

情報処理学会 第65回全国大会 特別トラック10

「e-Japanの進展」

「企業活動や社会生活に変革をもたらす電子政府・電子自治体」

電子自治体向けフォームベースシステムと
検索・記入・提出用ポータルサイトの構築法

中所武司

明治大学 理工学部

情報科学科

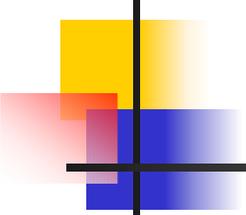
chusho@cs.meiji.ac.jp

藤原克哉

秋田大学 鉱山資源学部

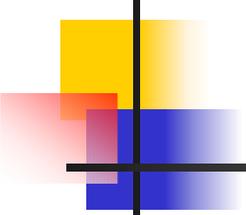
情報工学科

2003年3月27日



目次

- はじめに
 - 現状の考察
-
- 提案システム
 - フレームワーク
 - エージェント
 - モデリングツール
-
- おわりに



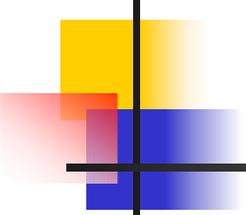
背景

- e-Japan戦略 [2001.1]

「すべての国民がITのメリットを享受できる社会」

- e-Japan重点計画 [2001.3]

「2003年度までに実質的にすべての
申請・届出等手続をオンライン化するために
個別手続のオンラインシステムの整備」



問題点

- 電子政府(中央官庁)

個別開発は無駄

「理念なしに多額の予算(1兆円弱*2年)が使われ、
電子政府は税金の無駄遣い」という批判 [2002.11]

- 電子自治体(地方自治体)

個別開発は無理

「いくつかの先進的な地域を除けば、総じて予算不足と人材不足のために、行政の情報化は遅れ気味」[情報化白書2002]

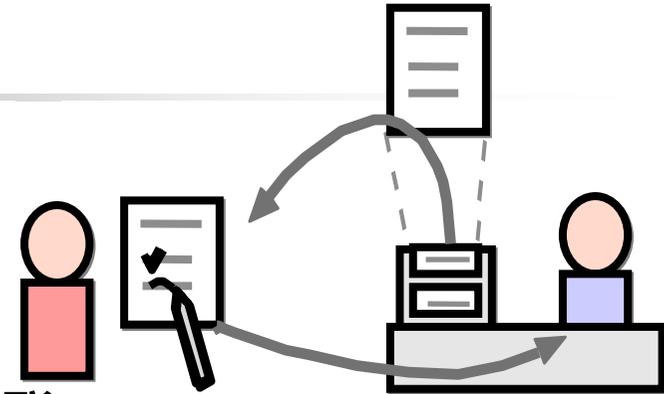
エンドユーザの立場でのシステム開発の視点が欠如

「なぜ(why)なにを(what)どのように(how)開発するか」

研究概要

■ 研究対象

日常生活の利便性向上が期待できる
電子自治体の電子申請のシステム開発



■ 論点

エンドユーザ（地域住民）の利便性

- ワンストップサービス，初心者向けユーザビリティ

エンドユーザ（業務担当者）主導の開発・保守

- 地域ごとに異なるサービス，頻繁な更新（機能変更）

システムの短期・低コスト開発と継続的保守

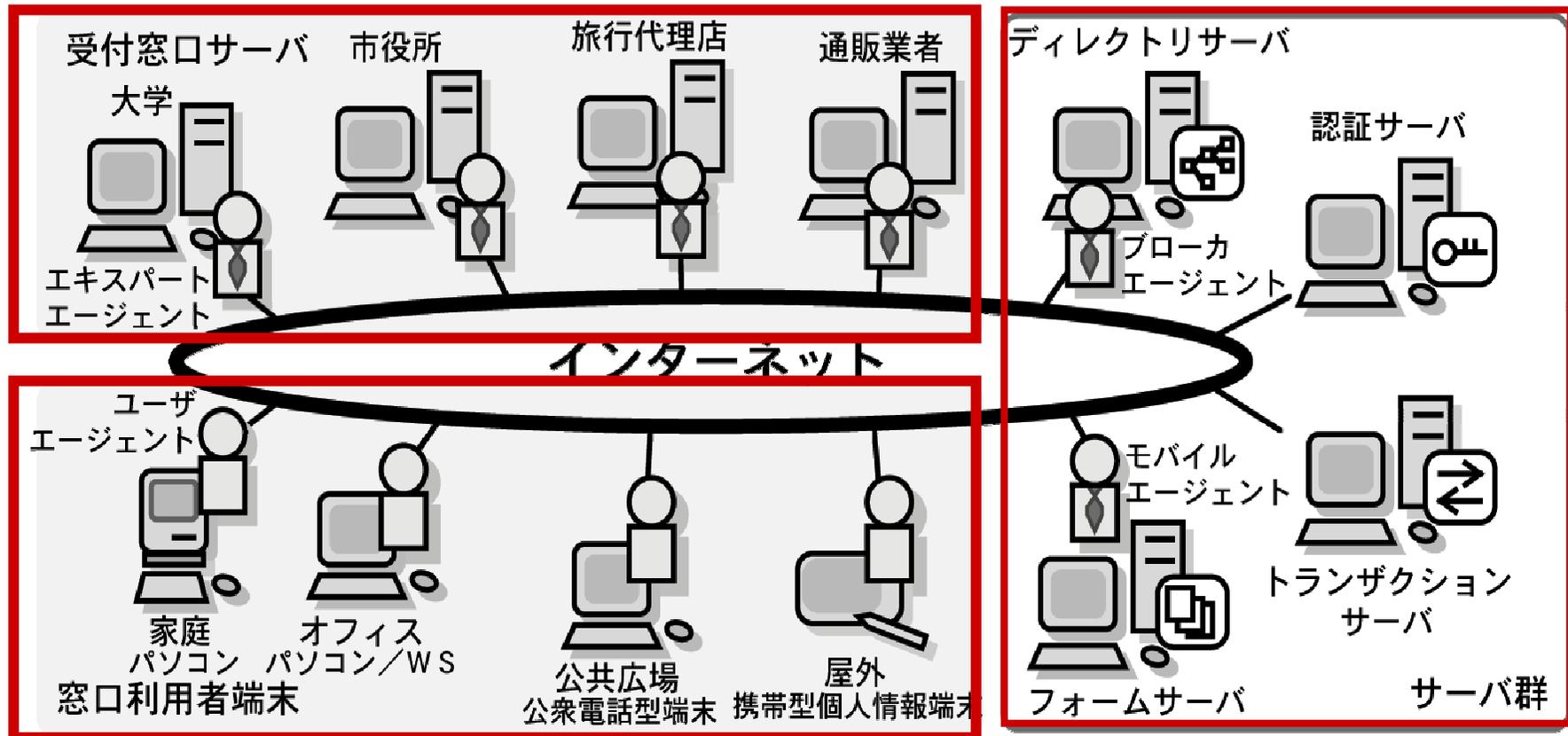
研究目標

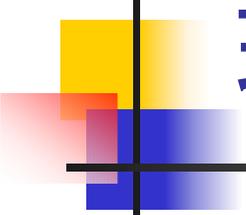
- エンドユーザ主導のアプリケーション開発技法
- 「1サービス = 1フォーム」のシステム構築
- コンポーネントベース開発 (C B S E)
 - フレームワーク技術
 - エージェント技術
 - モデリング技術



マルチエージェントに基づく フォームベースシステムの構成例

MOON (Multiagent-Oriented Office Network)





現状(1): 身近な例

** 問題点: 「IT革命??」 **

- 転居に伴う新住所、氏名、電話番号などの記入を数十回繰り返し、郵送または窓口提出。
- 某大学の教員向けの事務処理手続きに関して、各担当部署に数十種類の申請用紙が用意され、ほんの一部がインターネットから入手可能だが、記入および提出は従来通り。

現状(2):

フォームのポータルサイト

- 電子フォームのリンク集

(例) 総務省の電子政府の総合窓口

- 問題

電子フォームのダウンロードが主で、
直接電子申請できるものはまだ少ない。

(例) 川崎市の101件

- ・ 96件：PDF形式の申請書
- ・ 5件：電子申請

現状(3):

東京都23区図書館の場合

- 区役所のホームページから図書館のページまでのマウスクリック回数：平均2回弱
1回(9) 2回(7) 3回(6) 4回(1)
- 見つけやすさに差
- 検索機能：12 / 23
- 予約機能：5 / 12
- 個別開発の必要性は不明

東京都23区図書館の調査データ

クリック数

検索機能

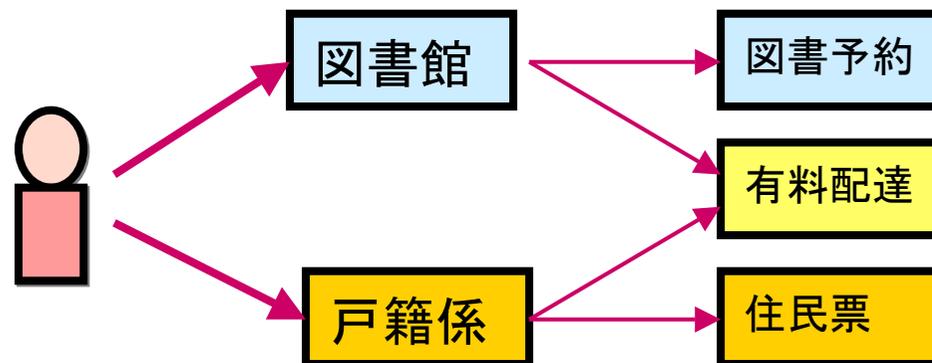
予約機能

千代田区	2	×			渋谷区	3	×		
中央区	1	○	60	登録制	中野区	4	×		
港区	3	×			杉並区	2	○	242	
新宿区	1	○	176		豊島区	1	○	38	
文京区	2	×			北区	3	×		
台東区	1	○	104	登録制	荒川区	1	○	78	登録制
墨田区	3	○	83		板橋区	2	×		
江東区	3	○	152		練馬区	1	○	129	
品川区	3	×			足立区	1	○	115	
目黒区	1	○	228	登録制	葛飾区	2	×		
大田区	2	×			江戸川区	1	○	190	登録制
世田谷区	2	×							

研究目標の例

図書予約機能に有料配達サービス機能を付加する場合

- 有料配達サービス機能の独立化
- 各種証明書の配達サービスでも利用
- サービス連携容易なシステム構成
- 業務の専門家によるサービス連携の実現

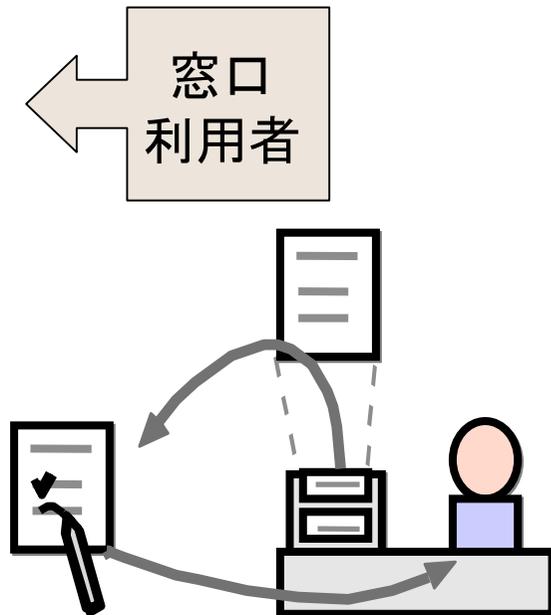


提案システム概要

簡単な形式で検索・記入・提出を実行
「どの部署に何をどのように依頼」

フォーム記入時に窓口担当者の説明を
ヘルプメッセージとして利用

名前や住所などの個人情報を自動記入

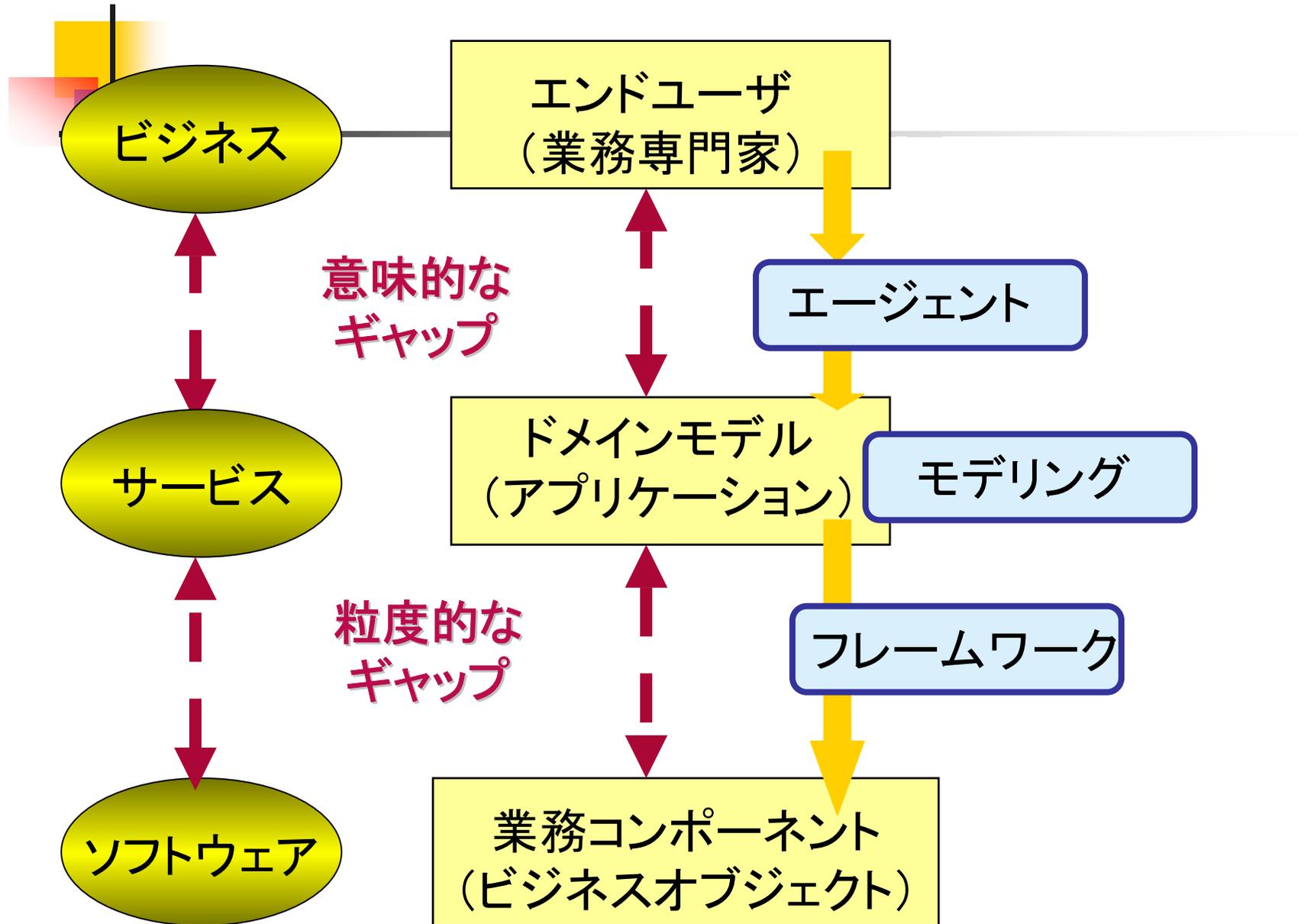


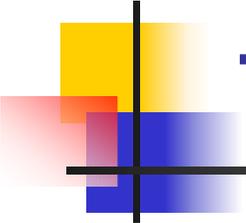
業務
担当者

ビジュアルツールを用いて専門知識を組み入れた
知的フォームを作成

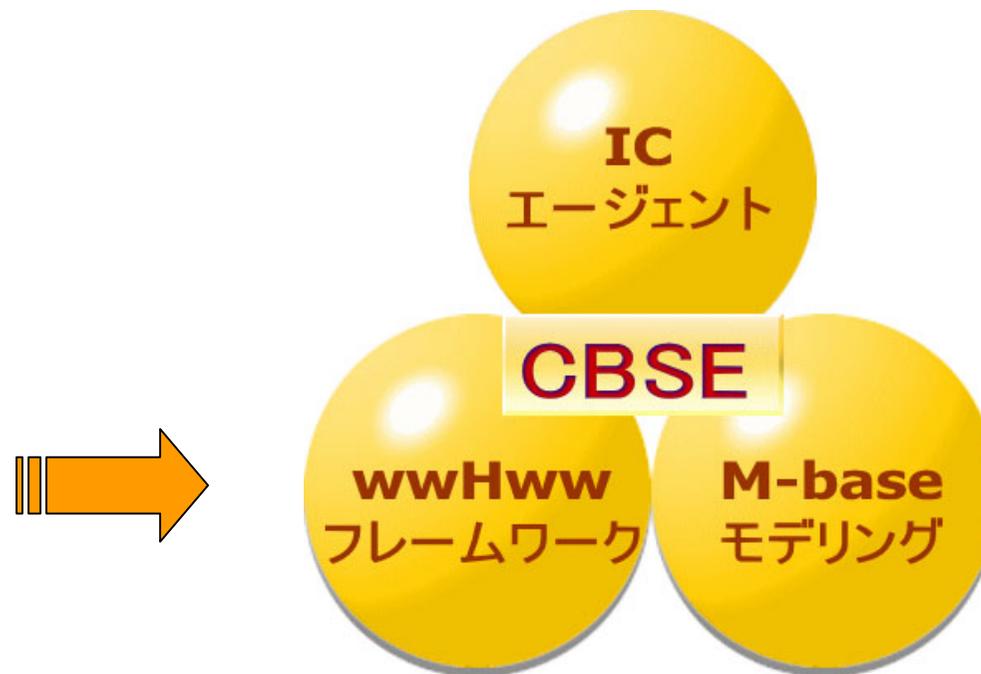
モデリングツールを用いてサービス連携を
フォーム変換ルールとビジネスルールとして定義

エンドユーザ主導のシステム構築技術





フレームワーク



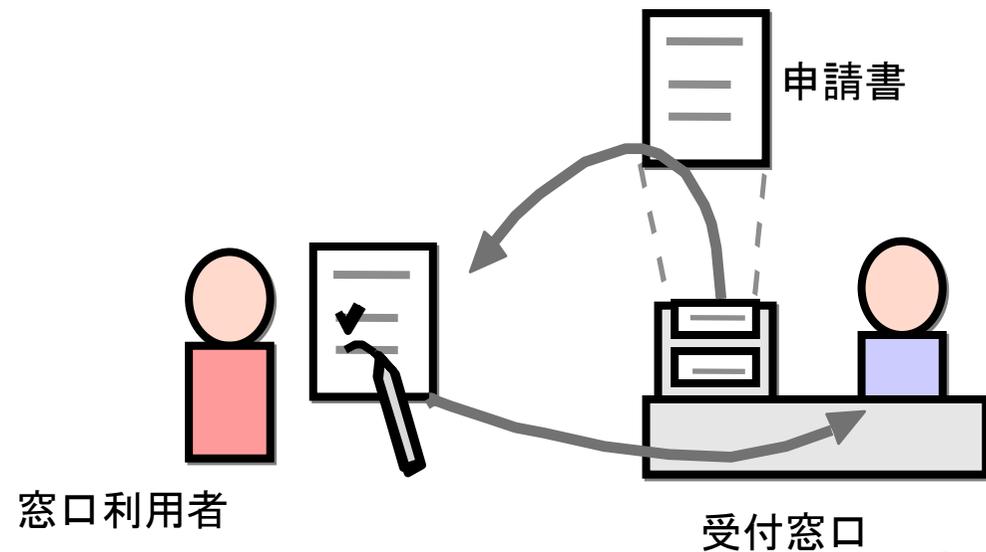
サービス授受のメタファー

「サービス授受＝窓口業務」

＜手順：「誰に何をどのように頼む」＞

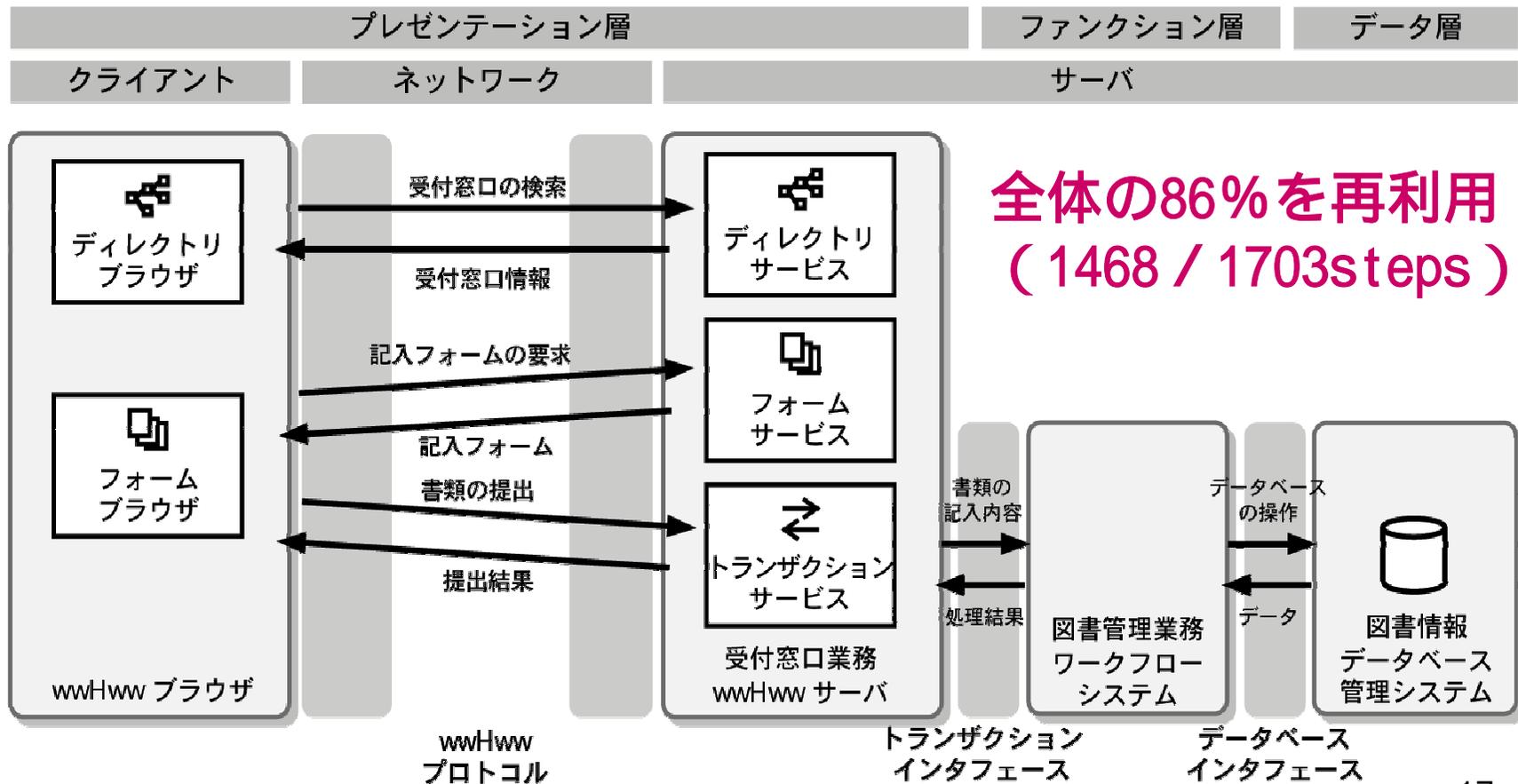
＜ Who What How ＞のインタフェース

1. 受付窓口の検索
2. フォームの取り寄せ
3. フォームへの記入
4. 受付窓口への提出



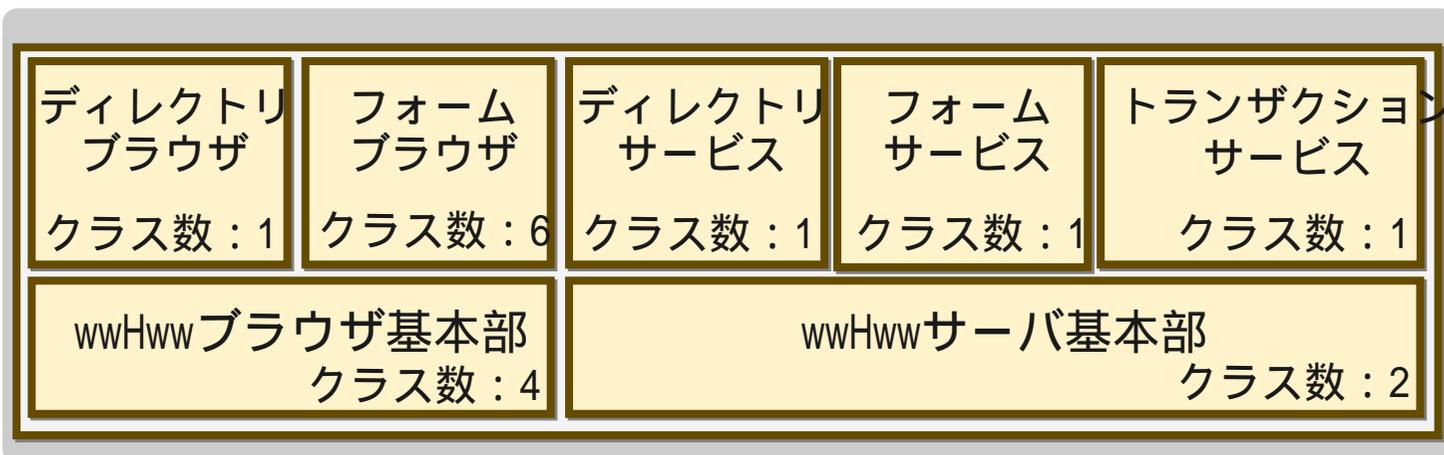
例題: フレームワークによる 図書管理システムの開発

3層アーキテクチャ & Webアプリケーション

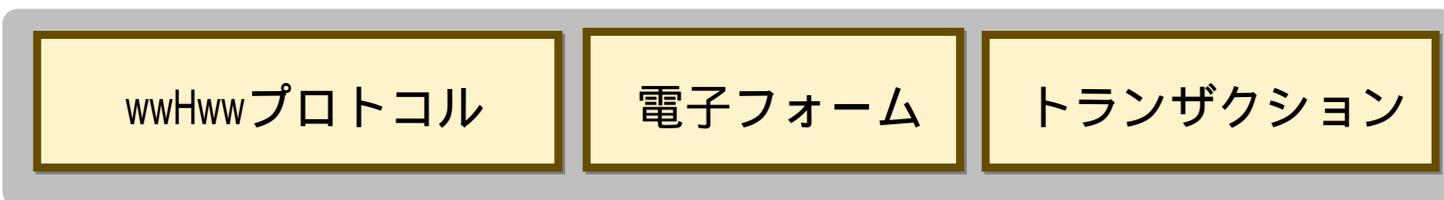


窓口業務フレームワークの構成

(特定領域向け)
窓口業務
フレームワーク



業務共通
フレームワーク



システムレベル
フレームワーク群



フレームワークのカスタマイズ方式

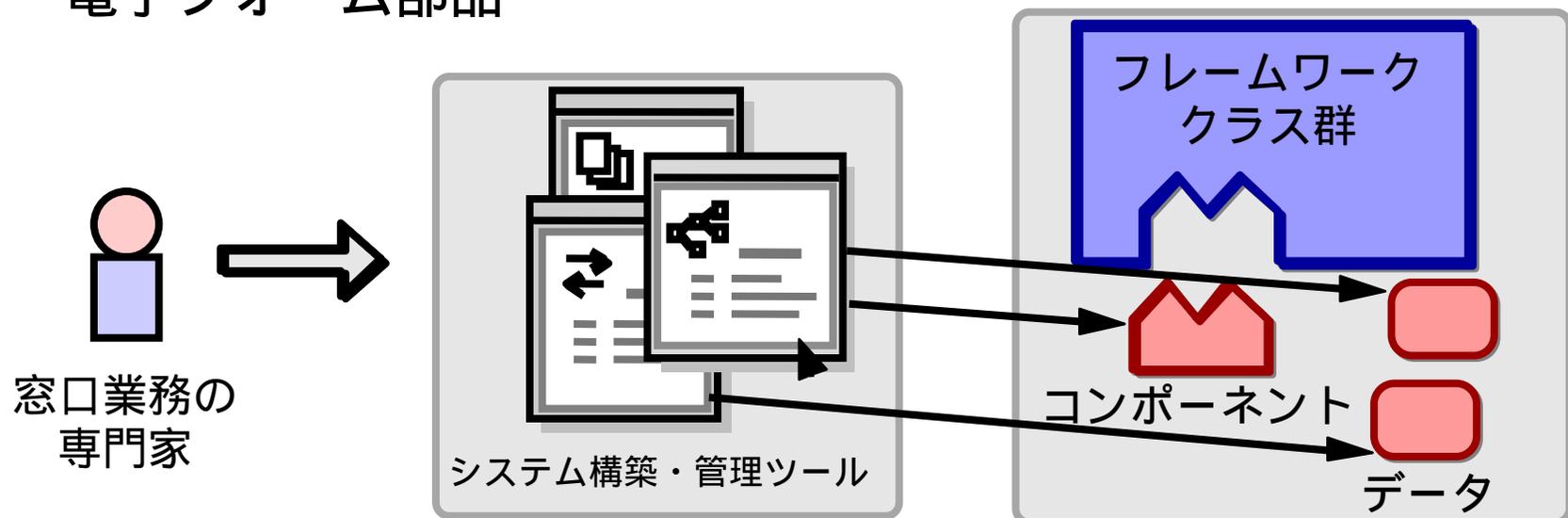
- プロパティデータ(メタデータ)の定義

【例】 窓口サービス定義、電子フォーム定義

- 差分コンポーネントのプラグイン

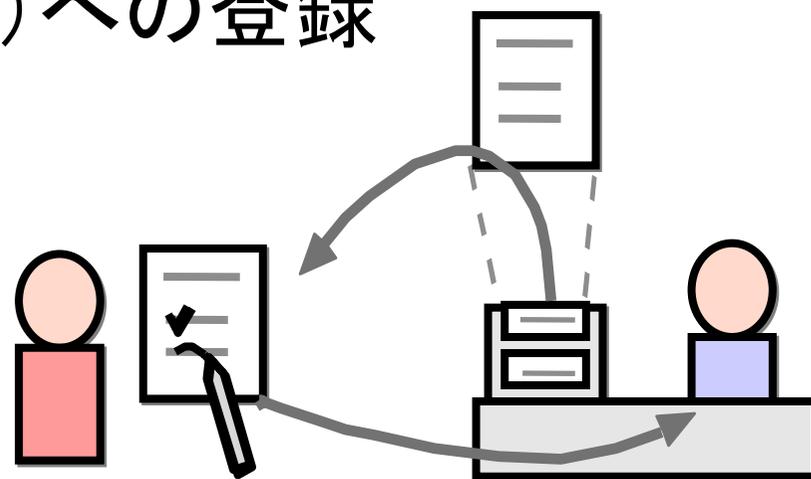
【例】 トランザクション(提出書類の処理)

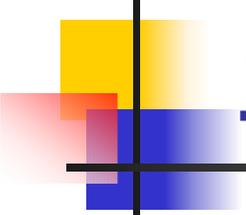
電子フォーム部品



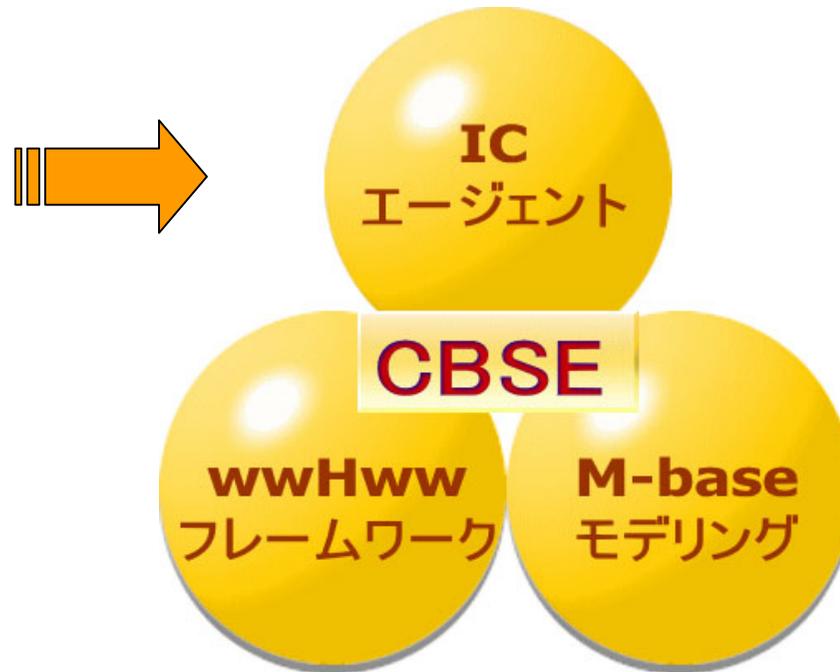
業務の専門家による サービス構築手順

1. サービスの定義
2. フォームの作成
3. 書類の処理方式の設定
4. サーバ(ディレクトリ)への登録





エージェント



マルチエージェントによる 知的ナビゲーション機能

■ 窓口検索支援

● ブローカエージェント

- ◆ サービスの多様な検索

■ フォーム記入支援

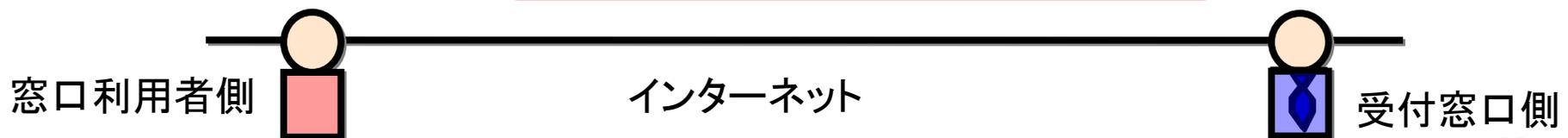
- ◆ 自動記入

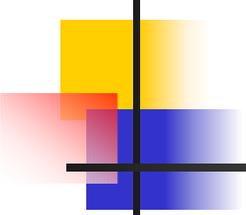
● ユーザエージェント

- ◆ 記入チェック、オンラインヘルプ

● エキスパートエージェント

● モバイルエージェント





エージェント間対話言語 FACL の特徴

FACL: form-based Agent Communication Language

- 「1サービス = 1フォーム」
- XMLによる窓口サービスのメタデータ定義
- XMLによるフォームナビゲーション定義
- XMLを用いたメッセージング方式

→ cf. UDDI, SOAP, WSDL

共通プロトコルの概要

すべての窓口への依頼をわかりやすいインタフェースで統一

Who	「依頼先」
What	「依頼項目」
How	「依頼内容」

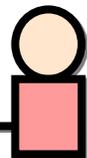
「誰に何をどのように頼む」

UDDIとの対応

White pages

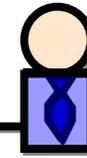
Yellow pages

Green pages



窓口利用者側

ユーザエージェント



受付窓口側

サーバエージェント

共通プロトコルの概念的UI

① 「駐車許可」を記入して検索

担当部署	<input type="text"/>
用件名	<input type="text"/>
申請書類	<input type="text"/>

②

検索結果：
「庶務課」
「駐車許可申請書」

③

フォーム取り寄せ
記入
提出

XMLによるメッセージ記述例

1. メッセージ記述 : <Message> ~ </Message>

- ▶ 依頼先(窓口名) : <who> ~ </who>
- ▶ 依頼項目(サービス名) : <what> ~ </what>
- ▶ 依頼内容 : <how rdf:resource="" /> (文書全体を指す)

<takeout>

```
<meta>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:w="http://wwhww.org/1.0/">
    <w:Message rdf:about="">
      <w:who>/明治大学/理工学部/情報科学科/ソフトウェア工学研究室/図書管理
</w:who>
      <w:what>図書貸出</w:what>
      <w:how rdf:resource="" />
    </w:Message>
  </rdf:RDF>
</meta>
<userid>fujiwara</userid><itemid>19991215A</itemid>
</takeout>
```

① 窓口検索支援：ディレクトリナビゲーション

XMLによる窓口のメタデータ定義例(1)

▶ 窓口情報定義：<Agent> ~ </Agent>

- a. 窓口の物理アドレス：rdf:about 属性 (RDF Syntax標準)
- b. 窓口の名称：<name> ~ </name>
- c. 窓口の説明文：<description> ~ </description>
- d. 検索用キーワード：<keywords> ~ </keywords>
- e. 提供するサービス定義：<service> ~ </service>

```
<Agent rdf:about="http://inside.se.cs.meiji.ac.jp/library/">  
  <name>/明治大学/情報科学科/ソフトウェア工学研究室/図書管理</name>  
  <description>ソフトウェア工学研究室の図書管理システム</description>  
  <keywords>専門書,書籍,室外持ち出し</keywords/>  
  <service><Form> ... </Form><Form> ... </Form>... </service>  
</Agent>
```

XMLによる窓口のメタデータ定義例(2)

▶ サービス(フォーム)情報定義：<Form> ~ </Form>

a. 物理アドレス：rdf:about 属性 (RDF Syntax標準)

b. サービスの名称：<name> ~ </name>

c. サービスの説明文：<description> ~ </description>

<Agent> d. 検索用キーワード：<keywords> ~ </keywords>

...

<service>

<Form rdf:about="<http://inside.se.cs.meiji.ac.jp/Library/takeout>">

<name>図書貸出</name>

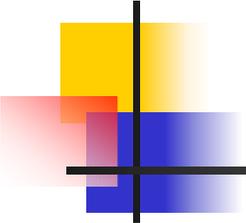
<description>図書の研究室外への持ち出しのための貸出し手続き</description>

<keywords>室外持ち出し, 帯出</keywords>

</Form>

</service>

</Agent>

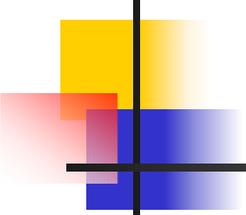


フォーム記入の問題点と解決案

- **問題点1**: 同じような記入の繰り返し
- **問題点2**: 記入の間違い、やり直し

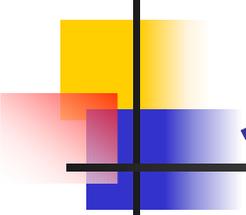
解決案：3つの記入支援方式（フォームナビゲーション）

- | | | |
|----------------|---|---------|
| (1) フォームへの自動記入 | → | 問題点1の解決 |
| (2) 記入内容チェック | → | 問題点2の解決 |
| (3) オンラインヘルプ | → | |



自動記入エージェント

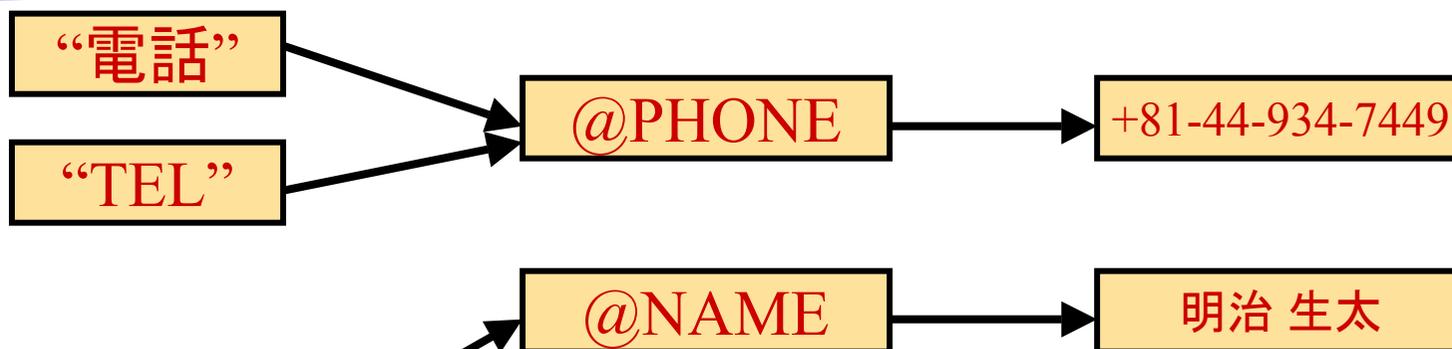
方式	特徴
事前に定義した記入ルールを利用	○記入精度はよい ×人手で定義
利用者の過去の記入内容を再利用	○name属性利用で処理が簡単 ×name属性値が不統一性で信頼性低い
入力欄の周辺の認知情報を利用	○人間の判断と同様 ×周辺情報に関する認知ルールを用意
他人の過去の記入内容を再利用	○統計的に最も多いものを入力 ×統計情報収集の仕掛け必要



入力欄周辺の認知情報利用方法

1. 「名前」, 「氏名」などの同義語を扱うため, @nameなどの**概念名**を導入
2. 上下左右の認知情報をこの概念名で置き換えたものを条件部とする**認知ルール**を導入
3. 条件部のマッチングで, 上下左右のいずれかが部分的にマッチした数を**確信度**化
4. 全ルール適用後にもっとも確信度が高いルールを採用

同義語から概念名への置き換え例



氏名	名前	申込者
ご氏名	お名前	申請者
御氏名	お名前（全角）	担当者名
本名	お名前（もしくは法人名）	ネーム

概念名の多様性

<http://www.ipsj.or.jp/mousikomi/m-nyukai.html>

正会員・学生会員・準会員入会申込書

氏名（漢 字）：
例) 情報 花子(姓名の間に全角スペース)

氏名（カ ナ）：
例) ジョウホウ ハナコ(姓名の間に全角スペース)

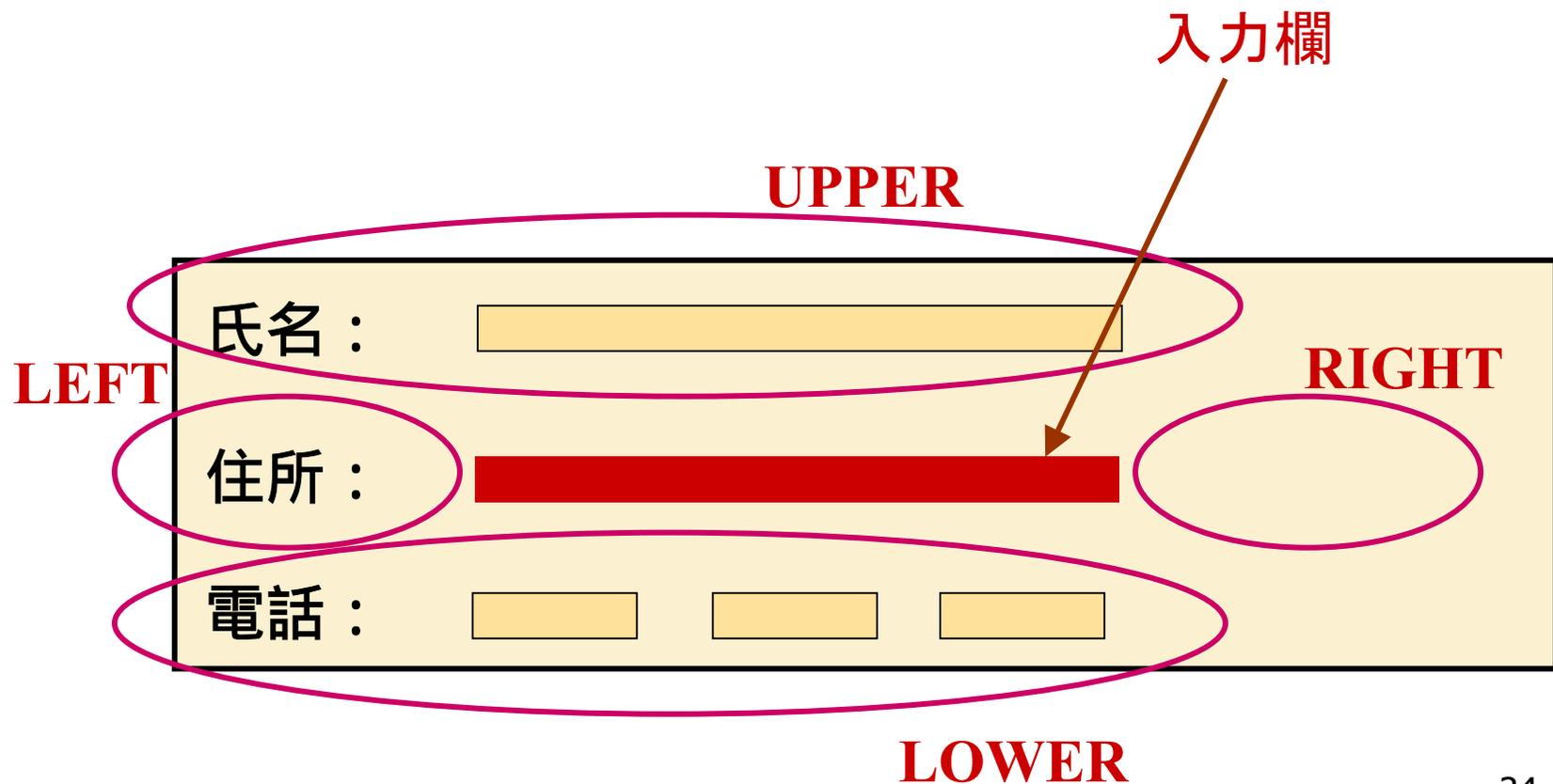
氏名（ローマ字）：
例) JOHO HANAKO(姓名の間に半角スペース)

生年月日 (yyyy/mm/dd)：

発信元 E-mail：

入力欄周辺の認知情報

◆入力欄の上下左右の情報を利用

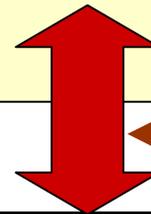


認知ルールによるマッチング

◆ 認知ルール

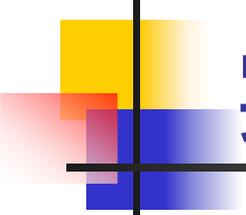
**IF (UPPER: (@address TEXTFIELD),
LEFT: @phone, RIGHT: NONE,
LOWER: (@email TEXTFIELD))
THEN @phone**

ワーキングメモリ



マッチング

<i>Attributes</i>	<i>Values</i>
UPPER	@address , TEXTFIELD
LEFT	@phone
RIGHT	
LOWER	@email , TEXTFIELD



実験結果の例

- 2番目の方法で40%であった記入成功率が240個の認知ルールを用いて74%に高まる。
- 不成功要因（認知ルール不足）：24%、
- 不成功要因（概念名置換え知識不足）：2%
- 対策：利用者が入力した正しい情報を
ユーザエージェントが対話型で学習
- 4番目の方法：ブローカエージェントが記入ルールと統計情報を管理することにより、ほぼ100%に近い記入成功率

XMLによるフォームメタデータ定義例(1)

1. フォームのメタデータ定義：<Form> ~ </Form>

- ▶ フォームの物理アドレス：rdf:about 属性
- ▶ フォームのオンラインヘルプ：<help> ~ </help>
- ▶ フォームの入力項目：<input> ~ </input>

```
<FORM action="post.cgi" method=POST> ...  
</FORM>
```

```
<Form rdf:about="http://inside.se.cs.meiji.ac.jp/library/takeout/____">  
  <name>図書貸出</name>  
  <description>図書の研究室外への持ち出しのための貸出し手続き</description>  
  <help>http://inside.se.cs.meiji.ac.jp/library/takeout/faq.html____  
</help>  
  <input>  
    <FormItem> ... </FormItem>  
    <FormItem> ... </FormItem>  
    ...  
  </input>  
</Form>
```

② フォーム記入支援：フォームナビゲーション

XMLによるフォームメタデータ定義例(2)

■ フォーム項目の定義：<FormItem> ~ </FormItem>

▶ 変数名：name 属性

◆ HTMLフォーム項目定義のNAME属性に対応

▶ 変数の型(記入チェック情報)：<datatype> ~ </datatype>

▶ 項目の値(自動記入情報)：<value> ~ </value>

▶ 項目のオンラインヘルプ：<help> ~ </help>

```
<INPUT TYPE=TEXT NAME="usr">
```

HTMLフォーム定義(一部)

```
<FormItem name="usr">
```

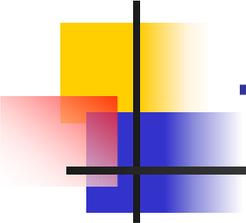
XMLによるフォームメタデータ定義(一部)

```
<datatype><Class subclassof="String" maxlength="100"/></datatype>
```

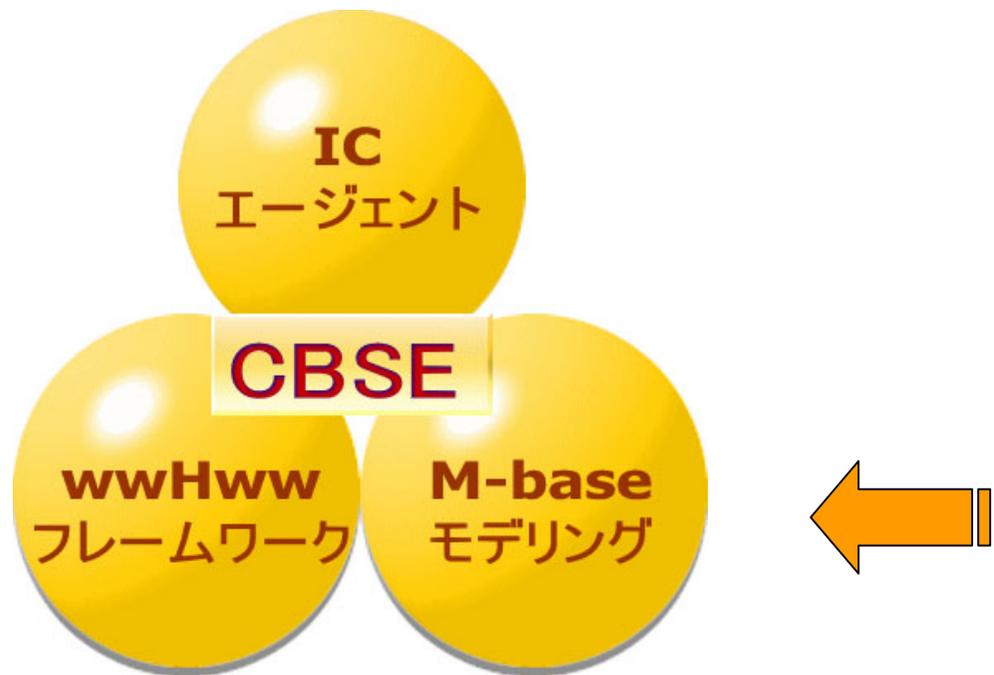
```
<value rdf:resource="urn:userprofile:@user_name_fullname"/>
```

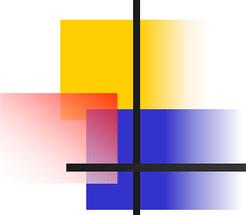
```
<help>http://inside.se.cs.meiji.ac.jp/library/takeout/help.html#usr</help>
```

```
</FormItem>
```



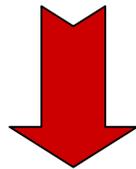
モデリングツール





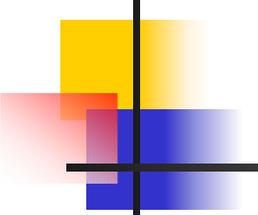
サービスの質的向上

- バックエンドシステムとの連携
- 複数のサービスの連携



エンドユーザ主導の業務のモデル化

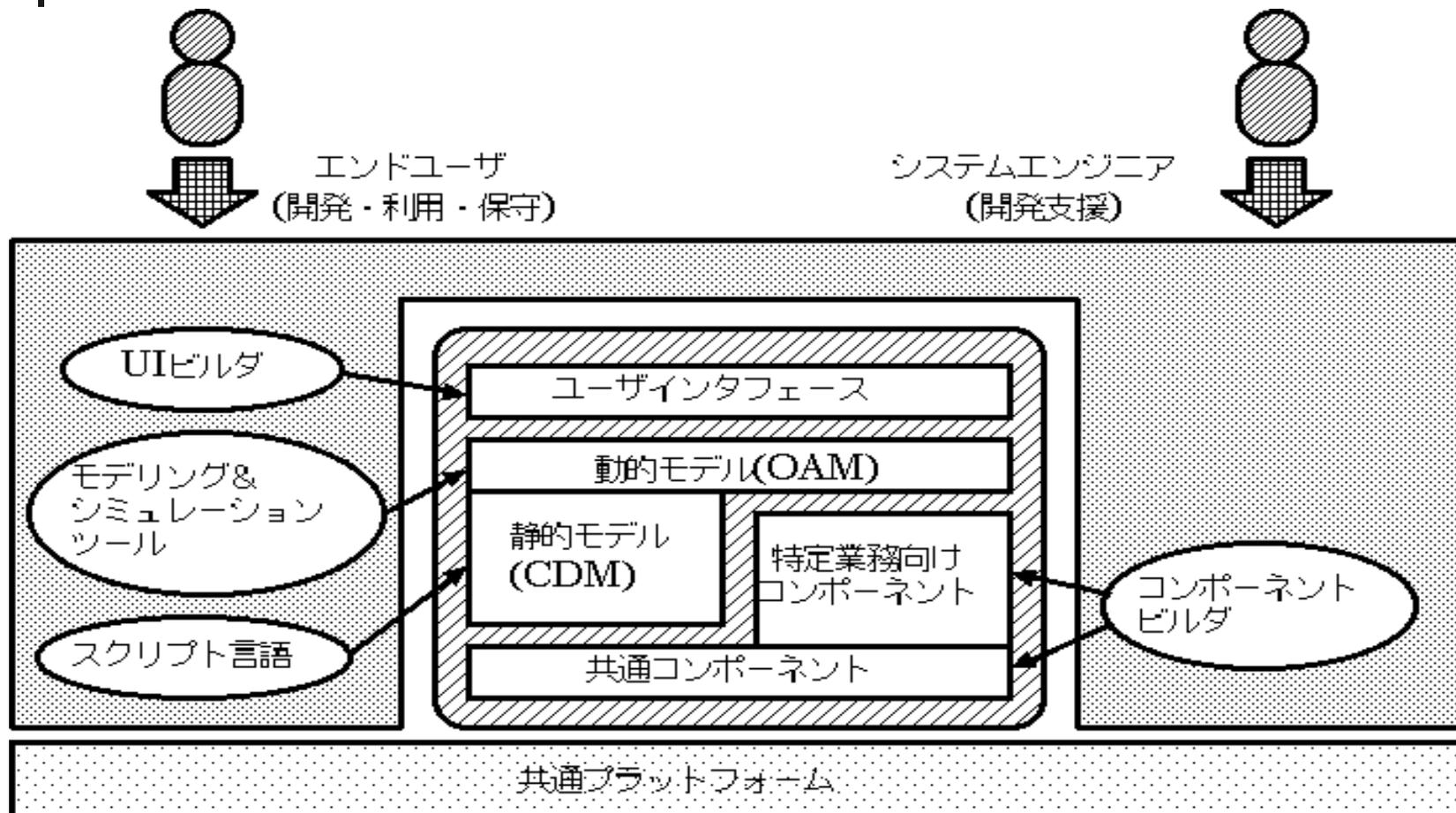
* * ビジュアルモデリングツール M-base * *

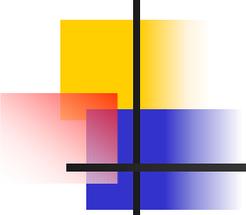


M-baseの概要

- 対象ソフトウェア
 - オフィスなどにおける小規模な業務アプリケーション
- 開発・保守形態
 - エンドユーザ主導のアプリケーション開発
 - 開発はSEの支援を受けるが保守はエンドユーザのみ
- 基本コンセプト
 - 「ドメインモデル 計算モデル」
 - 「分析 設計 プログラミング」
 - 「ソフト開発 = モデリング + シミュレーション」

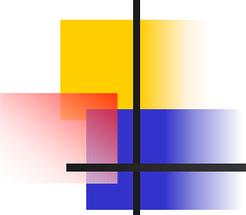
開発環境と アプリケーションアーキテクチャ





M-baseにおける エンドユーザのモデリング手順

1. **業務仕様の詳細化**
 - システム利用者とユースケースを抽出
2. **ドメインモデルの作成**
 - モデリングツールを用いて業務の動的振舞いを定義
3. **ユーザインタフェースの構築**
 - 自動生成を行い、必要に応じてカスタマイズ
4. **シミュレーション実行による検証**
 - 作成したドメインモデルの妥当性を検証



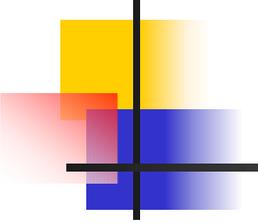
適用事例

- 例題の説明

- 情報処理学会 ソフトウェア工学研究会
要求工学ワーキンググループ

共通問題:

「国際会議のプログラム委員長の業務」



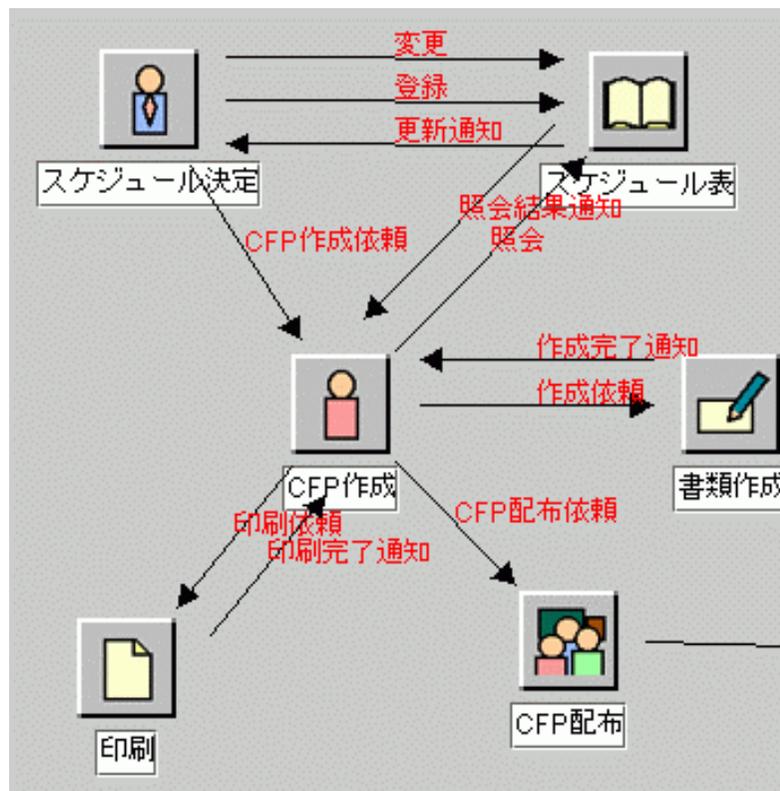
(1) 業務仕様の詳細化

— アクタとユースケースの抽出 —

- **アクター**
 - プログラム委員長
- **ユースケース**
 - スケジュール決定...(a)
 - CFPの作成・配布...(b)
 - プログラム委員選出...(c)
 - 投稿論文登録...(d), (e)

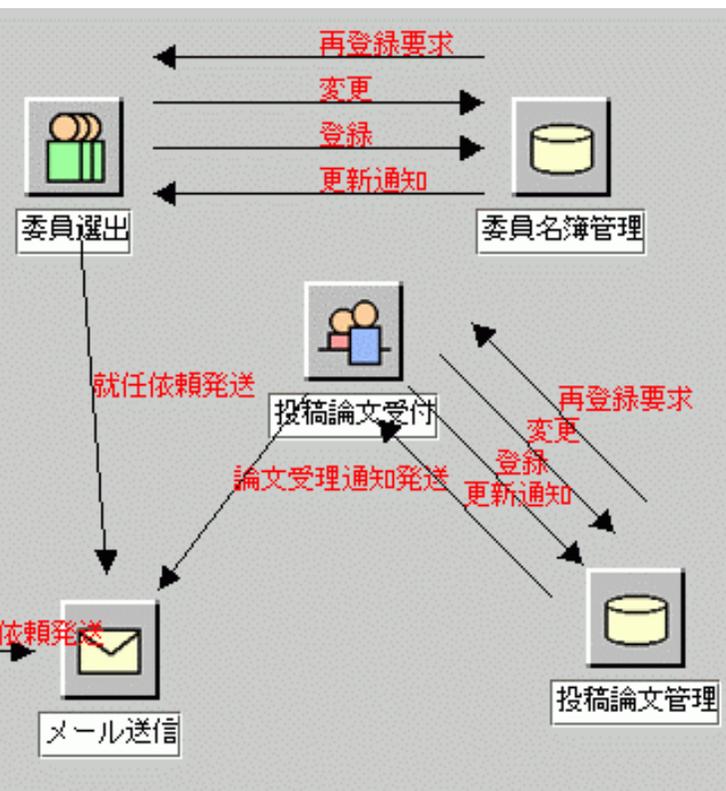
(2)ドメインモデルの構築 — 共通問題のドメインモデル —

<スケジュール決定>



<CFP作成・配布>

<プログラム委員選出>



<投稿論文登録>

(3) ユーザインタフェースの構築

■ 特徴

- UIを自動生成
- モデルとの連携処理部分も生成
- カスタマイズ可能



The screenshot shows a window titled "UIの確認" (UI Confirmation) with the following fields and controls:

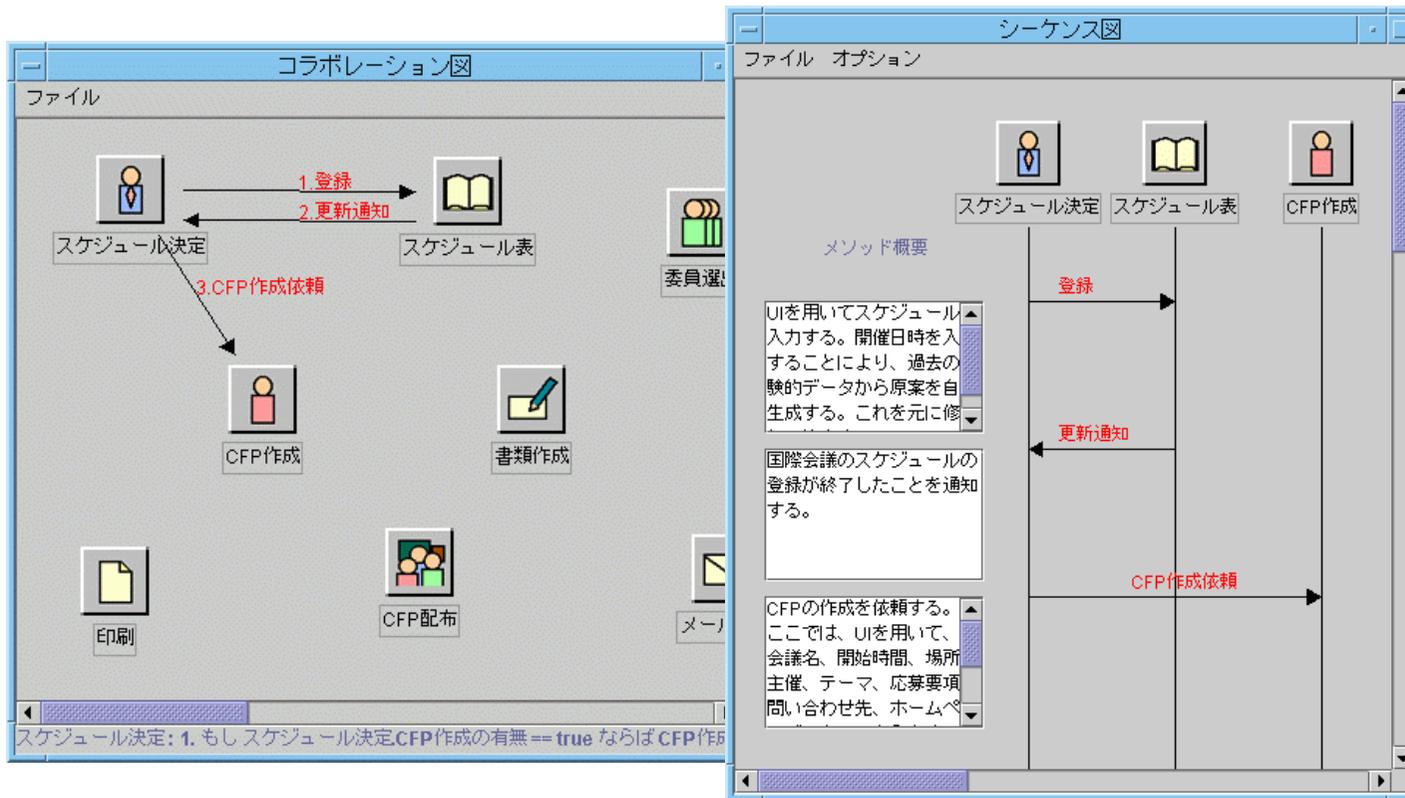
- 会議名: Text input field
- 開始時間: Time selection controls showing "0" hours and "00" minutes
- 場所: Text input field
- 主催: Text input field
- テーマ: Text input field
- 応募要項: Large text area
- 問い合わせ先: Text input field
- ホームページアドレス: Text input field with "http://" pre-filled
- 参加費: Text input field

At the bottom, there are two buttons: "了解" (Understood) and "カスタマイズ" (Customize).

<CFP作成画面>

(4) シミュレーションによる検証

■ ステップ実行によるモデルの動作確認



サービス連携

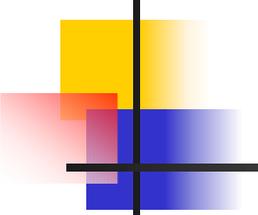
■ 複合サービスの実現

- 外部仕様: フォーム変換
- 内部仕様: 業務ロジックの実現

「モデリング + シミュレーション」

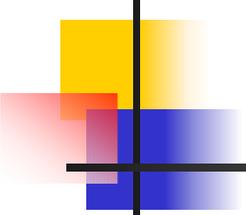
* 稼働後のサービス変更（モデル変更）への対応





おわりに

- 業務担当者主体でエンドユーザ主導型の開発を実現するための電子自治体向けフォームベースシステムの構築法
 - フレームワーク技術
 - エージェント技術
 - モデリング & シミュレーション技術
- 地域住民へのサービスの質の向上と使いやすさを実現する利用者中心のポータルサイト構築法
 - 簡明なインタフェースでの、検索・記入・提出のための〈Who, What, How〉プロトコル



参考文献

- 研究詳細

<http://www.chusho.jp/> から文献のダウンロード

「最近の論文・著作・講演リスト」