

2010年度卒研テーマ

上智大学
情報理工学科
伊藤 潔

1

ソフトウェア工学とシステム評価

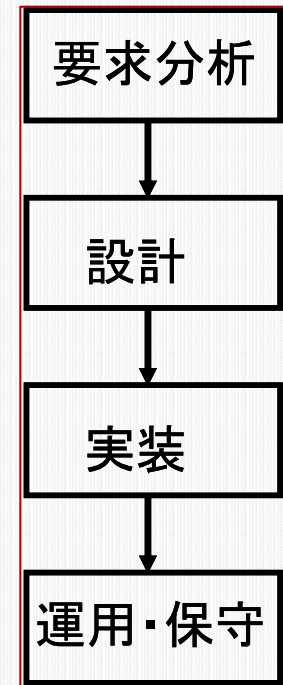
- A) Model-Based Systems Development
- B) Collaborative Timed State Transition Diagramモデルによる性能改善
- C) 委託・受託を対象とするコミットメントネットワークを導入したドメインとインタードメインのオントロジの構成と再利用
- D) 同種ダイアグラムと異種ダイアグラムの検索と相互変換による再利用法

ソフトウェア工学, ソフトウェア生産技術, SOFTWARE ENGINEERINGの分野

- 高い信頼性, 品質, 機能をもつソフトウェアを効率的に開発するための方法と, そのための環境を研究する学問分野
- 情報処理学会 (www.ipsj.or.jpより引用)の論文分野キーワード
- 開発技術
 - 要求獲得・分析法, 仕様記述法, 設計法, プログラミング方法論, プロトタイピング, 部品化・再利用技術, ドメイン分析・モデリング, プログラム自動合成, **OOA/OOD**, 代数的仕様, ソフトウェアアーキテクチャなど
- テスト・保守・管理
 - テスト, デバッグ, 検証, 仕様検証, レビュー/インスペクション, 保守運用管理, メトリクス, 版管理, 構成管理, プロジェクト管理 など
- ソフトウェアプロセス
 - ソフトウェア開発モデル, 工程管理, プロセス成熟度モデル, 標準化 など
- 開発環境
 - ソフトウェア開発支援環境/CASE, 文書化支援, リポジトリ, グループウェア など
- ヒューマンファクタ
 - ヒューマンインタフェース, 要員教育, プロジェクト管理 など
- ソフトウェア品質
 - 品質管理, メトリクス, 信頼性予測など

ソフトウェア開発サイクル

- ウォーターフォール(waterfall)型の開発サイクル
- それぞれの開発段階を明確に分離
 - 仕様の誤りによる大きな後戻りをなくそうとする
- 要求分析
 - ユーザや発注者の意図を漏れなく正確に把握し、それ以降に誤りや曖昧さを伝えない
- 設計
 - 要求分析の段階で明確となった要求仕様を逸脱しない
 - ソフトウェアの内部の実現方式や実行方式などを確定
- 実装
 - 設計によって定まった方式をプログラミング言語で計算機上で稼動できるプログラムとして実現
- 保守
 - 稼動後に判明した具合の悪さや稼動環境の変化に対応するための調整



MODEL-BASED SYSTEMS DEVELOPMENT

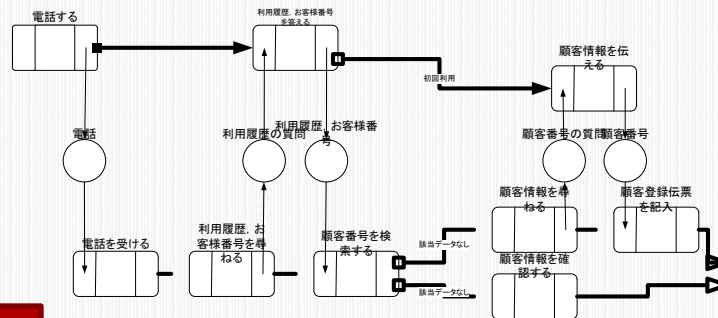
- diagram modelで開発対象のシステムをモデル化し, その要求分析と設計
- diagram modelの元になるものは, 開発対象のシステムのdomain model
 - Domain modelに基づいて, 開発対象のシステムをモデル化
- 機能プロトタイピング
 - Diagram modelは, そのまま実行可能な機能性をもつ
- 性能プロトタイピング
 - Diagram modelは, 機能的に実行させながら, 性能でデータを収集可能

MODEL-BASED SYSTEMS DEVELOPMENT

本年度は、リアルタイムシステムを組み込み型システムを対象とする

対象システムの
ドメイン
モデル

ドメインモデルに基づく **diagram model** により対象システムを要求分析・設計



コード生成

対象システムの
再利用可能
パッケージ

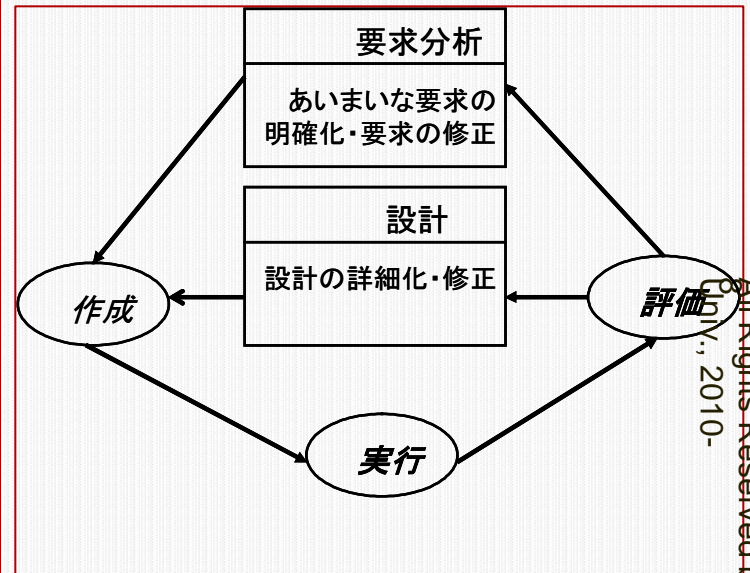
開発中に
対象システムの
functional
validationと
performance
evaluation

機能的プロトタイピング

性能的プロトタイピング

プロトタイピングサイクル

- プロトタイピングにもプロトタイプの作成・実行・評価の3フェーズからなるサイクル
- サイクルを繰り返して回りながら要求分析や設計
- プロトタイプの実行フェーズ, 各種の出力
- プロトタイプの評価では, この出力をみて,
 - 要求分析: 要求の確認, 抽象的な要求の具体化, あいまいな要求の明確化, あるいは要求の修正
 - 設計: 実現方法の確認・具体化・修正
- 対象のソフトウェアの規模があまり大きくない場合
 - ソフトウェアのライフサイクルに沿って厳格に開発を進めると, 過剰な工程管理
 - プロトタイピング手法の導入がより有効
 - 要求分析と設計の2つのサイクルを区別せずに任意に適切に切り換え

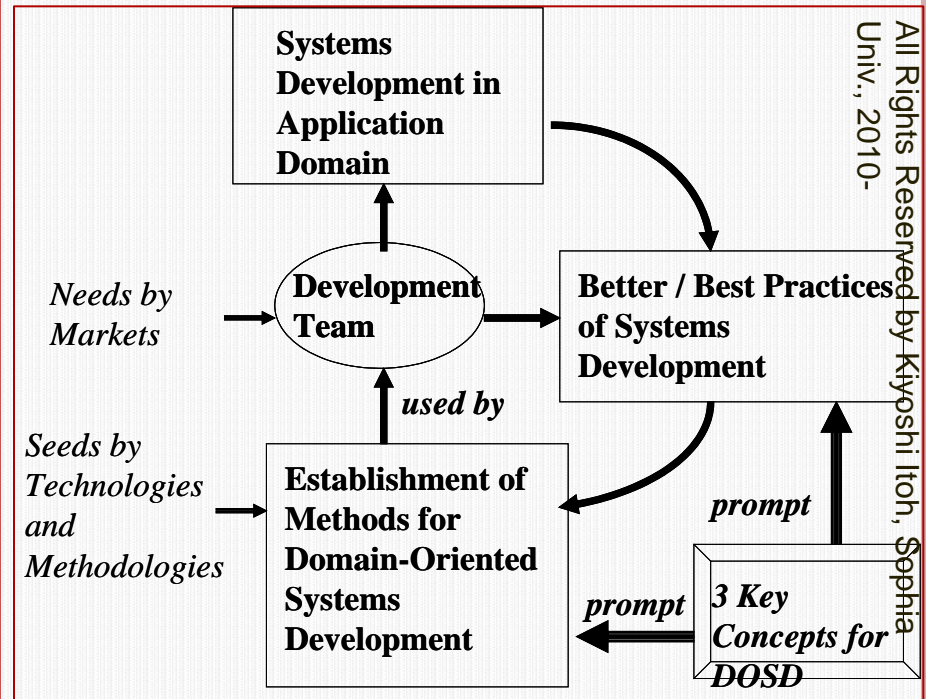


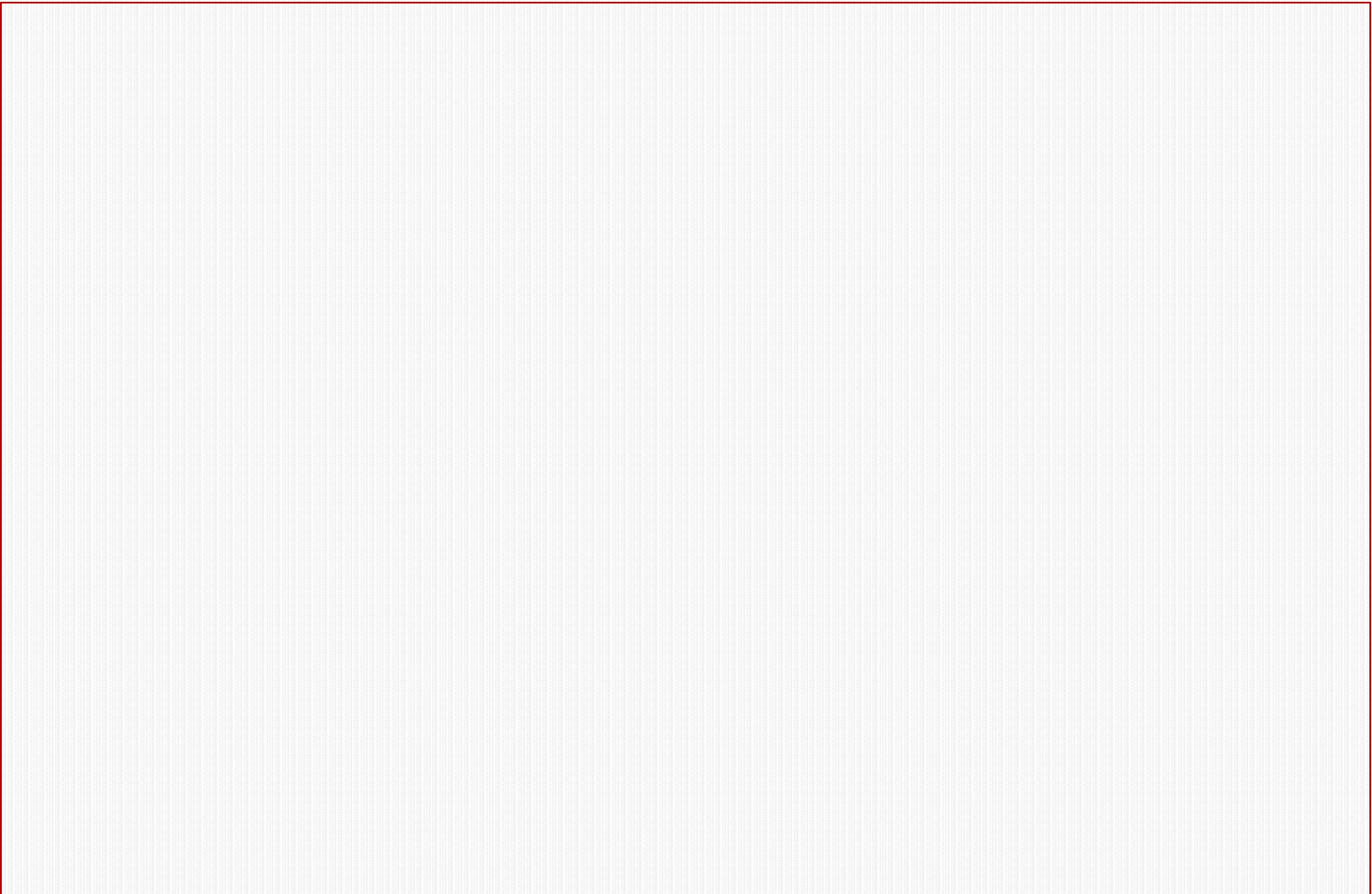
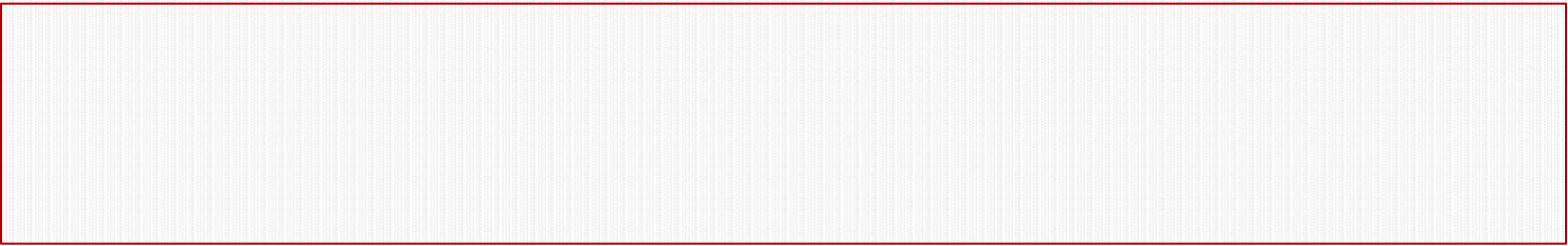
All Rights Reserved by Kiyoshi Itoh, Sophia University, 2010-

ドメイン分析： ドメイン指向システム開発

○ドメイン分析

- 対象システム自身が本来もつ各種の性質や開発上の様々な知識を十分に分析し認識して組織化
- システムの開発に有効な、共通の対象領域(ドメイン)に属する、用語、問題のとらえ方、システムの構造、システムの作り方などの、固有な概念構造を得るプロセス
- この概念構造をドメインモデル





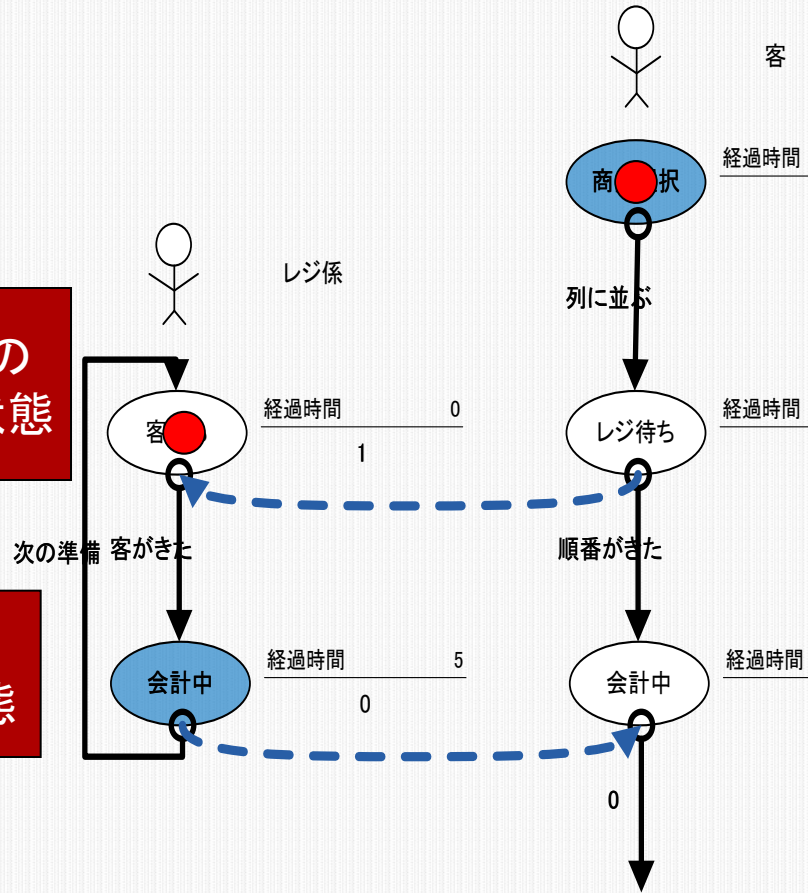
COLLABORATIVE TIMED STATE TRANSITION DIAGRAMモデルによる 性能改善エキスパートシステム

- ① 同期と時間概念を導入したState Transition Diagramでモデル化, シミュレーション
- ② 応答時間や待ち行列長によって, 過負荷のボトルネックサーバ(トランザクション)を検出
- ③ 定性推論によって, ボトルネック原因を究明
- ④ 事例ベース推論によりボトルネック改善策を提示

COLLABORATIVE TIMED STATE TRANSITION DIAGRAMモデル

客待ちの時間は、お客の
状態に依存→受動的状態

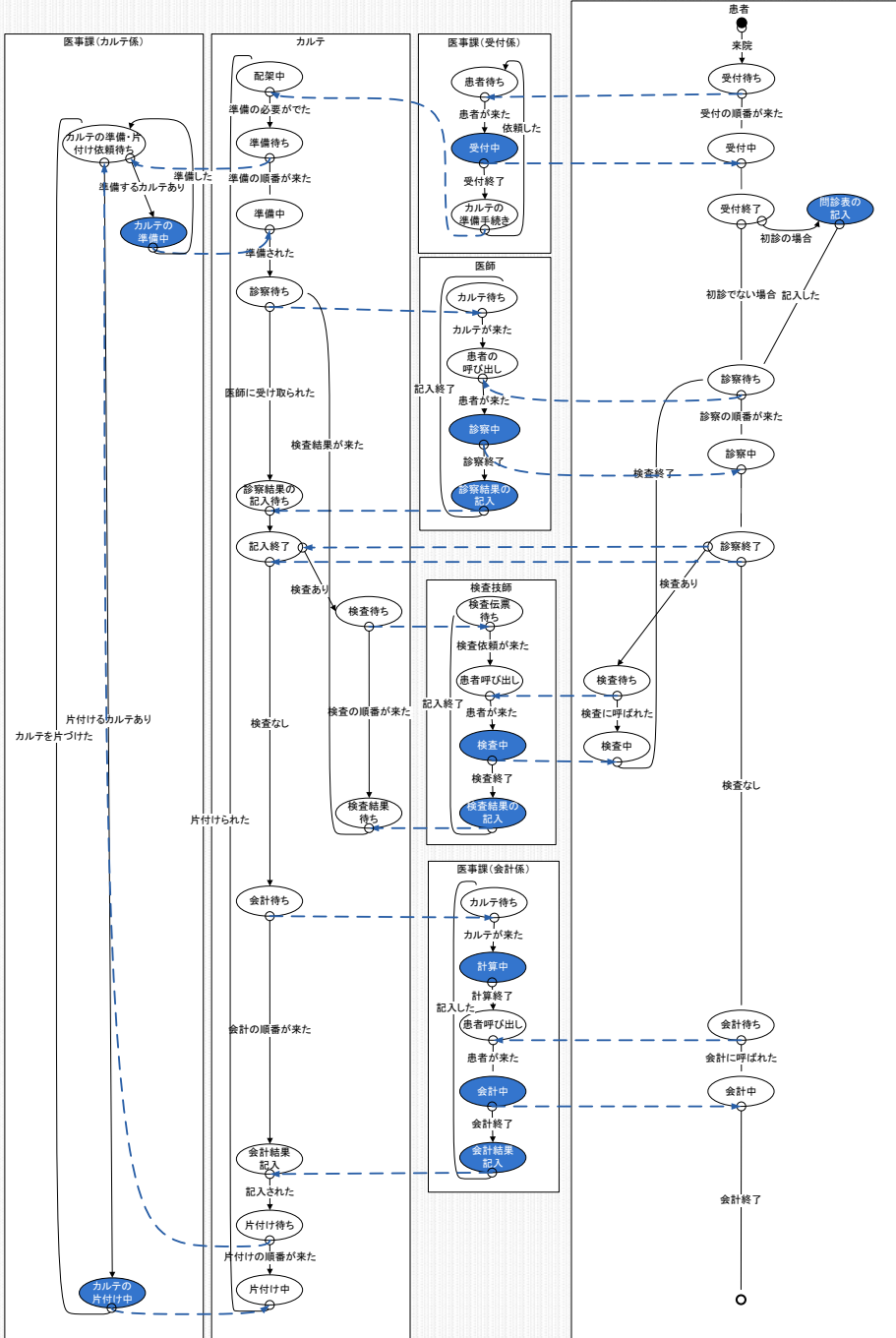
会計にかかる時間は、
店員次第→能動的状態



商品の選択にかかる時間
はお客次第→能動的状態

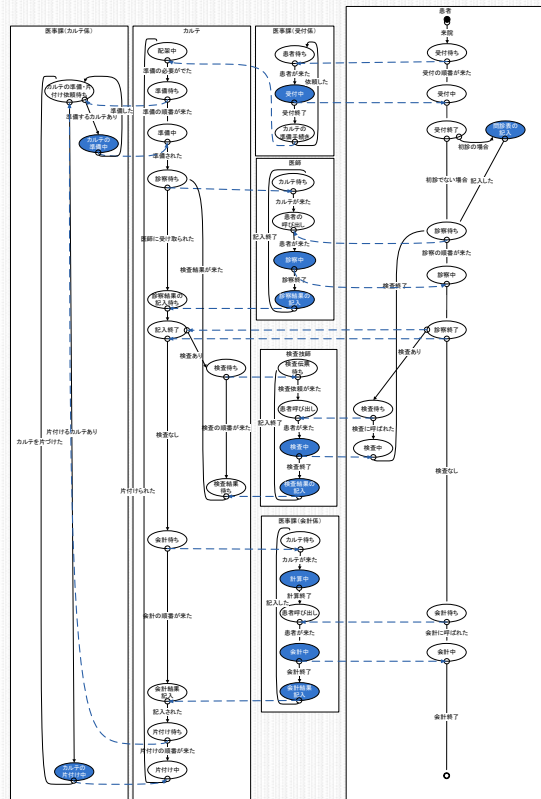
レジ待ちの時間は、店員の
状態に依存→受動的状態

会計にかかる時間は、
店員に依存→受動的状態



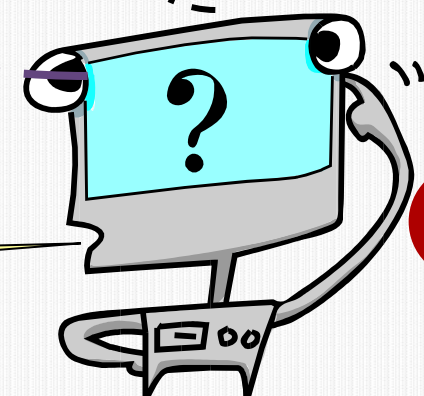
- ① 応答時間や待ち行列長によって、過負荷のボトルネックサーバ(トランザクション)を検出
- ② 定性推論によって、ボトルネック原因を究明
- ③ 事例ベース推論によりボトルネック改善策を提示

COLLABORATIVE TIMED STATE TRANSITION DIAGRAMモデルによる性能改善エキスパートシステム



Context-Name	λ	μ	ρ	q_i	q_j	q_k	(TMI)q
C1	12.3	13.11	0.93822	0.68	0.243	0.763	4.32
C2	11.2	15.1	0.74172	0.654	0.391	0.123	3.28
C3	13.12	19.01	0.69016	0.84	0.492	0.958	0.659
C4	4.23	8.12	0.52094	0.165	2.546	1.723	0.751
C5	12.4	15.17	0.8174	0.653	6.093	0.789	0.186
C6	5.87	15.69	0.37412	0.98	0.345	0.893	0.341
C7	11	10	1	0.506	1.234	0.123	0.876
C8	12.56	15.13	0.83014	0.675	8.745	0.456	0.176
C9	6.23	8.88	0.70158	3.465	0.4645	0.651	0.965
C10	8.56	10	0.856	0.7643	0.342	5.345	0.782
C11	9.54	10	0.954	0.564	0.6785	0.765	0.999
C12	14.19	15.45	0.91845	0.982	0.987	0.123	0.564
C13	15.37	20.98	0.7326	0.195	3.234	3.245	0.456
C14	3.89	5.55	0.7009	0.835	0.563	0.654	0.345
C15	12.12	15.01	0.80746	1.234	0.987	0.654	1.256
C16	17.32	15.01	1	12.783	1.022	0.723	6.389
C17	12.56	15.12	0.83069	0.763	0.542	0.534	0.568
C18	14.13	15.13	0.93391	0.567	0.238	2.178	0.257
C19	9.1	10.13	0.89832	0.632	0.876	1.234	9.3456

All Rights Reserved by Ken-ichi Itoh, Sophia



EXPERT SYSTEM

性能改善案

同種ダイアグラムと異種ダイアグラムの 検索と相互変換による再利用法

- 再利用可能なダイアグラムの検索方法の開発
 - ダイアグラムの構造を用いた検索方法
 - 同種ダイアグラムだけではなく、再利用可能な異種ダイアグラムの検索も可能
- 異種ダイアグラムの変換法の開発
 - 異種ダイアグラムを目的のダイアグラムに変換する変換ルール・プログラムの開発
- ダイアグラムの記述・検索・変換・再利用を支援するツール群の開発

ダイアグラムの知識表現

- 検索・変換を行うためにコンピュータが理解しやすい形式の知識にダイアグラムを変換
- KRSA(Knowledge Representation for System Analysis)
 - 同種ダイアグラムの再利用
 - 異種ダイアグラムの相互変換 に用いる知識表現
- GSRD(General Structure Representation for Diagrams)
 - ダイアグラムの構造検索に用いる知識表現

2つの形式の知識表現に変換

ダイアグラム



ダイアグラム(知識表現)

KRSA

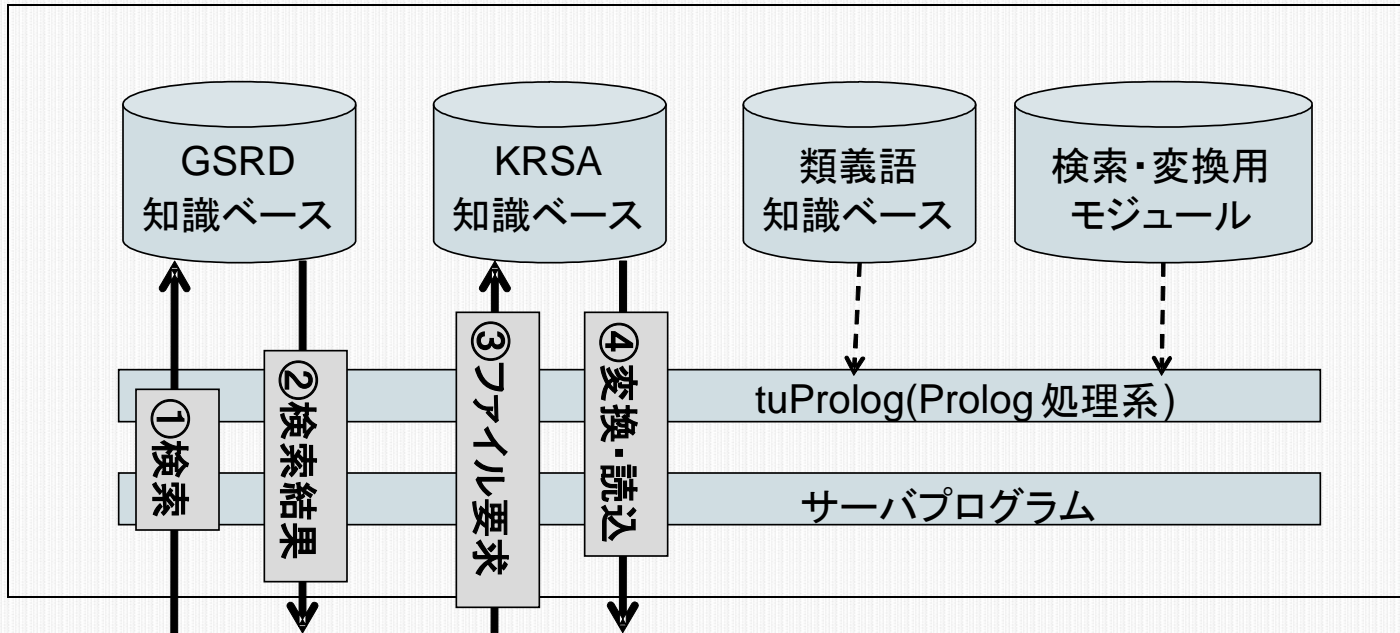
GSRD

同種ダイアグラムの再利用
異種ダイアグラムの変換

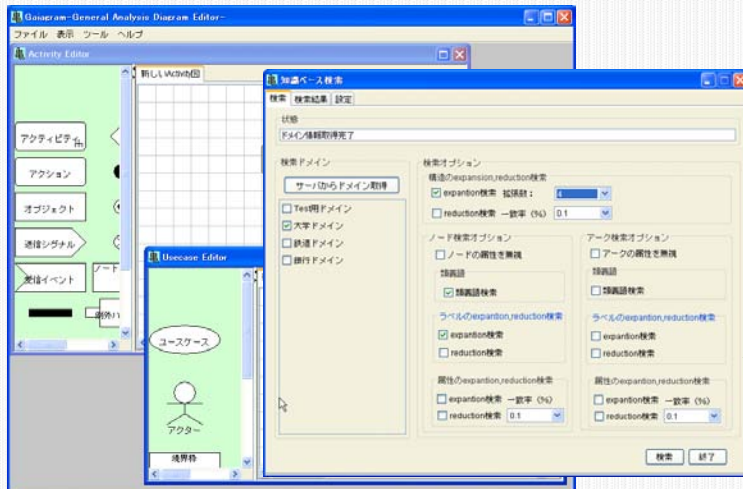
構造検索

ダイアグラム検索システムの構成

サーバサイド



クライアントサイド



ダイアグラムエディタ

記述 & 検索可能なダイアグラム

- アクティビティ図
- ユースケース図
- Petri net
- STD
- IDEF0
- DFD

MCM, DFD等を新規導入