であるアークに注目して、計算機が独自においた仮定が設計案に影響を及ぼしているかという有意性の判断をする。設計案作成過程情報においては、アークは直接設計案に影響を及ぼしているので、各ノードにおいて直接的に

影響を受けているアークの意味が「仮定」のアークのみから決定されている場合、有意性があるとする。一方、要求分析過程情報を対象としている場合は、具体機能ノードにおいて、初期入力要求ノードからの全ての道筋において「仮定」アークが含まれておれば、終点ノードは、システムのおいた仮定無しには成り立たないので、各「仮定」アークに有意性が成り立つ。図 5.5 の左側では、道筋は 1 つであり「仮定」が含まれているので仮定としての有意性があると判断できるが、右側では「仮定」を含まない道筋が存在するので有意性が無いと判断される。

以上をまとめると、有意性の判断基準は表 5.1 のようになる。ただし、表中の記号の定義は以下の通りである。

at_{arc} : 注目しているアーク

A : at_{arc} の終点ノードから具体機能ノードまでのアークの集合

B : at_{arc} からたどりつく具体機能ノードと初期入力ノードまでの道筋の集合

C : at_{arc} の終点ノードに入るアークの集合

表 5.1: 有意性の条件

	設計案・要求対応説明	確認必要項目
	$M(at_{arc}) = \text{要求}$	$M(at_{arc}) = \text{仮定}$
要求分析 モデル	アーク $x \in A$ ならば $M(x) \neq \text{仮定}$	道筋 $path \in B$ ならば $M(x) = \text{仮定}$ となるアーク x が $path$ 中に存在する
設計案モデル	なし	アーク $x \in C$ ならば $M(x) = \text{仮定}$

5.3.3 センテンステンプレート

前節で述べた有意性の判断基準に基づいて特定したモデル中の説明箇所をもとに、センテンステンプレートを用いて文章化を行う。センテンステンプレートとは、文章作成のための枠組であり、ノード中の語句を当てはめることによって、自然言語での説明文章が作成される。

3 種類の説明に対応してセンテンステンプレートを用意する。なお、設計根拠説明は、SE の要求に応じて作成する。SE が指定するのは、設計案作成過程情報中のノードである。説明にはそのノードがどのようにして決定されたかを述べるため、指定されたノードに入ってくるアークに注目する。注目したアークの始点と終点ノードについての説明を作成する。なお、アークの意味によって説明を変えなければならない。以上より文章化に用いるセンテンステンプレートは、以下の項目によって決定される。

- 説明の内容
- 対象モデル
- アークの種類 (設計案作成過程情報のみ)
- アークの意味 (設計根拠説明のみ)

表 5.2に確認必要項目のセンテンステンプレート、表 5.3に設計案・要求対応説明のセンテンステンプレート、表 5.4に SE 学習用説明のセンテンステンプレートを示す。センテンステンプレート中の括弧の中には、過程情報中のノード名が記述されることを示す。

表 5.2: 確認必要項目のセンテンステンプレート

モデル	アーク	テンプレート
要求分析		(初期入力要求) において、(仮定内容) を行うと仮定されるので、(機能) が必要であるとしました。
設計案	属性間	(初期入力要求) において、(始点属性)：(始点属性値) には、(終点属性)：(終点属性値) が必要であると仮定しました。
設計案	属性値	(初期入力要求) において、(属性) には (属性値) を用いると仮定しました。

表 5.3: 設計案・要求対応説明のセンテンステンプレート

モデル	アーク	テンプレート
要求分析		(初期入力要求) において、(機能) の実現のために、(属性)：(属性値) が設計案に含まれています。
設計案	属性間	(初期入力要求) において、(始点属性) のために、(終点属性)：(終点属性値) を設定しました。
設計案	属性値	(初期入力要求) において、(属性) には、(属性値) を用いました。

図 5.2 を例に文章化を示す。

確認必要項目として、要求モデル中の特定箇所は、

- 「販売管理」→「収支計算」, 「伝票整理」→「統計データ表示」

表 5.4: SE 学習用説明のセンテンステンプレート

種類	値	テンプレート
属性間	必須	(始点属性) には (終点属性) が必須です。
属性間	仮定	(始点属性) には (終点属性) が必要と仮定しました。
属性間	要求	要求情報より、(始点属性) には (終点属性) が必要としました。
属性間	コスト	コストによる制約によって、(始点属性) には (終点属性) が必要となりました。
属性間	相性	ハード・ソフト相性の制約によって、(始点属性) には (終点属性) が必要となりました。
属性間	トラフィック	トラフィックによる制約によって、(始点属性) には (終点属性) が必要となりました。
属性値	必須	(指定属性) は、(上位属性)：(上位属性値) のために (指定属性値) と決定しました。
属性値	仮定	(属性) には (属性値) を用いると仮定しました。
属性値	要求	要求情報より、(属性) には (属性値) を用いました。
属性値	コスト	コストによる制約によって、(属性) には (属性値) を用いました。
属性値	相性	ハード・ソフト相性の制約によって、(属性) には (属性値) を用いました。
属性値	トラフィック	トラフィックによる制約によって、(属性) には (属性値) を用いました。

であり、これらの特定箇所に対し、確認必要項目用 (表 5.2) の最初のセンテンステンプレートを用い文章化を行うと以下ようになる。

- 販売管理において、収支計算, 伝票整理を行うと仮定されるので、統計データ表示が必要であるとなりました。

次に、設計案・要求対応説明では、

- 「販売管理」→「通信」→「モデム」:「モデム 1」, 「相手先」:「仕入れ先」
- 「販売管理」→「通信」→「相手先」:「仕入れ先」
- 「販売管理」→「相手先」:「仕入れ先」

といった道筋が特定されていたので、表 5.3のそれぞれのテンプレートを用いて、文章化をすると以下ようになる。

- 販売管理において、通信の実現のために、モデム：モデム 1, 相手先：仕入れ先が設計案に含まれています。
- 販売管理において、通信のために、相手先：仕入れ先を設定しました。
- 販売管理において、相手先には、仕入れ先を用いました。

最後に、設計案がどのようにして作成されたかを説明する文章を作成例を挙げる。図 5.3の「Windows」に対して説明の要求があった場合を考えると、「Windows」に入ってくるアークの意味は「必須」であり、属性値接続アークであるので、用いるセンテンステンプレートは表 5.4の 7 番目の

- (指定属性) は、(上位属性)：(上位属性値) のために (指定属性値) と決定しました。

であり、出力結果は

- OS は、ソフト：Oracle Glue, Excel, Lan Maneger, SQL* Net のために Windows と決定しました。

といったものになる。

5.4 説明作成適用例

この節では、具体的な説明作成の例として、要求仕様と設計案をもとにどれだけの説明が作成されるかを示す。まず、本システムへの入力として、以下の情報が与えられているとする。

- 販売管理をしたい。
- 販売管理には受発注業務がある。
- 受発注にはオンライン発注を行いたい。
- カタログ作成をしたい。
- カタログ用の画像データは既存している。
- カタログ作成にはカラーのプリンタを使いたい。
- コンピュータに詳しい社員はいない。

以上の入力を基に、本システムによって、図 5.6 のような設計案が作成されたとする。この設計案を作成するための要求分析過程情報を図 5.7 に示す。また、設計案作成過程情報の一部は図 5.3 に示したものである。

作成される確認必要項目の説明文は、以下の通り要求モデルより 3 文、設計案作成過程情報より 7 文が作成される。

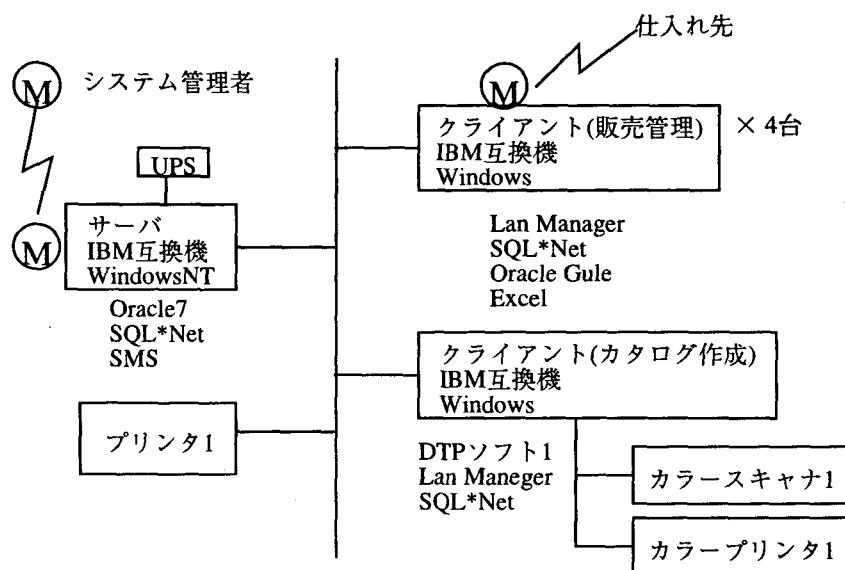


図 5.6: 設計案の例

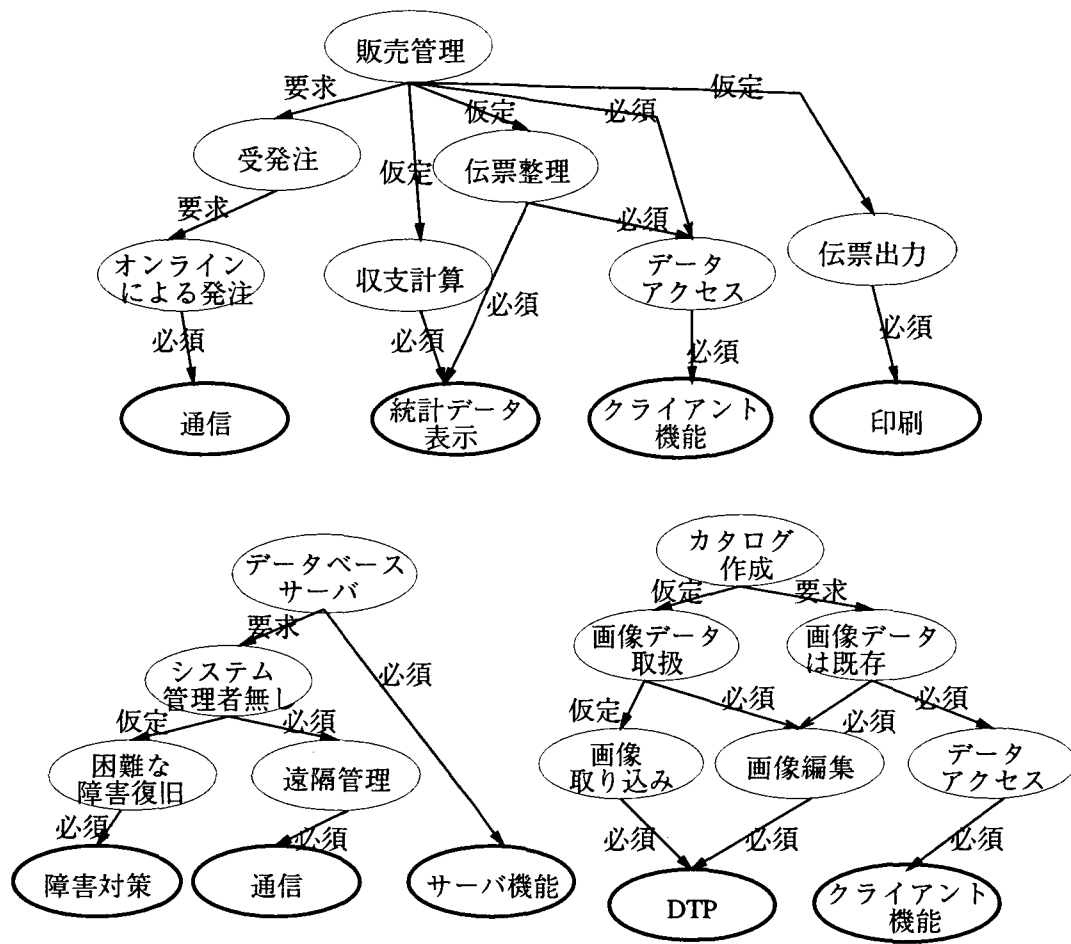


図 5.7: 作成される要求分析過程情報

- 販売管理において、収支計算, 伝票整理を行うと仮定されるので、統計データ表示が必要であるとなりました。
- 販売管理において、伝票出力を行うと仮定されるので、印刷が必要であるとなりました。
- データベースサーバにおいて、困難な障害復旧を行うと仮定されるので、障害対策が必要であるとなりました。
- 販売管理において、機種台数には4台を用いると仮定しました。
- 販売管理において、プリンタにはプリンタ1を用いると仮定しました。
- 販売管理において、プリンタ台数には1台を用いると仮定しました。
- カタログ作成において、DTPには、カラスキャナ：カラスキャナ1が必要であると仮定しました。

- カタログ作成において、機種台数には1台を用いると仮定しました。
- データベースサーバにおいて、無停電電源装置にはUPS1を用いると仮定しました。
- データベースサーバにおいて、機種台数には1台を用いると仮定しました。

設計案・要求対応説明文としては、以下の8文が作成される。

- 販売管理において、通信の実現のために、モデム：モデム1、相手先：仕入れ先が設計案に含まれています。
- データベースサーバにおいて、通信機能の実現のために、モデム：モデム1、相手先：システム管理者が設計案に含まれています。
- カタログ作成において、DTPの実現のために、ソフト：DTPソフト1、カラープリンタ：カラープリンタ1、カラーキャナ：カラーキャナ1が設計案に含まれています。
- カタログ作成において、クライアント機能実現のために、ソフト：Lan Manager, SQL*Netが設計案中に含まれています。
- 販売管理において、通信のために、相手先：仕入れ先を設定しました。
- 販売管理において、相手先には、仕入れ先を用いました。
- データベースサーバにおいて、通信のために、相手先：システム管理者を設定しました。
- カタログ作成において、DTPのために、カラープリンタ：カラープリンタ1を設定しました。

要求分析における説明として要求分析過程情報の目的ノードから具体機能ノードまでのすべての道筋、つまりは要求分析過程を説明すると13文が作成される。また、設計案作成過程情報では、属性間接続アークの数は計53本になる。本手法のように説明箇所の特定を行わず、設計案中のすべての部品に付いて決定理由や役割を説明すると66文も作成される。このため、実際上重要となってくる情報を取捨選択することが困難となる。本手法で作成した説明は全部で18文であり、作成した説明を全て理解することが困難な量ではなく、必要とする情報を確実に入手することができる。

その他の3つの事例について適用し、計4つの事例において作成された説明文数を表5.5に示す。本手法により説明文を作成することにより、説明文の数が可能な文の数の約30%に抑えることができる。

表 5.5: 作成される説明文数の比較

	本手法により作成	全文作成
case 1	18	66
case 2	16	60
case 3	15	55
case 4	22	84

なお、本手法により作成した説明を、SEに提示したところ、「計算機が設定した仮定が明確に示され、要求との対応も示されているので、設計案の理解が助かる」「わかりきったことを説明せず、検討すべき内容についての説明があるので、情報量が多い」といった評を得ている。

5.5 まとめ

本章では、CSS 構成設計支援システム CIDAC において SE に対しての設計技術獲得支援を実現するための教材を生成する主要な機能である設計案分析説明作成部における説明文の検討および作成方法を提案した。説明作成方法の特徴は、構造モデルとして表現された要求分析部の出力である要求分析過程情報と、設計案作成部の出力である設計案作成過程情報とを用いて、要求と設計案との対応を可能にした。また、説明内容を分析し、説明の種類によって説明箇所特定のための有意性の基準を設定し、文章のテンプレートの設定した。説明箇所の特定を行うことで、多量な説明文の作成を防ぎ、主要な説明文のみを提示することができる。

第 6 章

定性シミュレーション結果のシナリオ化 方式

6.1 まえがき

本章では、情報システム計画業務中のシステム導入による効果予測を支援するための、定性シミュレーションシステムにおける出力結果のシナリオ化 [71][72][73] について述べる。

企業などの組織においてその活動がうまくいかない、また、業績が上がらないなどという問題を抱えた場合には、組織を改善するための何らかの戦略を立案する必要がある。組織における戦略立案を行うためには、現状を打開するために考え出された様々な組織改善案に対して、各々の有効性や影響を知ることが重要である。そこで、組織改善案を実施した後、その影響により組織がどうなるかや、何か不都合は生じないかなどを予測する必要がある。これには、対象となる組織における要因間の因果関係を分析して、構造モデルと呼ばれる有向グラフに表すことが有効である。この構造モデルを用いて、各問題の原因に対する系の挙動、および各施策案に対する系の挙動を把握することにより、その組織における現状の分析や今後の戦略の立案を行うことが可能となる。これをシステムティックに実現するため、様々な構造モデル化手法 [10][11][12] が考案されているが、組織の大規模化、複雑化に従い、有効な計算機支援の必要性が認識されている。

因果関係に基づく挙動推定のために計算機を利用した試みとして、システムダイナミックス [13][14][15] に基づくシミュレーション手法が挙げられる。システムダイナミックスとは社会システムを対象に定量的分析を行う方法論である。対象における因果関係は、フローダイヤグラムと呼ばれる流れ図を用いてモデル化される。フローダイヤグラムの中で因果関係は数式やフローとして表現される。作成されたフローダイヤグラムを計算機にかけることにより系の挙動推定が行える。しかしながら、システムダイナミックス

は定量的な数式でモデル化が可能な対象を扱うものであり、組織における要因間の因果関係のように、元来、定量的に捉えることが困難な対象にはあまり適していない。組織における因果関係というものは本質的に定性的である。

そこで、システムダイナミックスと同じように社会システムを対象として、因果関係を定性的に扱うことにより構造を把握し、政策決定の分析に役立てようとする認知構造図 (Cognitive Map) 手法 [16][17] が開発されている。認知構造図手法とは、政策決定者の認知における因果関係の構造を表した認知構造図を作成した上で、これを用いて政策シミュレーションを行うものである。認知構造図における因果関係は、要素間の関係が有るか否か、関係があるときはその関係が正か負かの符合で区別したもので、3 値論理に相当し、関係の強弱に評点値を使った分析に比べて信頼性が高いといえる。このような性質から国際問題 [18]、都市問題 [19][20]、情報システムの要求分析 [21][22] のような分野に適用が試みられている。しかしながら、シミュレーションにおいて、因果関係に評点値が付与されていないため、ある要因に同時に逆方向 (異なる符合) の影響力が作用すると決定不可能な状態が起こる。認知構造図手法におけるシミュレーションでは、このようなときより多くの要因と因果関係のある影響力を採用しているが、その妥当性に対して問題が残る。

一方、人工知能の分野において、定性的モデルに基いた挙動推定や因果解析を目指す、いわゆる定性推論 [23][24][25] に関する研究が活発に行われている。定性推論の研究は物理システムを対象に発展してきた。科学分野では、全ての物理現象を最も精密に説明できる、最も普遍的で根源的な原理を見つけることが至上の目的であるが、工学分野では物理現象をなるべく単純化・抽象化して与えられた問題に関わる部分だけをうまく見つけ出したいことが多い。そこで、人間が普段考えるような単純化、抽象化というような定性的思考を利用しようという定性物理 [26] に関する研究が行われるようになった。物理システムに関わらず、我々の身の回りの生物システムや社会システムをはじめとする様々な動的システムを対象として、定性的な思考過程に焦点を当て、その分析、モデリング、実現について研究するのが定性推論である。定性推論の分野において、特に、動的システムの挙動を定性的な情報に基づいて予測する技術の研究として定性シミュレーション手法が挙げられる。これらの研究は、本来、定量的に捉えることが可能な対象に対して、定量値を定性値に写像することにより、不完全な情報からの挙動推定、より一般的な推定結果の獲得、複雑な対象における挙動の大局的把握などを実現するものである。このため、組織における要因間の因果関係のように本質的に定性的評価しかできない対象に対しては、定量的空間における特徴点を反映した境界標を設定できない。また、定性的情報に起因して発生した曖昧性に対し、物理的制約や定量的情報などを利用できない。以上の理由から、状態数の膨大化に伴う計算効率の悪化や、多数の冗長な挙動が導出されることが懸念され、その適用は極めて困難である。

組織における諸問題や施策案といった、本質的に定性的にしか評価できない対象についての挙動推定を目的とした計算機支援の試みとして、畑らは定性的な構造モデルを用いた類型挙動パターン帰着型定性シミュレーション手法 [35][36][37] を開発した。類型挙動パターン帰着型定性シミュレーション手法では、影響の伝播速度をもとにモデルを分割し、部分系毎に挙動をパターンで捉え、状態数の削減、導出される挙動の限定を行うことにより、効率的なシミュレーションを実現している。この類型挙動パターン帰着型定性シミュレーションシステムにおいて出力結果は、各要因における挙動把握を容易にするため、シミュレーション結果として導き出された定性値を基に描かれたグラフで与えられる。しかし、シミュレーション結果を意志決定に用いる場合、各要因における挙動を知るだけでは不十分であり、どのような影響をもとに得られた挙動であるかといった因果の流れについての分析が重要となる。この作業は構造モデルの複雑化、大規模化に伴い大変困難な作業となっている。

このようなシミュレーション結果の検討といった作業を軽減する方法として、シナリオを用いて、シミュレーション結果がどのような影響をもとに出された挙動であることを説明する機能を付け加えることを提案する [71]。この機能を導入することで、意志決定に用いるための因果の流れについての情報を直ちに知ることが可能となる。

シナリオの作成、説明生成などにより、結果の理解及び検討といった労力の軽減を目指したシステムに関する研究は初期の医療診断エキスパートシステム MYCIN 以来、様々なものが開発されてきた [38][39][40]。これらは、主に結果を導く際に利用されたルールの連鎖をたどることにより、その結果がどのようにして得られたかを説明するものである。しかしながら構造モデルに基づく定性シミュレーションシステムにおいては、システムの挙動は一連のルールの適用によって定められるというよりは、むしろ因果関係に基づく影響伝播の組合せとして統合的に導かれるものであり、可能な影響伝播の中でどの影響伝播が本質的であるかを特定することが重要となる。

そこで、定性シミュレーションシステムにおいてシナリオ生成機能を実現させるために、シミュレーション結果、構造モデル、利用者からの施策案の実施といった入力情報、注目している内容についての情報をもとに、注目している事柄へ実際に影響が伝播されたと考えられる因果関係を導き出すというアプローチをとる。シミュレーション結果として説明すべき本質的な因果関係の抽出には、シミュレーション結果の各ノードにおける挙動をもとに、シミュレーション実行時に影響伝播が存在したかどうか推測を行う。要因間で影響が伝播されていれば、各要因におけるシミュレーション結果の挙動に類似性が見られる。よって、実際に影響が伝播されている因果関係は、ノード間における挙動の類似性より推測ができる。各要因における挙動を注目している事柄の挙動と比較し、その類似性を検討することで、実際に注目している事柄に対して影響が伝播された因果関係かどうか判断が可能となり、多数ある因果関係の中から、本質的な影響伝播を支配する因果関係の抽

出を実現する。このようにして導き出された因果関係を粗筋としシナリオを構成する。

本章では、第 6.2 節に定性シミュレーションに用いる構造モデルについて、第 6.3 節に用いた類型挙動パターン帰着型定性シミュレーション、第 6.4 節にシナリオ生成の手法について説明し、第 6.5 節にその実行例を述べる。

6.2 構造モデル

6.2.1 因果関係の種類

構造モデルとは、要因間の因果関係を有向グラフで表したものであり、図 6.1 に示すようにノードに表された要因間の因果関係がアークを用いて明示される。

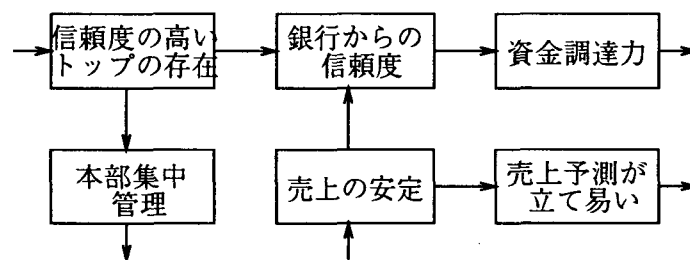


図 6.1: 構造モデルの例

図 6.1 のような構造モデル上では、売上が安定すれば売上予測が立て易いといった不変的な因果関係と、売上が確実に安定するまで銀行からの信用が上がらないが確実に安定すれば信用が得られるといった、状況に左右される因果関係が存在する。構造モデルを用いて定性シミュレーションを行なうに際し、このような因果関係の種類を明確にする必要がある。構造モデル上にどのような因果関係が存在するかを明確化するために、因果関係を分類した。分類に用いた構造モデルは、ノード数 27、アーク数 44 の比較的大規模な構造モデルである。その結果、大きく次の 3 種類に分類された。

1. 因果関係が不変的であり、要因間の影響は一定であるもの。
2. 影響を受ける側の要因に、影響を受ける度合の上下限が考えられるもの。
3. 影響を与える側の要因が大きく変化しないと、影響が考えられないもの。

これらの因果関係を構造モデル上で表現するため、以下の 3 種類のアークを用いる。ここで、原因ノードとは影響を与えるノード、結果ノードとは影響を受けるノードである。

1. 原因ノード、結果ノードの状態に関わらず、影響が伝播されるアーク。(図 6.2)

2. 結果ノードの状態が大きく変化するまでは、原因ノードからの影響が伝播されるが、状態が大きく変化すると影響伝播がなくなるアーク。(図 6.3)
3. 原因ノードの状態が大きく変化するまで、結果ノードへは影響が伝播されず、状態が大きく変化して初めて影響が伝播されるアーク。(図 6.4)



図 6.2: アークの種類 (a)

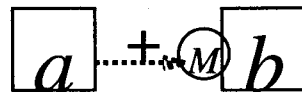


図 6.3: アークの種類 (b)

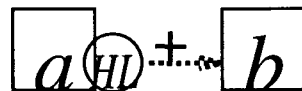


図 6.4: アークの種類 (c)

3 種類のアークの中で、後者 2 つはノードの状態が条件となり影響伝播の有効、無効を決定している。そこで、これらのアークによる影響伝播を、条件付影響伝播と呼ぶ。

6.2.2 定性的情報

構造モデルを用いて定性的にシミュレーションを行なうためには、構造モデルを構成するノードやアークに対し、パラメータとなる定性値を持たせる必要がある。定性的情報としてノードには、定性的状態値と状態の変化傾向が与えられ、アークには影響の方向と、影響の伝播条件に関する情報が与えられる。

(1) 定性的状態値 ノードのある時点における絶対的な状態を定性的に表した情報である。ここでは、特別な境界標を設定せずに、以下の 3 種類の定性値を定義する。また、

定性的状態値は条件付影響伝播において条件に対応するものであり、この値が変化することがノードにおいて状態が大きく変化していることを表す。

定義 1 (要因の定性的状態値) ノード a の時間 t における定性的状態値は $[a(t)]$ と記され、以下のように定義される。

$$[a(t)] = \begin{cases} \text{H: 十分に増加した場合} \\ \text{M: 増加も減少も十分に行なえる場合} \\ \text{L: 十分に減少した場合} \end{cases}$$

(2) 状態の変化傾向 ノードの状態の変化の方向を定性的に表した情報である。これは、状態値の定性的時間微分に相当し、以下の 3 種類の値を用いる。

定義 2 (状態の変化傾向) ノード a の時間 t における状態の変化傾向は $[\partial a(t)]$ と記され、以下のように定義される。

$$[\partial a(t)] = \begin{cases} \text{I: 状態値が増加しているとき} \\ \text{S: 状態値が一定のとき} \\ \text{D: 状態値が減少しているとき} \end{cases}$$

(3) 影響の方向 影響の方向とは、原因ノードの状態の変化が結果ノードの状態に与える影響の方向を表す情報であり、以下の 2 種類の値を用いる。

定義 3 (影響の方向) 原因ノード a から結果ノード b への影響の方向は $D(a, b)$ と記され、以下のように定義される。

$$D(a, b) = \begin{cases} +: a \text{ の状態値が増加すると} \\ \quad b \text{ の状態値も増加する} \\ -: a \text{ の状態値が増加すると} \\ \quad b \text{ の状態値は減少する} \end{cases}$$

(4) 影響の伝播条件 影響の伝播条件は、原因ノードと結果ノードとの因果関係の種類に関する情報であり、以下の 3 種類の値を用いる。

定義 4 (影響の伝播条件) 原因ノード a から結果ノード b への影響の伝播条件は $C(a, b)$ と記され、以下のように定義される。

$$C(a, b) = \begin{cases} \text{n: 伝播条件なし} \\ \text{m: } [b(t-1)] = \text{M のとき時間 } t \text{ において} \\ \quad \text{影響伝播が行われる} \\ \text{h: 時間 } t \text{ において } [a(t)] = \text{H, L の} \\ \quad \text{とき影響伝播が行われる} \end{cases}$$

6.3 類型挙動パターン帰着型定性シミュレーション

6.3.1 シミュレーションの概要

シミュレーションは、構造モデル内のある要因（ノード）の状態の変化に対する時間経過にともなう系全体の挙動の導出を目的とする。この手法では、因果の伝播速度を基準とした時間スケール毎に系の挙動を類型化して捉え、これを統合することにより、全体の挙動を導出する。すなわち、ある時間スケールにおけるシミュレーションにおいて系が平衡状態に達すると、その結果より状態値の移行し、移行した状態値により要因間の因果関係を変化させ、次時間スケールのシミュレーションを行う。ここで、平衡状態とは、既出の状態が導出され、以後同じ状態変化が繰り返されることを意味する。状態値の移行は、平衡状態をプリミティブな挙動パターンで捉えた類型挙動パターンによって決定する。類型挙動パターンは表 6.1 に従い特定され、それらをもとに状態値が、パターンが I^* の場合 $M \rightarrow H$, $L \rightarrow M$ と移行され、 D^* の場合 $H \rightarrow M$, $M \rightarrow L$ と移行される。

表 6.1: 類型挙動パターンの特定

繰り返し部の変化傾向	類型挙動パターン
S のみが出現	S^*
I が出現し, D が非出現	I^*
D が出現し, I が非出現	D^*
I, D が出現し, I が先出	$(ID)^*$
I, D が出現し, D が先出	$(DI)^*$

時間スケールは文献 [35] のようにモデル内のアークに明示的に与えた伝播速度から決定することも可能であるが、ここでは前節で述べた条件付影響伝播を利用する。すなわち、条件付影響伝播の利用によって起こる構造モデル上の変化を区切りとして、1つの時間スケールを構成する。そして、各時間スケールに対して、条件を満たすアークのみからなる部分モデルをその時間スケールにおけるモデルと考え、因果伝播による挙動の導出を行う。

6.3.2 シミュレーションの動作と例

最初の時間スケールに対応するシミュレーション段階を $T = 1$ とし、初期状態として各ノードに対して、時刻 $t = 0$ における定性的状態値、状態の変化傾向を与える。さらに、あるノードに施策案実施に相当する変化傾向を与え、シミュレーションを開始する。与え

られた変化傾向をもとに、伝播規則に従い、影響伝播し、全てのノードの状態が以前の状態と一致したならば、ノード毎に類型挙動パターンを表 6.1 に従い決定する。特定された類型挙動パターンをもとに、ノード毎に定性的状態値を移行し、次の時間スケールに対応するシミュレーションを行う。定性的状態値が移行されるノードがなければ、シミュレーションを終了する。シミュレーションにより、結果として、各ノードの変化傾向、および定性的状態値の系列が得られる。

図 6.5 の構造モデルにおいて各ノードの変化傾向、定性的状態値を各々 S, M に設定した後、「人件費」の変化傾向を I にして、シミュレーションを実行した。ただし、「企業利益」への影響伝播については、(「効率」の影響力) > (「経費」の影響力) > (「生産性」の影響力) と指定した。実行結果により得られた変化傾向の系列と定性的状態値を表 6.2 に示す。

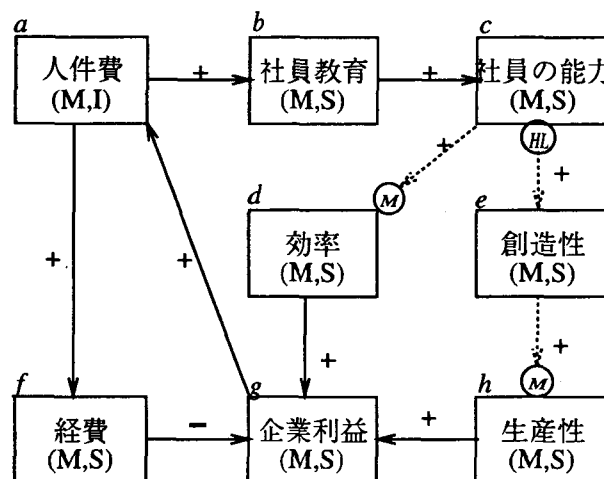


図 6.5: シミュレーションに用いる構造モデル

「社員の能力」から「創造性」へのアークの伝播条件 $C(c, e) = h$ より、 $T = 1$ において「社員の能力」の定性的状態値が M であるので「社員の能力」から「創造性」には影響伝播が行なわれない。 $T = 2$ において「社員の能力」の定性的状態値が H となるので「社員の能力」から「創造性」に影響が伝播される。また、「社員の能力」から「効率」へのアークは、伝播条件 $C(c, d) = m$ であるので、 $T = 1$ においては、「効率」の定性的状態値が M であるので「社員の能力」ノードから「効率」へ影響が伝播されるが、 $T = 2$ においては、「効率」ノードの定性的状態値が H となるため、「社員の能力」ノードから「効率」へ影響は伝播されない。

また、それらの結果をもとに描いたグラフ (縦軸: 定性的状態値, 横軸: 時間) は図 6.6 にように示される。

表 6.2: 実行結果

ノード	$T = 1$		$T = 2$	
	変化傾向の系列	定性的状態値	変化傾向の系列	定性的状態値
人件費	S(I)*	M	I(ID)*	H
社員教育	S(I)*	M	I(ID)*	H
社員の能力	S(I)*	M	I(ID)*	H
効率	S(I)*	M	I(S)*	H
創造性	S(S)*	M	S(ID)*	M
経費	S(I)*	M	I(ID)*	H
企業利益	S(I)*	M	I(ID)*	H
生産性	S(S)*	M	S(ID)*	M

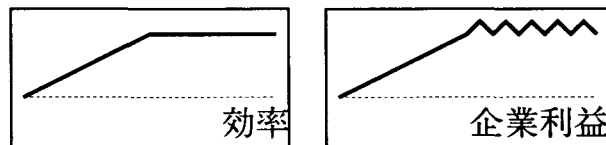


図 6.6: 結果のグラフ表示

6.4 シナリオ生成機能

6.4.1 シナリオ生成機能の概要

シミュレーションを意志決定に用いる際には、施策案の実施によりノードに記述された各要因がどのような挙動を示すかだけでなく、むしろ注目している要因の挙動がどのような理由に基づいて導かれたかが重要となる。そのような挙動を導く原因について、各要因間の影響伝播をもとに順を追って文章により説明したものをここではシナリオと呼ぶ。

シナリオを生成するためには、注目している要因に影響を与えた一連の因果の流れを導出する必要がある。一般に、この因果の流れは、施策案実施に相当するノード (入力ノード) から注目している要因のノード (注目ノード) までの影響伝播のつながりとして捉えることができるが、モデル中で入力ノードから注目ノードまでの因果関係のつながりは一通りではないため、それらの中のどの流れをシナリオに盛り込むかが重要となる。ここではシミュレーション結果を参照し、ノードの挙動の類似性に着目することにより、本質的な影響伝播を与えた因果の流れを特定する。

特定された因果の流れの中には、通常多数のノードが埋め込まれており、その全てについて述べると、大規模な構造モデルでは非常に長いシナリオを生成し、理解しにくく、また、影響の流れを伝える上で、冗長な情報を与えることになる。そこで、そのような冗長な情報を与えないために、因果のつながりの中から、シナリオに述べるべきノードを選出する。このとき、シミュレーション実施時におけるノードの果たす役割に着目し、影響の流れを伝えるのに重要なノードのみを選出する。

特定された因果の流れ、ならびにその中に含まれる適切なノードをもとに文章を構成し、最終的なシナリオを生成する。この時、シミュレーション結果そのものをノード毎に文章化し、それを並べるだけでは、影響の流れを表現するのに不適切であり、また、単調な文章となり読みにくい。そこで、ノードの役割に応じた文章の形を予め決めておくことにより、スムーズな文章の作成を実現する。

以上述べたように、シナリオ生成は

1. 因果の流れの特定
2. 重要ノードの選出
3. 文章化

の各ステップにより実現される。以下その詳細について述べる。

6.4.2 因果の流れの特定

ここでは、最初に入力ノードから注目ノードまでのすべての因果関係のつながりを列挙し、その中から本質的な影響伝播が生じた因果関係をたどることにより、重要な因果の流れの導出を試みる。

入力ノードから注目ノードまでの全ての因果関係の列挙は単純なグラフの探索問題と捉えることができ、深さ優先探索で実現する。

次に、探索により求めた入力ノードと注目ノードの間に存在するすべてのパスの中から、シナリオ構成に用いるために重要となる因果関係の強いパスを選択する。本手法で用いている定性シミュレーションシステムでは、ある要因において複数の要因から影響を受ける場合、それらの要因の中からただ1つの要因からのみ影響が伝播される。よって、シナリオ構成に用いるべき因果関係は1つに特定できる。また、実際に影響が伝播されている因果関係は、ノード間における挙動の類似性より推測することができる。より強い因果関係をもつアークは、実際にシミュレーション実行時に影響を伝播しており、シミュレーション結果は同様な挙動を見せると考えられる。本論文で用いている定性シミュレーション手法では、類型挙動パターンを用いており、シミュレーション結果の挙動をパターン化している。このパターンを用いることで、要因毎に挙動の類似性を容易に検討できる。

よって、類型挙動パターンを比較することにより、実際に影響が伝播された因果関係かどうか容易に推測ができる。図 6.7 で示す通り、シミュレーション結果を時間を横軸、定性的状態値を縦軸にとったグラフ表示で表した場合、ノード z に対してはノード x, y といった 2 つのノードからの影響が考えられるが、ノード z においてはシミュレーション結果がノード x と同様な挙動を見せており、明らかにノード x からの影響を大きく受けたことがわかる。シミュレーション実行時に大きな影響を受けたということは、その要因間ではより強い因果関係を持つことが予測される。そこで、このことを利用し、注目ノードから順に各パス中のノードをたどり、パス中の全てのノードが同様の挙動をしているならば、そのパスは強い因果関係を持つパスと考える。

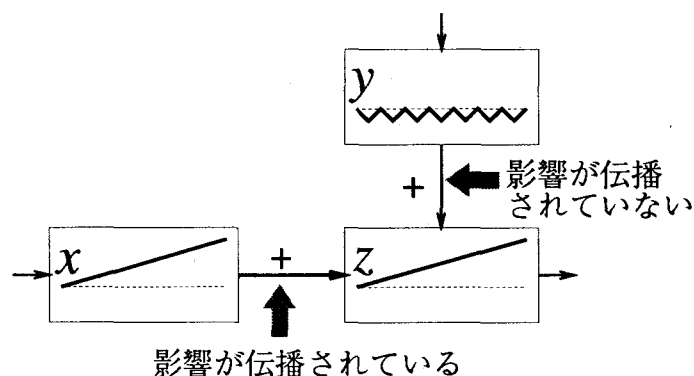


図 6.7: 因果関係の強弱

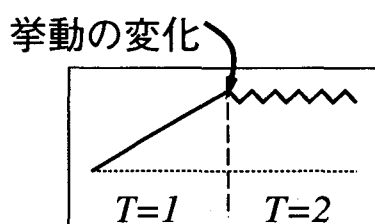


図 6.8: シミュレーション段階による挙動の変化

ただし、注目ノードにおいて図 6.8 に示すように、挙動がシミュレーション段階によって変化している場合には、それぞれのシミュレーション段階において、影響伝播された経路が違う可能性がある。そこで、まず注目ノードにおける挙動を参照し、挙動が一様であればパスを 1 つだけ選択し、シミュレーション段階によって挙動が違っておれば、段階毎に挙動を比較し、複数のパスを選択する。以下にその手続きを示す。

<パスの選択>

Step1: $T = 1$ より行なう。

Step2: 注目ノードの T におけるシミュレーション結果を参照する。 $n = 1$ より行なう。

Step3: $\text{path}(n)$ で注目ノードに近いノードを x とする。

Step4: x の T におけるシミュレーション結果を参照し、注目ノードの結果とを、類型パターンを用いて比較する。類型パターンを $\{I^*, D^*\}, \{S^*\}, \{(ID)^*, (DI)^*\}$ という3つのグループにわけ、同じグループの類型挙動パターンが結果として得られておれば、結果が同様であるとみなす。

Step5: 結果が同様であれば、 $\text{path}(n)$ を x から1つもどり、そのノードを x とし、**Step4** が繰り返される。結果が違っている場合、または x が入力ノードであれば、 $n = n + 1$ として、**Step3** より行う。全てのパスについて行なえば比較終了し **Step6** へ。

Step6: パス中のすべてのノードについて比較が行なわれているパスが1つであれば、そのパスを選択、**Step9** へ。複数存在すれば、**Step7** へ。存在しなければ、**Step8** へ。

Step7: 入力ノードまで比較が行なわれたパスについて、アークの強度の比較を行なう。パス中のノードを結ぶアークのうち、より注目ノードに近いアークにおいてアークの強度が強いものを選択。**Step9** へ。

Step8: 新たにパスの設定を行なう。設定されるパスは、注目ノードと同様なシミュレーション結果をもつノードのみから構成されるフィードバックループを検出し、パスとして選択する。

Step9: 選択されるパスが1つの場合はパス選択処理終了。複数選択される場合は、注目ノードのシミュレーション結果が変化した T について **Step2** 以降を繰り返す。

図 6.5 の構造モデルを用いて具体的な例を挙げる。表 6.2 より、注目ノード「企業利益」の結果は、類型挙動パターンが $T = 1$ では I^* であるが、 $T = 2$ では $(ID)^*$ である。よって、パスを2つ選択する必要がある。図 6.9 に示すように、 $T = 1$ における、注目ノードの類型挙動パターンとパス中のノードの類型挙動パターンの比較する。その結果、入力ノードまで比較されたパスは $\text{path}(2), \text{path}(3)$ の2つである。「企業利益」は「効率」からの影響を強く受けていると考えられるので、 $\text{path}(2)$ を重要となるパスとし、シナリオ構成に用いる。また、 $T = 2$ における類型挙動パターンの比較では、 $\text{path}(1), \text{path}(3)$ の2つが入力ノードまで比較されている。 $(\text{「経費」の影響力}) > (\text{「生産性」の影響力})$ であるので、 $\text{path}(3)$ がシナリオ構成に用いられるパスと決定される。

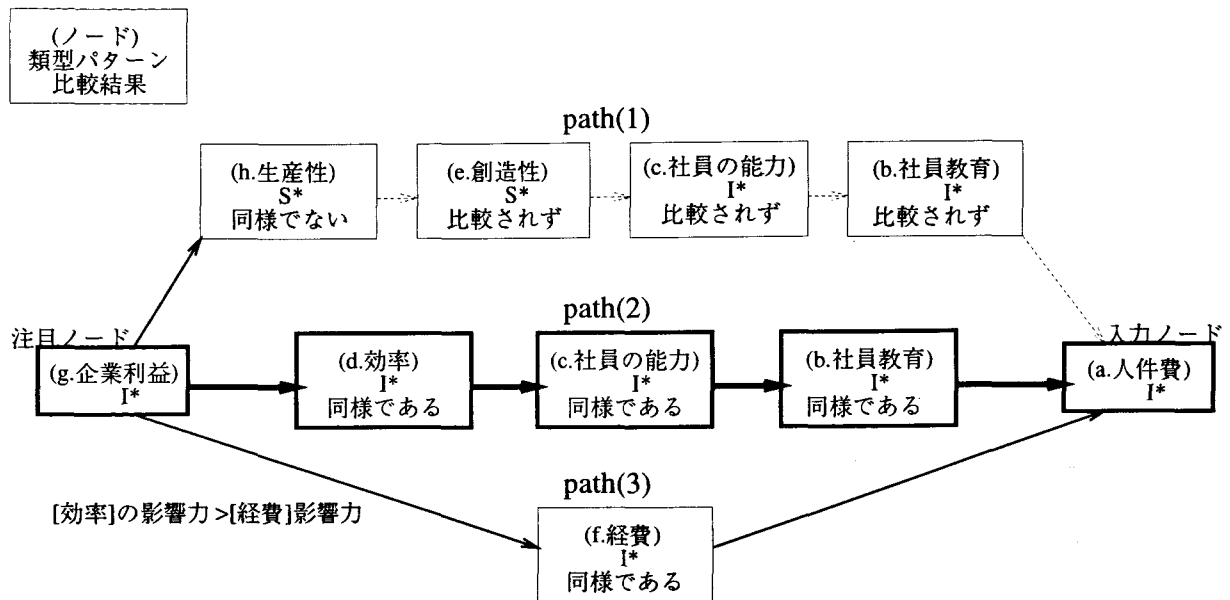


図 6.9: $T = 1$ におけるパスの選択

表 6.3: シナリオ構成における重要性

ノード x の種類	$\text{reason}(x)$
入力ノード	input
注目ノード	attention
設定されたパス中で条件付影響伝播により 影響を受けるノード	condition
影響伝播の衝突を起こす、 または衝突が起こるノード	conflict
注目ノードと直接因果関係が 結ばれているノード	direct
その他のノード	nothing

6.4.3 重要ノードの選出によるシナリオ構成の生成

選択されたパス中で重要となるノード、因果の流れを説明する上でポイントとなるノードを選択し、シナリオ構成を生成する。シナリオ構成とは、パスの流れに沿ったノードのリストであり、作成すべきシナリオの骨格を成すものである。重要となるノードには

表 6.3に示した「その他のノード」以外の5種類がある。表中に示された $\text{reason}(x)$ とは、ノード x に定義される重要性である。この重要性とは、そのノードが何故重要となってくるかという情報であり、後段のプロセスでノードの種類に応じた文章の作成を実現するため、シナリオ構成に組み込まれる。 $\text{reason}(x)$ は x によっては複数定義される場合がある。

選択された全てのパスにおいて入力ノードまたはパスの最初のノードより、各ノード毎に $\text{reason}(x)$ を決定し、 $\text{reason}(x)=\text{nothing}$ となるノード以外をその reason とともにシナリオ構成に加える。よって、シナリオの構成形式は、 $(x, \text{reason}(x))$ の列として作られる。このようにして、作られたシナリオ構成に基づき文章化し、シナリオを生成する。

6.4.4 シナリオ構成の文章化

生成したシナリオ構成をもとに、その中に含まれる各ノードのシミュレーション結果を参照しながら、文章化をすすめる。ここでは、文章の雛型であるセンテンステンプレートを用意し、ノードの内容やその挙動に応じた単語を埋め込むことによりこれを実現する。

センテンステンプレートは、シナリオ構成の構造に依存して準備する必要がある。すなわちシナリオ構成は大きくわけて注目ノードが一連の因果関係のつながりから影響を受け、シナリオ構成が1つのパスから構成されている場合と、シミュレーション段階毎の複数のパスから構成されている場合がある。また、複数のパスより構成されている場合には、それが最初の入力時による影響の伝播経路であるか、ループにより構成された伝播経路であるかなど、それぞれに応じた状況説明が必要となる。そこで、シナリオ構成中の構造に応じ、以下の4種類のセンテンステンプレートを準備した。

Group1 一つのパスのみから全体のシナリオ構成が作られている場合に適用するテンプレート

Group2 複数のパスからなるシナリオ構成の最初のパスに適用するテンプレート

Group3 複数のパスからなるシナリオ構成の2つ目以降のパスに適用するテンプレート

Group4 ループの検出によって設定されたパスに適用するテンプレート

各テンプレート群には、ノードがシナリオ構成に含まれた理由に応じたテンプレートが用意される。表 6.4に用意したテンプレートを示す。また、表中にテンプレートの要素には以下の様な意味がある。

(input-node) 入力ノードの名前を参照

(input-tendency) 入力ノードにおいて始めに指定された変化傾向を参照

(result) そのノードにおけるシミュレーション結果の類型パターンを参照

(node) ノードの名前を参照

(condition-E) 条件付影響伝播のアーキにおいて条件とその利用状況を参照

(conjunction) 適当な接続詞をいれる

(condition-C) 条件付影響伝播のアーキにおいて条件とその利用状況、用いたパス中以外でそのノードに影響を与えているノードの名前を参照

表 6.4: センテンステンプレート

reason	sentence template
(a) Group1	
input	(始めに)⇒(input-node)⇒(input-tendency)⇒(、) ⇒(その後は)⇒(result)⇒(。)
condition	(node)⇒(condition-E)⇒(、)⇒(result)⇒(、) ⇒(condition-E)⇒(result)⇒(。)
conflict or direct	(conjunction)⇒(node)⇒(result)⇒(。)
condition-conflict	(name)⇒(condition-C)⇒(、) ⇒(result)⇒(、)(condition-C)⇒(、) ⇒(result)⇒(。)
attention	(その結果)⇒(attention)⇒(result)⇒(。)
(b) Group2	
input	(はじめに)⇒(input-node)⇒(input-tendency)⇒(、)
condition or conflict or direct	(node)⇒(result)⇒(、)
attention	(その結果)⇒(attention)⇒(result)⇒(。) ⇒(その後は)⇒(attention) ⇒(因果関係の変化により以下のようにになる。)
(c) Group3	
attention	(その結果)⇒(attention)⇒(result)⇒(。)
attention 以外	(node)⇒(result)⇒(、)
(d) Group4	
attention	(その結果)⇒(attention)⇒(result)⇒(。)
attention 以外	(loop-node)⇒(が含まれるループの影響で、)

表 6.5: 単語辞書 (Group1)

要素	参照される情報	抽出される単語
input-tendency	変化傾向 I 変化傾向 D	しばらく増加させると しばらく減少させると
result (類型挙動 パターンを参照)	$I^* (T = 1)$ $D^* (T = 1)$ $T = 1$ のとき S^* $(ID)^*, (DI)^* (T = 1)$ $I^* (T > 1)$ $D^* (T > 1)$ $S^* (T > 1)$ $(ID)^*, (DI)^* (T > 1)$	始めは増加するが 始めは減少するが 始めは一定となり 始めはほぼ変化せず 増加する 減少する 一定になる 不安定であるがほぼ 一定になる
condition-E	伝播条件 h 利用 off 伝播条件 m 利用 off 伝播条件 m,h 利用 on	(node) の変化が 小さいと影響を 受けないので ある程度変化すると (node) からの影響を 受けないので (node) からの 影響を 受けるので
conjunction		その影響で
condition-C	条件 h 利用 off 条件 m 利用 off 条件 m,h 利用 on	(node) の変化が小さい と影響受けず (node) からの影響を 受けるので ある程度変化すると (node) からの影響を 受けず (node) からの 影響を受けるので (node) からの影響を 受けるので

表 6.6: 単語辞書 (Group2,Group3,Group4)

テンプレート中の要素	参照される情報	抽出される単語
input-tendency	変化傾向 I 変化傾向 D	しばらく増加させると しばらく減少させると
result (reason $\neq$ attention)	類型挙動パターン I* 類型挙動パターン D* 類型挙動パターン S*,(ID)*,(DI)*	増加し 減少し 変化せず
result (reason = attention)	類型挙動パターン I* 類型挙動パターン D* 類型挙動パターン S* 類型挙動パターン (ID)*,(DI)*	増加する 減少する 一定となる 不安定であるが ほぼ一定となる

(attention) 注目ノードの名前を参照

シナリオ構成中の全てのノードに対し、選ばれたテンプレートを基に、ノードの名前や、シミュレーション結果を参照し、単語辞書より単語を抽出し、テンプレート中に当てはめ文章化する。単語の抽出に用いられる単語辞書を表 6.5ならびに表 6.6に示す。

6.5 シミュレーション及びシナリオ生成例

6.5.1 実行例

提案手法を計算機上に実装し、4通りの入力に対してシミュレーション及びシナリオ生成実験を行った。適用したモデルを図 6.10に示す。これはある企業における人材教育の充実、マーケティング、品質の向上、CMの製作といった施策案が収益にどのように影響するかを表した構造モデルである。構造モデル上で複数から影響を受けているノードに対する影響の強弱は表 6.7 の様に設定する。設定されないものは影響の強弱は同等である。また、各ノードの変化傾向、定性的状態値の初期値を各々S,Mに設定し、注目ノードを「収益」ノードと設定した。

(1) 実行例 1 入力として「人材教育の充実」ノードに $t=1\sim 2$ において、変化傾向Iを与える。つまり、人材教育の充実させるという施策案を実行する場合。

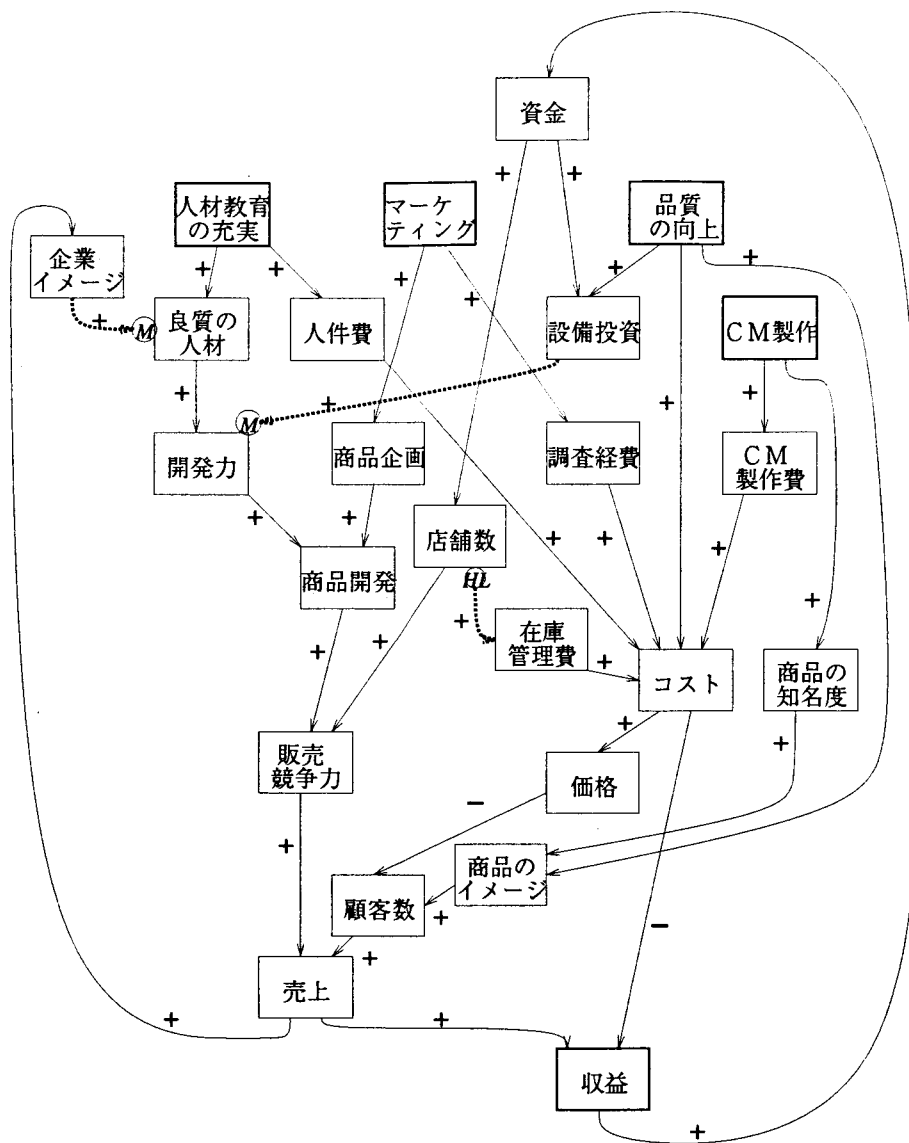


図 6.10: 実験に用いた構造モデル

表 6.7: 影響の強弱

影響を受ける	影響の強い	影響の弱い
開発力	良質の人材	設備投資
販売競争力	商品開発	店舗数
売上	販売競争力	顧客数
収益	コスト	売上

- (2) 実行例 2 入力として「マーケティング」ノードに $t=1\sim 2$ において、変化傾向 I を与える。つまり、マーケティングという施策案を実行する場合。特に、「商品開発」ノードにおいて「商品企画」ノードからの影響が「開発力」ノードからの影響より強いと設定した。
- (3) 実行例 3 入力として「品質の向上」ノードに $t=1\sim 2$ において、変化傾向 I を与える。つまり、品質の向上を計るという施策案を実行する場合。
- (4) 実行例 4 入力として「CM製作」ノードに $t=1\sim 2$ において、変化傾向 I を与える。つまり、CM製作による広告をするといった施策案を実行する場合。

それぞれの実行例におけるシミュレーション結果を図 6.11, 6.12, 6.13, 6.14 に示す。また、出力されたシナリオは以下のものであった。

(1) 実行例 1

始めに、人材教育の充実をしばらく増加させると、その後は、良質の人材は増加し、開発力は増加し、商品の開発は増加し、販売競争力は増加し、売上は増加し、その結果、収益は増加する。その後は、資金と店舗数と在庫管理費とコストを含むループの影響で、収益は不安定であるがほぼ一定になる。

(2) 実行例 2

始めにマーケティングをしばらく増加させると、その後は、商品企画は変化せず、資金と設備投資と販売競争力と売上とを含むループの影響で、収益は増加する。その後は、資金と店舗数と在庫管理費とコストとを含むループの影響で、収益は不安定であるがほぼ一定になる。

(3) 実行例 3

始めに品質の向上をしばらく増加させると、その後は、開発力は増加し、商品開発は増加し、販売競争力は増加し、売上は増加し、その結果、収益は増加する。その後は、収益への因果関係の変化により以下のようなになる。コストは不安定であるがほぼ変化せず、売上は少し増加するがほぼ一定となり、その結果、収益は少し増加するがほぼ一定となる。

(4) 実行例 4

始めに CM 製作をしばらく増加させると、その後は、CM 製作費は変化せず、コストは変化せず、資金と設備投資と開発力と商品開発と販売競争と売上とを含むループの影響で、収益は少し減少するがほぼ一定になる。その後は、収益への因果関係の変化より以下のようなになる。コストは変化せず、その結果、収益は一定になる。

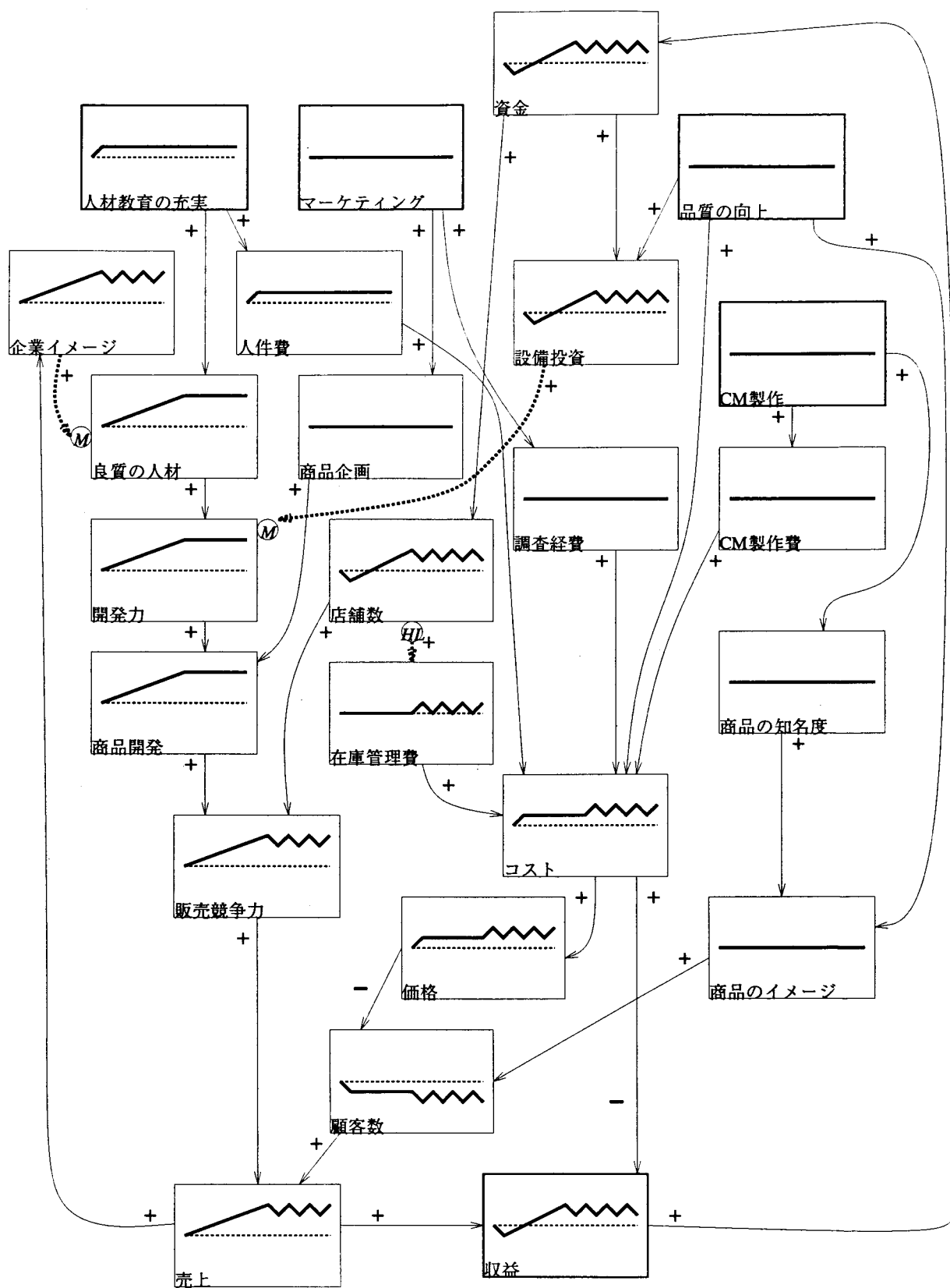


図 6.11: 実行例 1 の各ノードにおける挙動グラフ

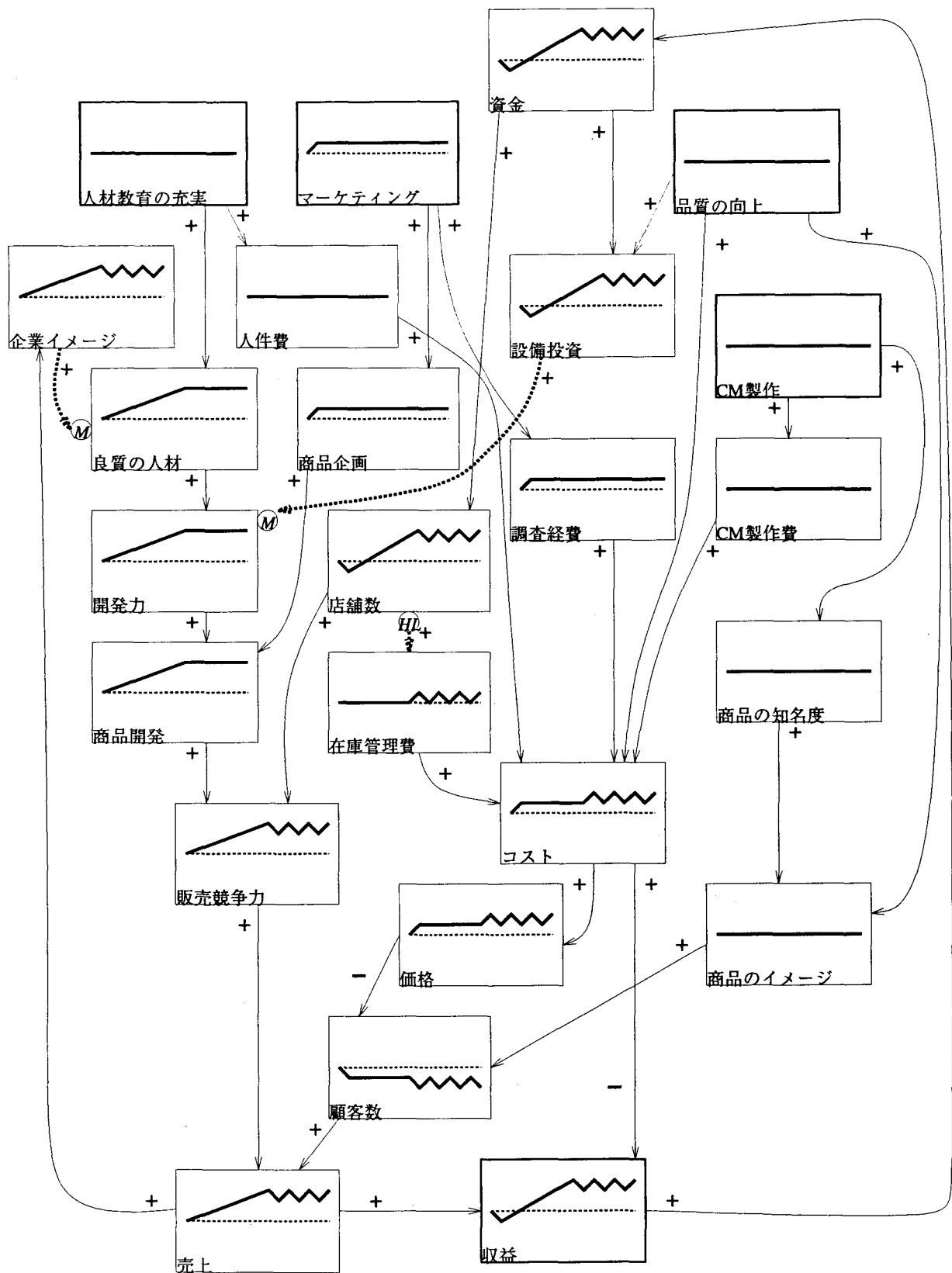


図 6.12: 実行例 2 の各ノードにおける挙動グラフ

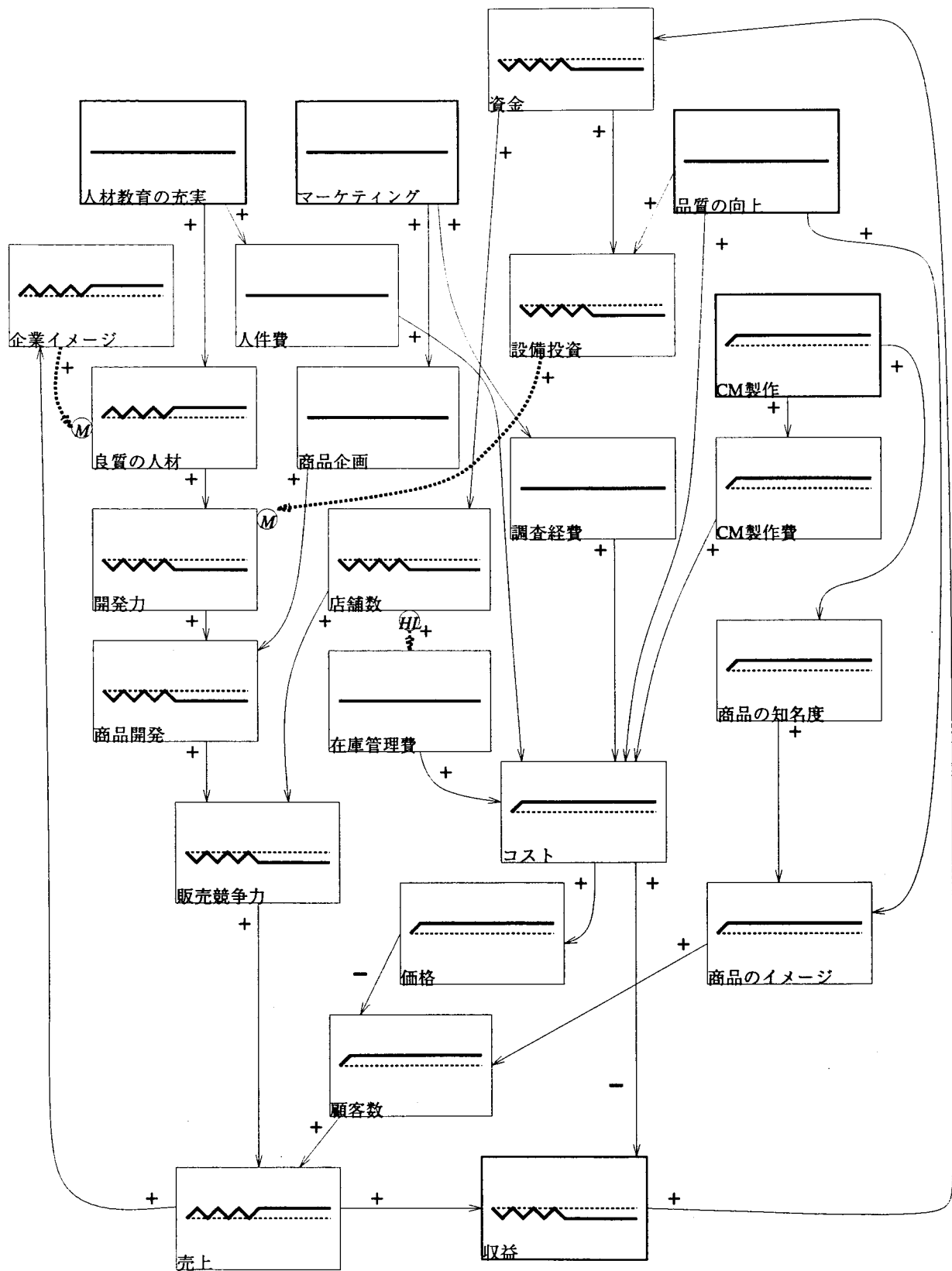


図 6.14: 実行例 4 の各ノードにおける挙動グラフ

このように、それぞれの施策案実行例に対して注目している事柄である「収益」がどのような影響のもとにどのような挙動を見せるが、出力されたシナリオを読むことで、シミュレーション結果から解析する必要がなく、直ちに知ることが可能となる。

6.5.2 検討

本手法により生成されたシナリオが、注目している事柄に対して因果の流れを明確に表現したシナリオであるか検討するため、本手法の結果出力シナリオと、任意に導出されたパスを用いて生成されたシナリオとの比較実験を行った。比較実験方法は以下の通りである。まず、被験者に構造モデルの内容について理解してもらい、実験に用いるシナリオが内容的に粗筋が通っているかどうかといった観点で選択してもらうため説明書きを提示した。その説明書きには、「この構造モデルは、営業成績の落ち込みによる利益の減少といった問題を抱えている企業を対象とし、この問題を解決するために提案された4つの施策案は収益にどのように関係してくるかを示している。…」などといったシミュレーションをする目的、各ノードの意味といったものが書かれている。次に、1つのシミュレーション結果に対し5つのシナリオ、シミュレーション結果のグラフ表示を提示し、被験者にシミュレーション結果を解析しながらシナリオを参照してもらい、シミュレーション結果と注目している事柄に対する因果の流れを表現するのに妥当と思われるシナリオを1つ選択してもらう。

比較実験に用いた5つのシナリオは以下のように生成したものである。

- (a) 最初に探索されたパスのみをシナリオ構成に用いて生成されたシナリオ (first)
- (b) $T = 1$ のとき2番目に探索されたパスと、 $T = 2$ のとき最初に探索されたパスをシナリオ構成に用いて生成されたシナリオ (second-first)
- (c) $T = 1$ のとき3番目に探索されたパスと、 $T = 2$ のときループの検出により4番目に検出されたループのパスをシナリオ構成に用いて生成されたシナリオ (third-fourthloop)
- (d) $T = 1$ のときループの検出により2番目に検出されたループのパスと、 $T = 2$ のとき最初に探索されたパスをシナリオ構成に用いて生成されたシナリオ (secondloop-first)
- (e) 本手法により選択したパスをシナリオ構成に用いて生成したシナリオ (selected-path)

5人の被験者に対して選択してもらった結果を表6.8に示す。表中の数字は選択した人数を示す。

実行例1,3と実行例4のグラフ表示を比較すると、前者は、注目ノードにおいてシミュレーション結果が大きく変動しているのに対し、後者は、注目ノードにおいてあまり変動

表 6.8: 比較実験結果

	(a) first	(b) second-first	(c) third-fourthloop	(d) secondloop-first	(e) elected-path
実行例 1	0	0	0	0	5
実行例 2	0	2	2	0	1
実行例 3	1	0	1	0	3
実行例 4	0	4	0	1	0

が見られないものであった。このことから、適切なシナリオを導くためには、施策案実施により注目している事柄が大きく影響を受けることが要求される。注目している事柄が施策案実施による影響をあまり受けない場合には、施策案実施による全体的な状況を説明するシナリオの生成、または、ユーザにとって最も納得し易いパスを特定しシナリオを生成するといった必要がある。例えば、各ノードにおけるパラメータの変化が、どの様に注目している事柄に対し影響を及ぼすかといった感度解析を行い、施策案の実施に関わらず、構造モデル中に潜在的に存在する強い因果関係を抽出し、シナリオ構成に用いるといった工夫が考えられる。

また、実行例 2 については、注目ノードにおけるシミュレーション結果が実行例 1,3 と同様に大きく変動しているが、選択率が低い。実行例 2 で生成されたシナリオを詳しく検討してみると、グラフの挙動に対し、必ずしも不適切とは考えられない。しかしながら、初期入力からの影響とループの影響が続けて 1 つの文の中に記述されており、影響源に関して文章的な曖昧さが見られた。このようなシナリオは 4 つの実行例を通して選択率が他に比べて非常に低く、より適切な表現方法を考える必要がある。

本実験に関して被験者は、「明らかにつじつまがあっていないシナリオがあり、ある程度、消去法で絞り込むことができた」といった意見が出された。また、「5 つのシナリオとも出てくる項目が同じであるので、違いを見い出しにくく選択が困難だった」といった意見も出された。そこで、実験後にそれぞれのシナリオがどのようなパスを用いて生成されたものであるか、説明したところ、本手法により選択したパスを用いて生成したシナリオに対して、「このシナリオの粗筋は間違っていない」といった意見が多く出された。このように、本手法では最適なシナリオが生成されない場合があるが、概ね正しくシミュレーション結果を示しているシナリオを生成していると考えられる。

6.6 まとめ

組織における要因間の因果関係といった本質的に定性的にしか評価できない対象における挙動把握が可能な定性シミュレーションシステムにおいて、シミュレーション結果を挙動の導出の理由となった因果の流れについて説明するシナリオを生成し提示する機能を提案した。本手法では、注目している事柄に対して、影響が強い因果関係を、各要因における挙動の類似性を基に抽出し、その因果関係を用いてシナリオを生成する。これにより、注目する事柄についての挙動、及びその挙動の原因となった因果の流れについての説明が可能となる。また、比較実験により本手法により生成されたシナリオは、施策案実施により注目している事柄が大きく影響を受ける場合には適切であることがわかった。

本方式をより有効なものにするためには、施策案の実施により状況があまり変化しないといった特殊なシミュレーション結果に対しても、適切な説明ができるシナリオを生成することが必要である。各ノードにおいて変化傾向のパラメータを動かしながら、どの程度の影響をが伝播するかを解析するなどにより、このような場合に対しても、構造モデル中に潜在的に結ばれている強い因果関係を抽出可能な手法を開発することが、残された課題の1つである。

第 7 章

結論

本研究では、情報システム開発の上流工程である情報システム計画において、クライアントサーバシステム (CSS) の提案及びシステム導入に対する効果予測を、熟練者の知識の代わりに過去に作成したシステム計画事例を活用することにより、経験の少ない初級 SE に対して教育と業務遂行を同時に行える様に、知識の提供、理解の補助という立場からの計算機支援方式を提案した。

本論文ではこれらの研究成果を 2 章から 6 章において述べた。

第 2 章では、CSS 構成設計支援システムの構想について述べた。CSS 構成設計業務では要求分析、設計案作成、設計案の分析とプロセスの繰り返しによって、顧客のシステムに対する要望をより明確にする。本方式では、このような業務の支援を行うために各プロセスの分析を行った。分析結果をもとに必要とされる機能を明確にし、CSS 構成設計支援システム CIDAC を構想した。CIDAC は、曖昧な要求を具体化する要求分析、事例ベース推論によって設計案を作成する設計案作成、出力結果の設計案を SE が理解しやすいように説明を作成する設計案分析説明作成、技術の進歩に対応するための知識を獲得する知識学習の 4 つの部分から構成する。

第 3 章では CIDAC 中の要求分析部における業務ツリー作成方式について述べた。要求仕様の一つである業務ツリーは、システム導入対象の業務内容をツリー形式で表現したものである。本方式では過去の業務ツリー作成事例を用いて新たな業務ツリーの作成を行う。SE から得られ事例の的確・不的確の情報を基に、複数の事例を結合させ業務ツリーを作成する手法を提案する。この事例の結合には、業務ツリーに含まれる階層的な因果関係を考慮するため、語句のマッチングを上位ノードから行う。複数の事例を用いることで、目的とする業務ツリーの不足箇所を減少できる。ユーザは事例中の各ノードに対して的確であるか、不的確であるかという二者選択の入力のみにより、対象システムに関する業務ツリーの作成が可能である。また、本方式により業務ツリーを作成することで、短時間での作成が可能となる。

第 4 章では CIDAC 中の設計案作成部における事例修正方式について述べた。事例修正

により設計案を構成するには、CSS 構成案を、機能とそれを実現するために必要なソフトウェアやハードウェア (以下部品と呼ぶ) とを対応できるように表現しなければならない。そこで、本方式では、CSS 構成案を各部品の機能や関係を明記できるような階層的な木構造で表現した。また、CSS を構成する際にシステム構成上の実現可能性が重要となる。実際に稼働する組合せであるがどうかという部品間の相性を考慮するために、部品の相性を記述した相性知識を用いて、設計した CSS が実際に動作するかどうかをチェックする方法を提案した。

第 5 章では CIDAC 中の設計案分析説明部における説明文作成方式について述べた。本説明機能では、要求分析過程情報または設計案作成過程情報の構造モデルをアークに付加された意味をもとに説明箇所の特定を行うことで、説明が必要となる重要箇所を特定することである。説明箇所の特定を行うことで、多量の説明文の作成を防ぎ、主要な説明文のみを提示することができる。

第 6 章では定性シミュレーション結果のシナリオ化方式について述べた。定性シミュレーションシステムにおいて出力結果は、定性値を基に描かれたグラフで与えられる。しかし、シミュレーション結果を意志決定に用いる場合、各要因における挙動を知るだけでは不十分であり、因果の流れについての分析が重要となる。このようなシミュレーション結果の検討作業を軽減する方法として、シナリオを用いて、シミュレーション結果がどのような影響をもとに出された挙動であるかを説明する機能を付け加えることを提案した。本手法では、注目している事柄に対して、影響が強い因果関係を、各要因における挙動の類似性を基に抽出し、その因果関係を用いてシナリオを生成する。これにより、注目する事柄についての挙動、及びその挙動の原因となった因果の流れについての説明が可能となる。また、比較実験により本手法により生成されたシナリオは、施策案実施により注目している事柄が大きく影響を受ける場合には適切であることがわかった。

最後に今後の課題についてまとめる。

本論文で提案した CSS 構成設計支援システムは開発の途中段階であり、課題としては本システムの完成があげられる。また、CSS 構成技術は現在も急速な進歩を遂げている。特に、イントラネットの普及が著しく、イントラネットを活用した三層 CSS の開発が主流となりつつある。支援システムの基本的な考え方、手法ではそのまま適用できると考えられるが、CSS 内のハードウェアとしてどのようなマシン用いるのかではなく、どのようなネットワーク構成を取るのかを中心に考え設計案を作成するという点で、調整が必要であると考えられる。

定性シミュレーションにおいては、構造モデルの良否が全ての結果を支配する。そのため、適切な構造モデルを容易に作成するための支援が必要である。組織における要因間の因果関係を表す構造モデルを作成するには、対象における問題の分析、関連する情報の獲得などが必要となり、計算機による情報提供などの支援が必要となる。また、構造モデル

に対する定性的情報の付与を支援する必要もある。

情報システム計画における事例の活用に対する課題としては、SE が持つ知識の管理が挙げられる。SE が持つ知識は、経験的に得られるものが多く継承されることが難しい。今後、SE が持つ知識を蓄積するために、SE に対するヘルプデスクシステムの回答を事例として活用することが考えられる。

謝辞

本研究の全過程を通じ、懇切なる御指導、御鞭撻を賜った大阪大学大学院工学研究科情報システム工学専攻 薦田憲久教授に心から感謝の意を表します。

本研究にまとめるにあたり、貴重なお時間を割いて頂き、丁寧なる御教示を賜りました大阪大学大学院工学研究科情報システム工学専攻 西尾章治郎教授、鈴木胖教授に深く感謝申し上げます。

大阪大学工学部、同大学院工学研究科において日頃より講義等を通じて研究分野の基礎的な学問を御指導を頂いた、大阪大学大学院工学研究科情報システム工学専攻の 寺田浩詔名誉教授(現 高知工科大学教授)、白川功教授、藤岡弘教授、村上孝三教授、滝根哲哉助教授、石浦菜岐佐助教授、中前幸治助教授、塚本昌彦助教授、朴炳植助教授ならびに電子情報エネルギー工学専攻 岸野文郎教授に深く感謝の意を表します。

本研究の過程で適切な御助言、御指導を賜った大阪大学大学院工学研究科情報システム工学専攻 大川剛直助教授に心より御礼申し上げます。

本研究を遂行するにあたり共同研究者として直接御指導頂き、有益なる御助言を頂きました大阪大学大学院工学研究科情報システム工学専攻 一階良知助手に厚く御礼申し上げます。

本研究を進める上で共同研究者として、クライアントサーバシステム構成設計に関する多くの資料を快く提供下さり、専門家としての立場から様々な助言を頂きました新日鉄情報通信システム株式会社関西支社第一事業部テクニカルサポートグループの森久博室長に厚く御礼申し上げます。

本研究を進めるにあたり、専門家としての立場から様々な助言を頂きました新日鉄情報通信システム株式会社関西支社第一事業部技術担当の竹中孝真氏に深く感謝致します。

また著者が取り組む以前に、本研究の基礎を築かれ、御指導下さいました大阪大学工学部情報システム工学科薦田研究室の卒業生である 曾我修治博士(現(株)日立製作所)、畑慎也氏(現 インターネットオペレーションズ(株))、に心から御礼申し上げます。

そして同研究室の 上山晴久氏(現 奈良先端科学技術大学院大学情報システム学専攻博士前期課程)、内藤淳氏(現 奈良先端科学技術大学院大学情報システム学専攻博士前期課程)、鄭忠氏、大西克彦氏、本田真吾氏には共同研究者として本研究の細部に至るまで昼夜を問わない熱心な討論を行い、研究の進捗の上で様々な御助力を頂きました。また同研

究室秘書の八川三保子氏、および同研究室の方々には日頃から多大な御支援を頂きました。ここに記して以上の方々に深く感謝の意を表する次第です。

最後に、暖かく励ましてくれた両親、祖母、姉、友人に深く感謝します。

参考文献

- [1] 情報処理学会編: 情報システムの計画と設計, 培風館 (1991).
- [2] 楠崎哲生, 渡辺共祥, 玉樹正人, 長谷川一行, 堀内一, 明石吉三: “戦略情報システム構築アプローチ -情報システム統合計画技法 HIPLAN-,” 日立評論, vol.71, no.2, pp.9-16(Feb. 1989)
- [3] 藪田和夫: “システム要求分析技法 C-NAP II と支援ツール,” 情報処理学会第 40 回大会予稿集, pp.1043-1050(March 1990).
- [4] T.Demarco:Structured Analysis and System Specification, Youdon, Inc. (1978).
- [5] P. Coad and E.Yourdon:Object-Oriented Analysis,Prentice-Hall (1991).
- [6] 曾我修治, 山本康文, 大川 剛直, 薦田憲久:“特徴的データ抽出によるアンケート分析支援方式,” 電気学会 C 部門論文誌, vol.115, no.11, pp.1355-1361 (Nov. 1995).
- [7] 曾我修治, 大川 剛直, 薦田憲久:“システム計画における構造モデル作成支援方式,” 電気学会 C 部門論文誌, vol.114, no.11, pp.1128-1135 (Nov. 1994).
- [8] T. Ohkawa, S. Soga, N. Kugumiya, and N. Komoda:“Case-based Support System for Problem Analysis in Business Information Systems Planning,” Studies on Infomatics and Control, vol.5, no.4, pp.361-372 (Dec. 1996).
- [9] 國友義久: 情報システムの分析・設計, 日科技連出版社 (1994).
- [10] J. N. Warfield: “Binary Matrices in System Modeling,” IEEE Trans. on Systems, Man and Cybernetics, vol. SMC-3, no. 5, pp. 441-449 (1973).
- [11] 寺野寿郎: システム工学入門, 共立出版 (1985).
- [12] 田村担之: 大規模システム, 昭晃堂 (1986).
- [13] J. W. Forrester:Industrial Dynamics, MIT Press (1961).
- [14] 吉田茂: 経営シミュレーション, オーム社 (1988).

- [15] 大成幹彦: シミュレーション工学, オーム社 (1993).
- [16] C. Eden: "Cognitive mapping," *European J. of Operational Research*, vol. 36, pp. 1-13 (Sept. 1988).
- [17] 谷明良: "認知構造図 (cognitive map)," 計測自動制御学会講習会「問題発掘と問題解決の諸方法」, pp. 63-78 (1980).
- [18] R. Axelrod: *Structure of Decision*, Princeton University Press (1976).
- [19] W. R. Zhang, S. S. Chen, and J. C. Bezdek: "Pool2: A Generic System for Cognitive Map Development and Decision Analysis," *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-19, no. 1, pp. 31-39 (Jan. 1989).
- [20] W. R. Zhang, S. S. Chen, W. Whenhua, and R. S. King: "A Cognitive-Map-Based Approach to the Coordination of Distributed Cooperative Agents," *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-22, no. 1, pp. 103-114 (Jan. 1992).
- [21] D. W. Conrath, A. R. Montazemi, and C. A. Higgins: "Evaluating Information in Ill-Structured Decision Environments," *J. of the Operational Research Society*, vol. 38, no. 5, pp. 375-385 (1987).
- [22] A. R. Montazemi and D. W. Conrath: "The Use of Cognitive Mapping for Information Requirements Analysis," *MIS Quarterly*, vol. 10, pp. 45-56 (Mar. 1986).
- [23] B. J. Kuipers: "Qualitative Simulation," *Artificial Intelligence*, vol. 29, pp. 289-338 (1986).
- [24] 西田豊明: "定性推論の考え方とその知的問題解決への応用", *情報処理*, vol. 32, no. 2, pp. 105-117 (Feb. 1991).
- [25] 西田豊明: 定性推論の諸相, 朝倉書店 (1993).
- [26] B. Faltings and P. Struss: *Recent Advances in Qualitative Physics*, MIT Press, Massachusetts, USA (1992)
- [27] 志度昌宏, 佐竹三江: "C/S時代を生き抜くSE," *日経コンピュータ*, no. 382, pp.102-127 (Jan. 8 1996).
- [28] 松井一郎: "SEに問われる対話技術," *日経コンピュータ*, no. 395, pp.124-131 (Jul. 8 1996).

- [29] 野間彰: “システム開発に潜む問題点を検証,” 日経コンピュータ, no.393, pp.155-160 (Jun. 10 1996).
- [30] 篠山勲: ソフトウェア開発のプレゼンテーション技術, 日刊工業新聞社 (1992).
- [31] K. J. Hughes, R. M. Rankin, and C. T. Sennett: “Taxonomy for Requirement Analysis,” in Proc. of The First Int. Conf. on Requirement Engineering, pp.176-179 (Apr. 1994).
- [32] O. Grady: System Requirement Analysis, McGraw-Hill Inc.(1993).
- [33] 薦田憲久: 情報システム工学入門, 朝倉書店 (1994).
- [34] 大川剛直, 畑慎也, 薦田憲久: “類型挙動パターンを用いた時間スケール多重化定性シミュレーション,” 電気学会 C 部門論文誌, vol.114-C, no.11, pp.1141-1147(Nov. 1994).
- [35] T. Ohkawa and N. Komoda: “Multiple Time Scaling Qualitative Simulation Using Typical Patterns”, in Proc. of IEEE Int. Workshop on Emerging Technologies & Factory Automation (ETFA'93), pp.129-136 (Sept. 1993).
- [36] S. Hata, T. Ohkawa, and N. Komoda: “Development of Qualitative Simulation System Using Typical Patterns,” in Proc. of 1994 Int. Conf. on Systems, Man and Cybernetics, pp.2749-2754(Oct. 1994).
- [37] 畑慎也, 大川剛直, 薦田憲久: “類型挙動パターン帰着型定性シミュレーションシステムの開発,” 情報処理学会第 48 回全国大会, 7S-5, pp.251-252(1994).
- [38] 溝口理一郎: エキスパートシステム I 入門, 朝倉書店 (1993).
- [39] 大場みち子, 薦田憲久, 川嶋一宏: “知識型計画システムにおける制約指向型説明機能,” 情報処理学会論文誌, vol.31, no.11, pp.1688-1695(Nov. 1990).
- [40] 安信千津子, 橋本和広, 西松正仁, 山田弘: “知識処理を適用したシナリオジェネレータの開発,” 情報処理学会論文誌, vol.34, no.11, pp.2344-2353(Nov. 1993).
- [41] E.H.Shortliffe: Computer Based Medical Consultations: MYCIN, American Elsevier (1976).
- [42] M. R. Wick and J. R. Slagle: An Explanation Facility for Today's Expert Systems, IEEE Expert, vol.4, no.1, pp. 26-36 (1989).

- [43] A. Salek, P. G. Sorenson, J. P. Tremblay, and J. M. Punshon : “The REVIEW System: From Formal Specifications to Natural Language,” in Proc. of The First international Conference on Requirement Engineering, pp.220-229(April 1994).
- [44] 長澤一嘉, 堂坂辰: “クライアント・サーバ システムの現状と展望,” 三菱電機技報, vol.69, no.5, pp.2-6 (1995).
- [45] 吉田浩三, 藤井泰文: “クライアントサーバシステムによる企業情報システムの現状と将来展望,” 日立評論, vol.76, no.6, pp.4-10 (June 1994).
- [46] 篠原茂, 芦田元之: “クライアント/サーバシステムの基本プラットフォーム,” 沖電気研究開発, vol.60, no.3, pp.17-22 (July 1993).
- [47] B. Francis: “CLIENT/SERVER The Model for the '90s,” Datamation, pp.34-40 (Feb. 15, 1990).
- [48] D. B. Davis: “Clients, Servers And The Glass House,” Datamation, pp.72-76 (Nov. 1, 1991).
- [49] 石田厚子, 影井宏司: “ソフトウェア開発プロセスの再構築 二階層反復拡大型プロセスモデルに基づくクライアント・サーバ・システム開発標準手順の開発と適用,” 情報処理学会シンポジウム論文集 (1994).
- [50] 井上晴高, 大西達雄, 疋田定幸, 坪井俊洋: “クライアント/サーバシステムの基本コンセプト,” 沖電気研究開発, vol.60, no.3, pp.3-10 (July 1993).
- [51] 井上貞夫, 坂田賢志, 島田恵夫, 萩原昇治, 木谷徹, 塩尻浩司: “クライアント・サーバ システム構築・運用支援ミドルウェア “Dolphine”, ” 三菱電機技報, vol.69, no.5, pp.25-29 (1995).
- [52] 藤沼彰久, 角田勝, 斎藤直樹, 西本進, 沼田薫 : C/S システム構築入門, 日経 BP 出版センター (1995).
- [53] W. B. Rauch: Distributed Open Systems Engineering, Wiley Computer Publishing (1996).
- [54] 石川孝: “設計対象モデリングへの AI アプローチ,” 人工知能学会誌, vol.7, no.2, pp.237-243 (Mar. 1992).
- [55] 長沢勲: “設計エキスパートシステム,” 情報処理, vol.28, no.2, pp.187-196 (Mar. 1992).
- [56] J. McDerott: “R1: A Rule-Based Configurer of Computer System,” Artificial Intelligence, vol.19, pp.39-82 (1982).

- [57] A. Howe, P. Cohen, and J. Dixon : “Dominic: A Domain-Independent Program for Mechanical Engineering Design,” Appl. of Artif. Intell. in Engineering Problems, vol. 1, pp.289-299, Springer (1986).
- [58] D. C. Brown and B. Chandrasekaran: “Knowledge and Control for a Mechanical Design Expert System,” IEEE Computer, vol.19, no.7 pp.92-100 (1986).
- [59] S. Mittal, C. L. Dym, and M. Morjaria: “PRIDE: An Expert System for the Design of Paper Handling Systems,” IEEE Computer, vol.19, no.7, pp.102-114 (1986).
- [60] 服部雅一, 田中利一, 末田直道: “事例ベース推論による機械設計,” 人工知能学会誌, vol.7, no.4, pp.597-602 (July 1992).
- [61] 小林重信: “事例ベース推論の現状と展望,” 人工知能学会誌, vol.7, no.4, pp.559-566 (July 1992).
- [62] 吉浦裕: “計算機室のレイアウト,” 人工知能学会誌, vol.7, no.4, pp.592-596 (July 1992).
- [63] 松原仁: “推論技術の観点からみた事例に基づく推論,” 人工知能学会誌, vol.7, no.4 pp.567-575 (July 1992).
- [64] 奥田健三, 山崎勝弘: “電力系統事故時復旧支援,” 人工知能学会誌, vol.7, no.4, pp.582-586 (July 1992).
- [65] 仲谷善雄, 築山誠, 福田豊生: “事例ベース推論によるエレベータの設計支援,” 人工知能学会誌, vol.7, no.4, pp.587-591 (July 1992).
- [66] J. Kolodner: Case-Based Reasoning, Morgan Kaufmann Publishers(1993).
- [67] M. L. Maher, A. Gomez, and de Silva Garza: “Developing Case-Based Reasoning for Structural Design,” IEEE Expert Intelligent Systems, vol.11, no.3, pp.42-53 (June 1996).
- [68] K. Moy Gupta and A. R. Montazemi: “Empirical Evaluation of Retrieval in Case-Based Reasoning Systems Using Modified Cosine Matching Function,” IEEE Trans. on Systems, Man. and Cybernetics-Part A: Systems and Humans, Vol.27, No.5, pp.601-612 (Sept. 1997).
- [69] 中村孝太郎, 小林重信: “事例ベース推論の対話型モデルとその機会調整支援への適用,” 人工知能学会誌, vol.4, no.6, pp.704-713 (Nov. 1989).

- [70] M. Redmond: "Distributed Cases for Case-Base Reasoning; Facilitating Use of Multiple Cases," in Proc. of American Association for Artificial Intelligence 90, pp.304-309 (1990).
- [71] A.Hiramatsu, S.Hata, T.Ohkawa, and N.Komoda: "Scenario Generator for Qualitative Simulation System," in Proc. of 1995 IEEE Int. Conf. on Systems, Man and Cybernetics, pp.143-148(Oct. 1995).
- [72] 平松綾子, 畑慎也, 大川剛直, 薦田憲久: "挙動の類似性に着目した因果関係の抽出による定性シミュレーション結果のシナリオ化方式," 電気学会 C 部門論文誌, vol.116, no.3, pp.381-388 (Mar. 1996).
- [73] 平松綾子, 大川剛直, 薦田憲久: "因果関係に基づく定性シミュレーション結果のシナリオ化," 電気関係学会関西支部連合大会, G3-33, p.G100 (Nov. 1995).
- [74] A.Hiramatsu, Y.Ikkai, T.Ohkawa, and N.Komoda; "Requirement Analysis for Client Server System Design based on Case base," in Proc. of the 15th IMACS World Congress - Scientific Computation Modelling and Applied Mathematics - vol.4, pp.31-36 (Aug. 1997).
- [75] A.Hiramatsu, Y.Ikkai, H.Morihisa, T.Ohkawa, and N.Komoda; "Requirement Analysis for Client Server System Design based on Case base," Studies in Informatic and Control, vol.6, no.3, pp.281-293 (Sept. 1997) .
- [76] 平松綾子, 一階良知, 大川剛直, 薦田憲久: "事例修正によるクライアントサーバシステム構成設計支援システムの構想," 平成 8 年電気学会 電子・情報・システム部門大会, pp.585-588(Sept. 1996).
- [77] 平松綾子, 一階良知, 大川剛直, 薦田憲久: "有意性判定によるクライアントサーバシステム設計案説明文作成方式," 電気学会 C 部門論文誌, vol.117, no.5, pp.618-624(May 1997) .
- [78] 平松綾子, 小野田仙一, 大川剛直, 薦田憲久: "事例ベースを用いたクライアントサーバシステム構成設計手法," 第 7 回 A I シンポジウム (人工知能学会研究会), pp.57-62 (Dec. 1996) .
- [79] 平松綾子, 大川剛直, 薦田憲久: "事例ベースを用いたクライアントサーバシステム構成設計向け要求分析方式," 平成 8 年電気関係学会関西支部連合大会, G3-28, p.G100 (Nov. 1996) .

- [80] Ayako Hiramatsu, Atushi Naito, Yoshitomo Ikkai, Takenao Ohkawa, and Norihisa Komoda : “Case Based Function Tree Generator for Client-Server Systems Configuration Design,” in Proc. of 1997 IEEE Int. Conf. on Systems, Man and Cybernetics (SMC’97), vol.3, pp.2069-2074(Oct. 1997).
- [81] 平松綾子, 一階良知, 森久博, 大川剛直, 薦田憲久: “クライアントサーバシステム構成設計向け事例修正方式,” 電気学会C部門論文誌, vol.118, no.4(Apr. 1998)(掲載決定).
- [82] 平松綾子, 一階良知, 森久 博, 大川剛直, 薦田憲久: “クライアントサーバシステム構成設計向け事例修正方式,” 平成9年電気関係学会関西支部連合大会, G14-3, p.G309 (Nov. 1997) .