2009年2月3日

平成20年度　博士論文

User Interface Architecture with Abstract Interaction Description

抽象インタラクション記述を用いた ユーザ・インタフェース・アーキテクチャ

北海道大学　大学院情報科学研究科
コンピュータサイエンス専攻
数理計算科学講座　知能情報学研究室
柳田拓人

論文の構成

1. General Introduction
   i. User Interfaces
   ii. General Objectives
2. Interface Client/Logic Server
   i. Introduction
   ii. Logical Description Language
   iii. Interface Client/Logic Server
   iv. Implementations
   v. Discussion
   vi. Conclusion and Future Work
3. Flexible Widget Layout
   i. Introduction
   ii. Layout Problem
   iii. Formulation
   iv. Method of Layout
   v. Discussion
   vi. Conclusion and Future Work
4. General Conclusion
   i. Conclusion
   ii. Future Work

発表の流れ

1. 背景
2. 関連研究
3. 目的
4. UI アーキテクチャ （第２章）
5. GUI レイアウト自動生成 （第３章）
6. まとめ
7. 今後の課題

背景

- 技術発展に伴いインタラクティブ・サービスに用いられるプラットフォーム（デバイス）が増加

- ユーザ側からの新しい課題
  - 状況に応じた異なるUIによるサービスの利用
    - アダプティブUI（ユーザへの適応）
  - 実行中タスクのデバイス間での移動
    - UIマイグレーション

マルチ・
プラットフォーム開発

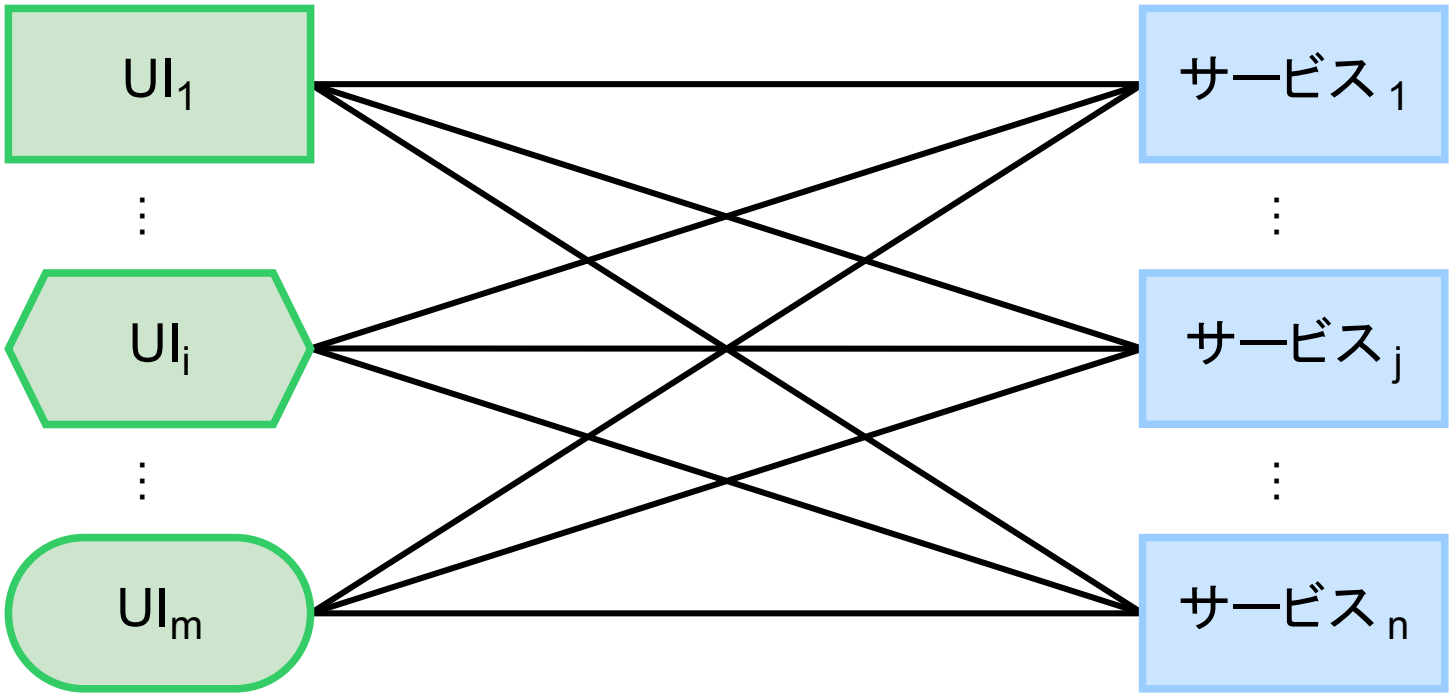
関連研究（1/8）

- マルチ・プラットフォーム開発 [34]
  - 個々のプラットフォーム毎に専用の UI
  - 同一の UI を複数のプラットフォーム共通に提供
    - ウェブ・ブラウザ，Java Swing, Tk...
  - 共通な UI 記述 + プラットフォーム毎の UI 記述
    - XML+ スタイル・シート ...
  - モデル・ベース UI 設計 [9, 10]

[9] J. Eisenstein, J. Vanderdonckt, and A. Puerta. Applying model-based techniques to the development of UIs for mobile computers. In Proc. of IUI 2001, pp. 69–76, USA, 2001.
[10] J. Vanderdonckt and F. Bodart. Encapsulating knowledge for intelligent automatic interaction objects selection. In Proc. of CHI ’93, pp. 424–429, The Netherlands, 1993.
[34] M. Florins and J. Vanderdonckt. Graceful degradation of user interfaces as a design method for multiplatform systems. In Proc. of IUI 2004, pp. 140–147, Portugal, 2004.

関連研究(2/8)

- 個々のプラットフォーム毎に専用の UI
  - ✖開発コスト，一貫性維持に問題（m·n 通りの開発）



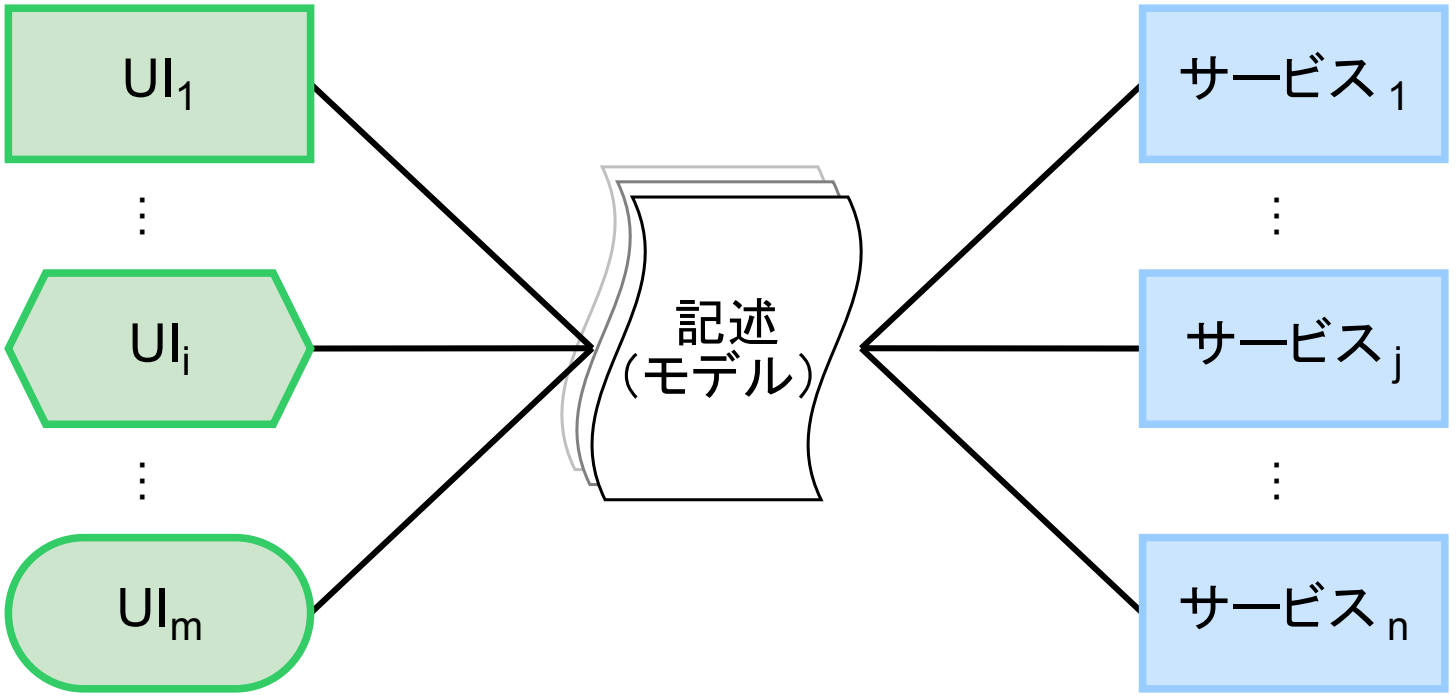
関連研究(3/8)

- モデル・ベース UI 設計
  - 高度に抽象化された記述(モデル)
    - タスク・モデル，ユーザ・モデル，インタラクタ・モデル ...
  - 共通のモデルからの多彩な UI 生成
    - ユーザへの適応を実現可能
  - Ubiquitous interactor (UBI) [13]
  - Personal universal controller [16]

[13] S. Nylander, M. Bylund, and A. Waern. Ubiquitous service access through adapted user interfaces on multiple devices. Personal and Ubiquitous Computing, 9(3):123–133, 2005.
[16] J. Nichols, B. A. Myers, K. Litwack, M. Higgins, J. Hughes, and T. K. Harris. Describing appliance user interfaces abstractly with XML. In Developing User Interfaces with XML: Advances on User Interface Description Languages, 2004.

関連研究(4/8)

モデル・ベース UI 設計



m+n 通りの開発

関連研究(5/8)

- モデル・ベース UI 設計の課題

✖UI マイグレーションなし
  - ウェブのマイグレーションは存在 [6]

✖UI の具体性の欠如
  - UI 設計者の関与が困難

} UI 記述の抽象化に伴う問題点

✖UI 生成手法が未確立
  - どのようにモデルから UI を生成するか
  - どのようにユーザ適応を実現するか

[6] R. Bandelloni and F. Paternò. Flexible interface migration. In Proc. of IUI 2004, pp. 148–155, Portugal, 2004.

関連研究(6/8)

- 「UI の具体性の欠如」への対応
  - Ubiquitous interactor (UBI) [13]
    - Customization forms によるプラットフォーム特有の表示
    - プラットフォームごとの開発が必要 ...
  - Personal universal controller [16]
    - Smart templates を追加 [17]，UI の慣習に対応
    - どのようにどれだけ定義するのかが不明 ...

[13] S. Nylander et al. 2005.
[16] J. Nichols et al. 2004.
[17] J. Nichols, B. A. Myers, and K. Litwack. Improving automatic interface generation with smart templates. In Proc. of IUI 2004, pp. 286–288, Portugal, 2004.

関連研究(7/8)

- 「UI生成手法が未確立」への対応
  - モデルからの GUI 生成 [34, 35, 36]
    - ウィジェット(GUI 部品)の選択とレイアウトの概念
    - 開発環境の提供が主な目的
    - 実行時の動的生成なし
      - ユーザへの適応が困難

[34] M. Florins and J. Vanderdonckt. Graceful degradation of user interfaces as a design method for multiplatform systems. In Proc. of IUI 2004, pp. 140–147, Portugal, 2004.
[35] M. Florins, F. M. Simarro, J. Vanderdonckt, and B. Michotte. Splitting rules for graceful degradation of user interfaces. In Proc. of AVI 2006, pp. 59–66, Italy, 2006.
[36] B. Collignon, J. Vanderdonckt, and G. Calvary. An intelligent editor for multi-presentation user interfaces. In Proc. of SAC 2008, pp. 1634–1641, Brazil, 2008.

関連研究(8/8)

- 「UI生成手法が未確立」への対応
  - ウィジェット選択 + レイアウト問題(動的 GUI 生成)
    - レイアウト・マネージャ
      - 開発者によるウィジェットの決定が必要
    - 施設配置問題(FLP) [39]
      - GUI 生成を対象とした研究はない?
    - LSI レイアウト [40]
      - スピードの観点がない → ユーザを待たせないためには スピードが重要

[39] S. K. Peer and D. K. Sharma. Human-computer interaction design with multi-goal facilities layout model. Computer and Mathematics with Applications, 56:2164–2174, 2008.
[40] H. Kitazawa. Overview of the LSI layout CAD algorithms and their applications to image processing. In Technical Report of the Institute of Electronics, Information and Communication Engineers VLD2006–38, vol. 106, pp. 25–30, 2006.

本論文の概要

- 対象とする課題
  - モデル・ベース UI 設計

1. 生成される UI の具体性向上
2. UI マイグレーションの実現

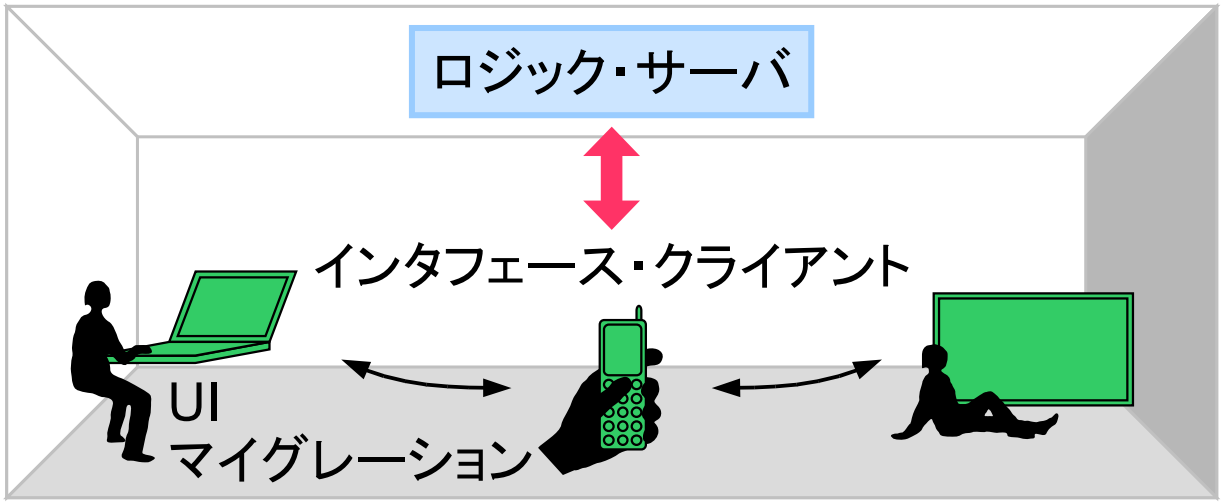
第 2 章
UI アーキテクチャ

3. モデルからの UI 生成手法
   - 特に広く用いられる GUI の生成について

第 3 章
GUI レイアウト自動生成

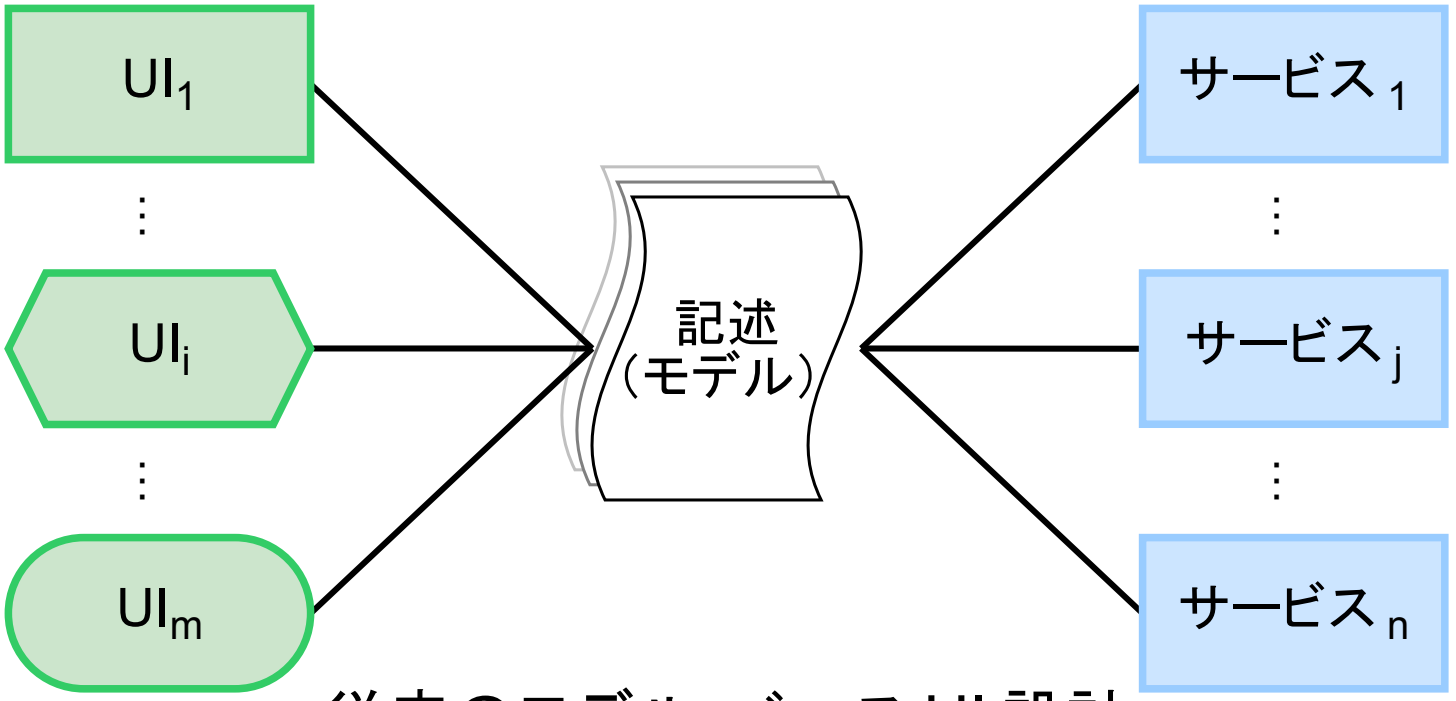
目的(1/3)

- Interface client/logic server (ICLS)
  - モデル・ベース UI アーキテクチャ
    - 具体性を伴ったアダプティブ UI
    - モデル・ベースの UI マイグレーション



目的(2/3)

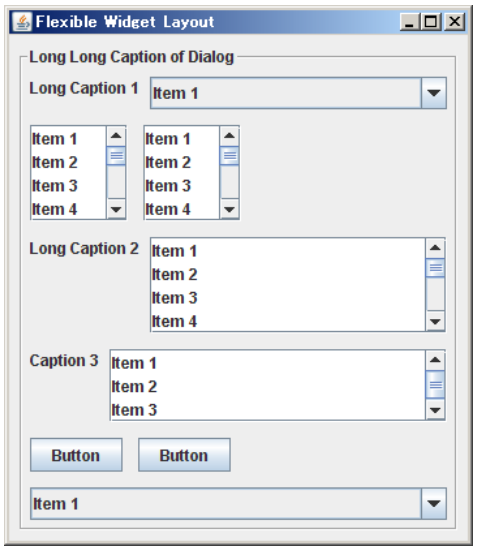
- Interface client/logic server (ICLS)
  - モデル・ベースの UI マイグレーション



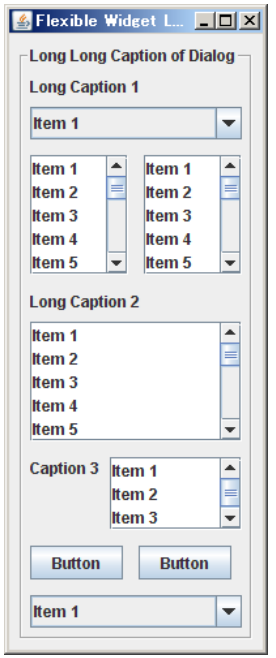
従来のモデル・ベース UI 設計

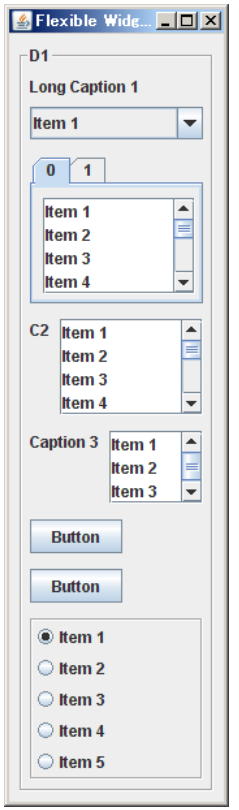
目的(3/3)

- Flexible Widget Layout (FWL)
  - UI モデルからの自動 GUI 生成
    - どのウィジェットを使用するのかの動的な決定
    - レイアウトの速やかな完了



同一の UI モデルに
対応する GUI





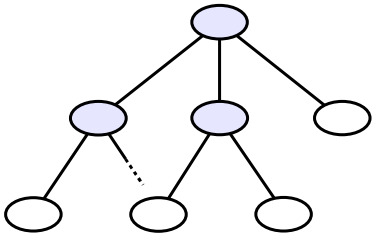
UIアーキテクチャ

- 対象とする課題
  - モデル・ベース UI 設計
    1. 生成される UI の具体性向上（の可能性）
    2. UI マイグレーションの実現
    3. モデルからの UI 生成手法
       - 特に広く使われている GUI の生成について

UIアーキテクチャの研究

論理仕様(モデル)記述(1/2)

- Abstract interaction description language (AIDL)
  - 選択行為モデルに基づいたUI機能記述
    - 選択要素(行為)
      - 選択肢のリスト，選択要素の意味 ...
      - 現在の選択状態(S)
    - グループ要素，記述要素(文字列)
  - 情報の入力を重視



... モダリティを越える情報の出力

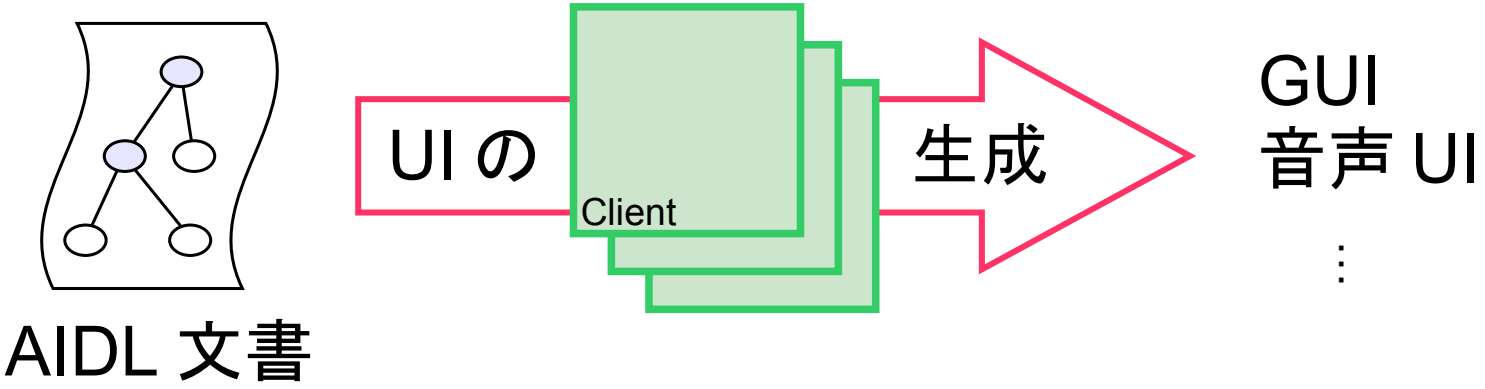
AIDL の記述能力

提案手法の対象： アプライアンスの遠隔操作

論理仕様(モデル)記述(2/2)

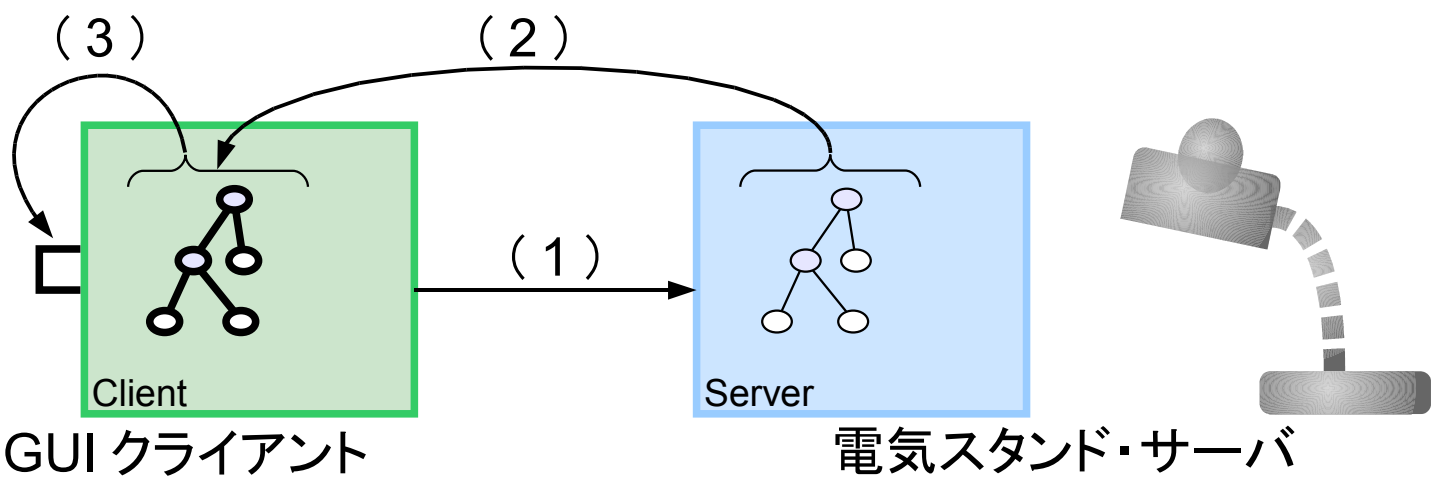
- ユーザ側における UI の生成
  - 様々なクライアントを用意すれば UI の多様化が可能
  - ユーザへの適応(カスタマイズ)が可能

開発時における UI の生成



クライアント・サーバの連携（1/2）

- 接続～ UI 生成
  1. クライアントがサーバに，セッション開始を要求
  2. クライアントはサーバから AIDL 文書を受信
  3. クライアントはモデルの木構造に基づき UI を生成



UIアーキテクチャ

- 対象とする課題
  - モデル・ベース UI 設計
    1. 生成されるUIの具体性向上(の可能性)
    2. UI マイグレーションの実現
    3. モデルからの UI 生成手法
       - 特に広く使われているGUIの生成について

UIアーキテクチャの研究

UIの生成と問題点

- 抽象的 UI 機能と具体的 UI 部品とのマッピング

UI 機能（選択要素）

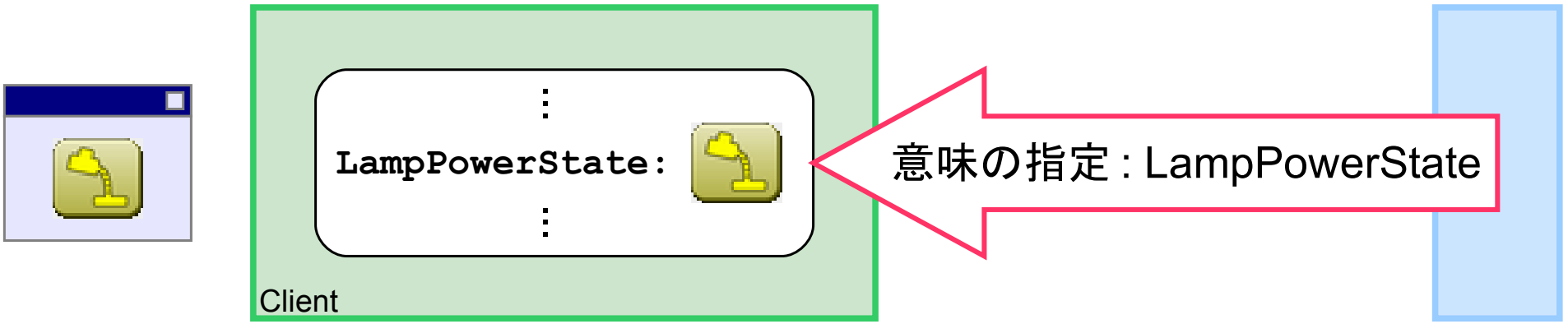
UI 部品（ウィジェット）

- UI の具体性の欠如
  - UI 設計者の関与が困難

論理仕様記述（モデル）
による UI 生成の問題点

意味の導入(1/2)

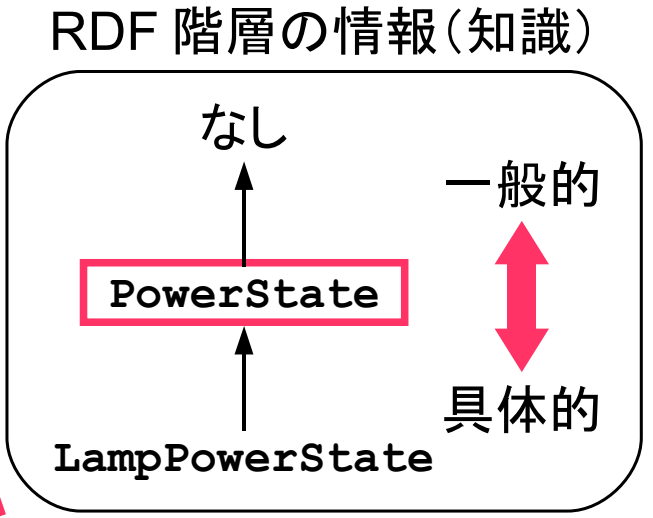
- 選択要素の意味
  - UI 記述の具体性を向上
  - サービスの内容に具体的な UI 生成の手助け
  - RDF[25] クラスとして表現(RDF の記述能力を利用)

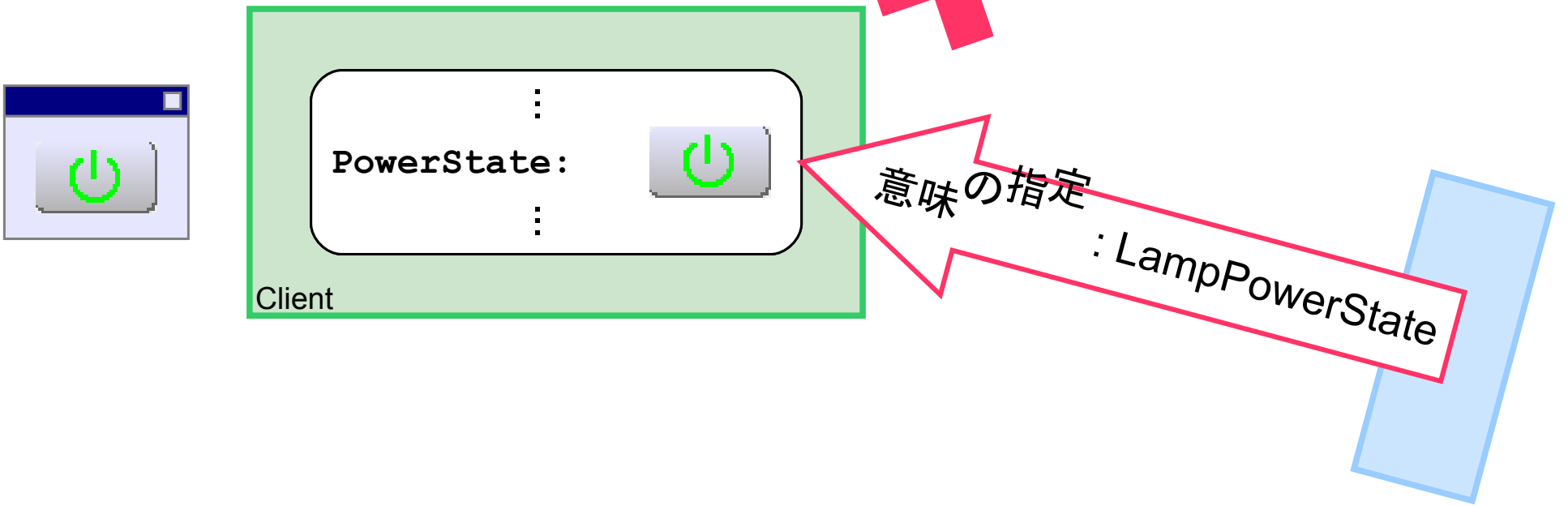


[25] E. Miller, R. Swick, and D. Brickley. Resource description framework (RDF), 2004. Available at http://www.w3.org/RDF/.

意味の導入(2/2)

- 意味の推論
  - RDF クラス階層の情報を利用
  - 指定されたよりも一般的な意味に対応したウィジェットで代替





意味の記述の利点と課題

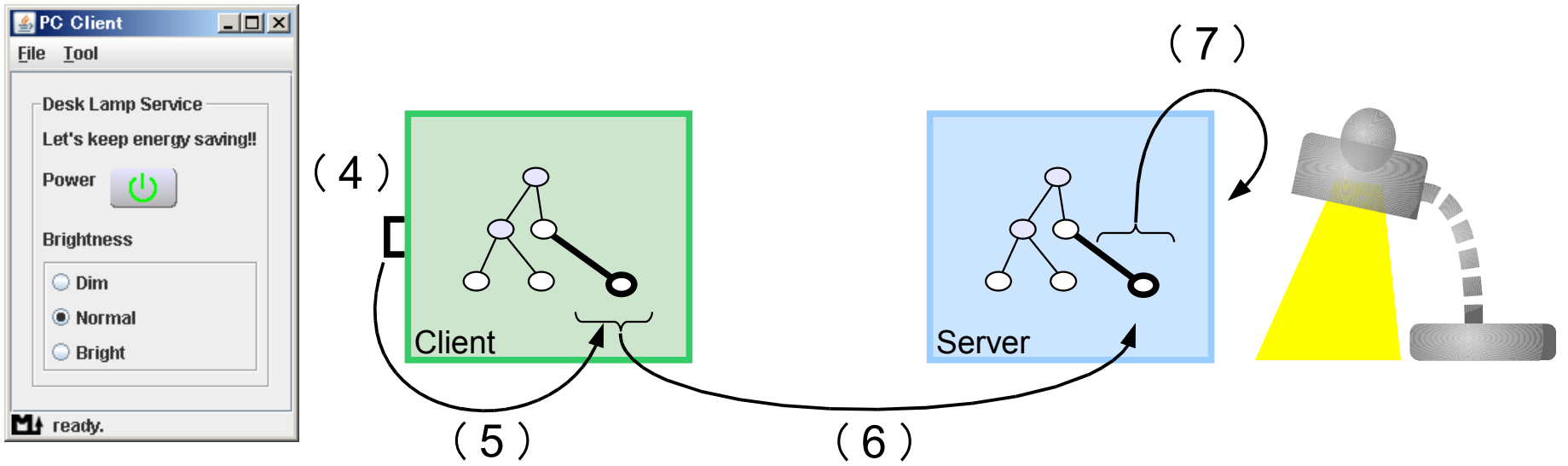
- 利点
  - 特定のプラットフォームに依存しない
  - 規格制定の段階で全てを定義する必要なし
  - 拡張性
- 課題
  - 概念ラベルとしての開発者間での合意形成が鍵
  - 複雑な意味指定を伴うUI 生成

… 意味の指定を
有効利用したUI 生成

クライアント・サーバの連携(2/2)

- UI 操作～送信
  4. ユーザが UI を操作
  5. モデルの現在の選択状態を変更
  6. 木構造の変化をメッセージとして送受信
  7. サーバが対応するサービスを実行

クライアント・サーバ間の状態の共有



UIアーキテクチャ

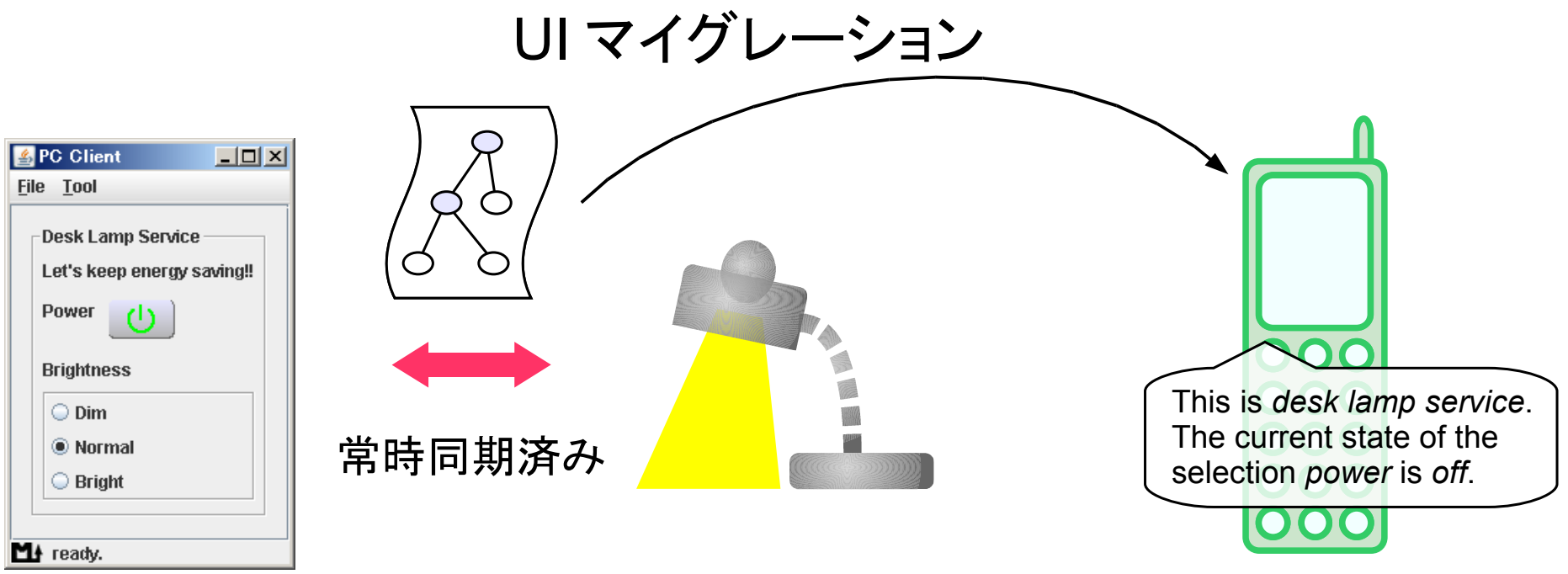
- 対象とする課題
  - モデル・ベース UI 設計
    1. 生成される UI の具体性向上（の可能性）
    2. UI マイグレーションの実現
    3. モデルからの UI 生成手法
       - 特に広く使われている GUI の生成について

UIアーキテクチャの研究

UIマイグレーション

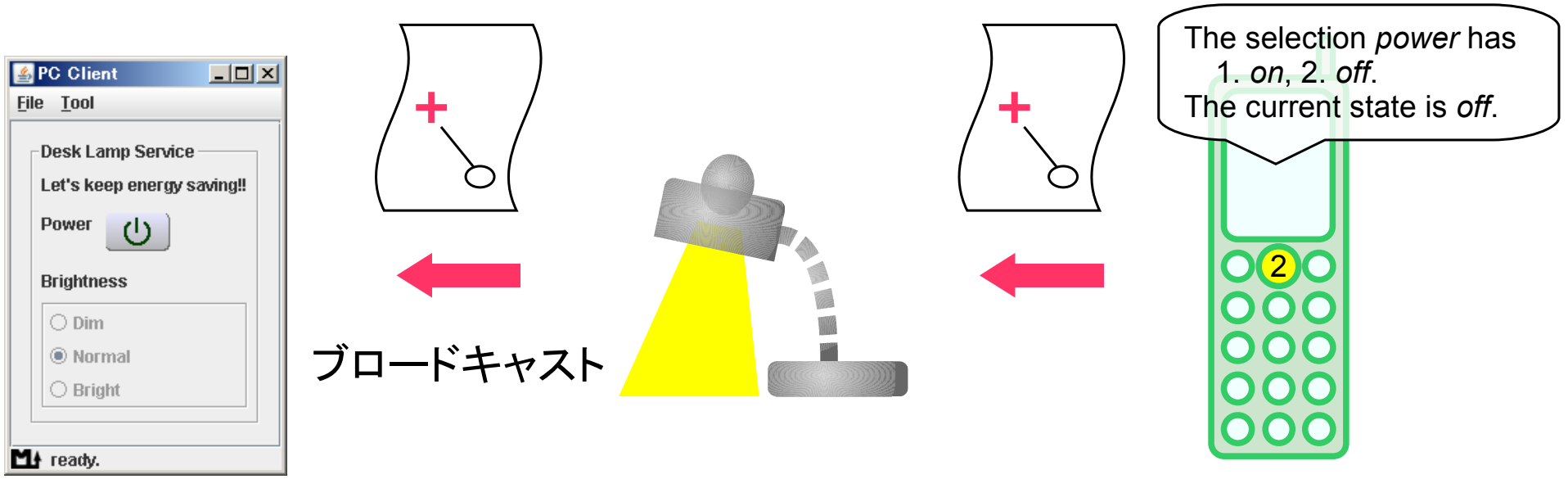
- クライアント・サーバ間で共有された状態の利用
  - 新しいクライアントによるサーバからの状態の受け取り
  - 現在の状態を収集する必要なし

→ 設計の利点



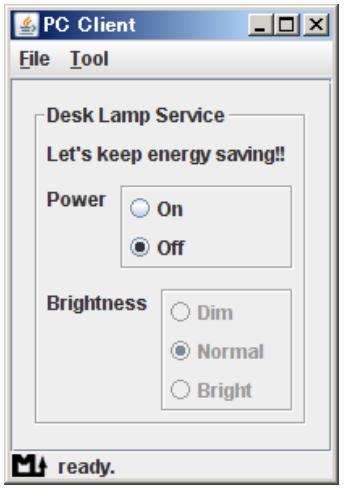
同期UI

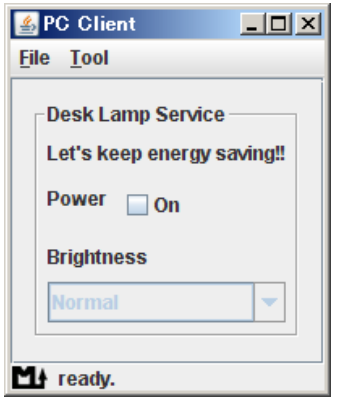
- クライアント間での状態の共有
  - サーバはクライアントの状態を常に同期済み
  - 他クライアントへのメッセージのブロードキャスト

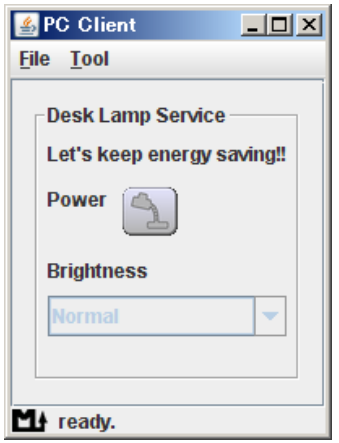


実現可能性の検証(1/2)

- ICLS ライブラリ
  - Java による実装
- サーバ(シミュレーション)
  - 卓上スタンドサーバ
  - AV 機器サーバ
- GUI クライアント
  - Swing ツールキット採用
  - レイアウト手法



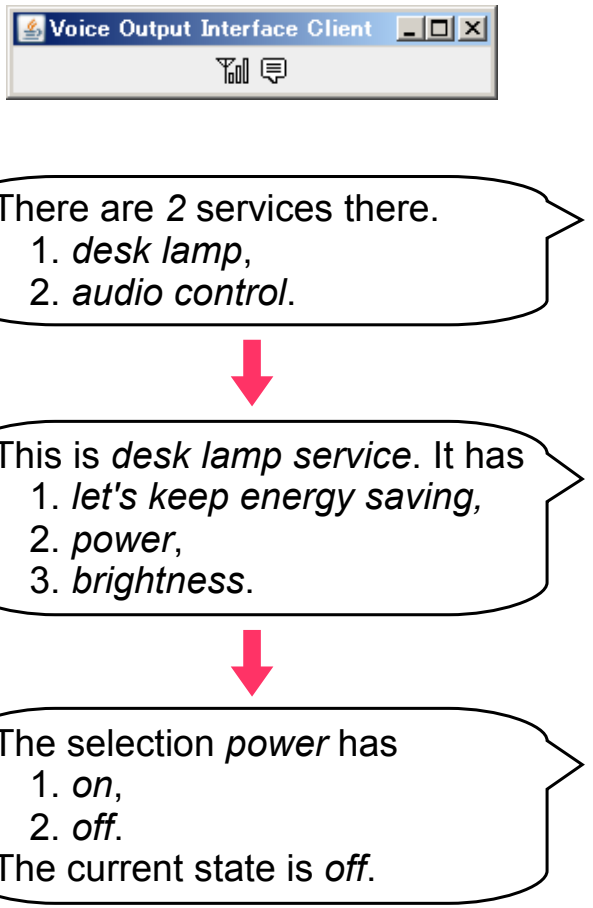




実現可能性の検証(2/2)

- 音声出力クライアント
  - 音声出力 [30] による階層メニュー構造の表現
  - キーパッドによる入力

携帯電話のような端末を想定



[30] Sun Microsystems. FreeTTS 1.2, 2005. Available at http://freetts.sourceforge.net/.

実装における課題

- UI 生成手法の検討
  - GUI → GUI レイアウトの自動生成（第 3 章）
  - 音声 UI
    - 木構造のモデルを音声 UI に変換する手法
    - ユーザへの適応（カスタマイズ性）
  - その他の UI
- UI としての使いやすさの評価

…GUI 以外の UI の
生成手法，評価

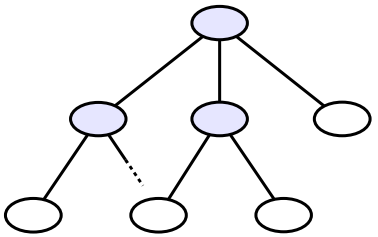
GUIレイアウト自動生成

- 対象とする課題
  - モデル・ベース UI 設計
    1. 生成されるUIの具体性向上(の可能性)
    2. UIマイグレーションの実現
    3. モデルからのUI生成手法
       - 特に広く使われているGUIの生成について

GUI(レイアウト)自動生成の研究

課題（1/3）

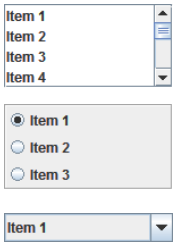
- モデル・ベース UI 設計における GUI 生成 [32]
  - UI モデルからどのように GUI を生成するか



How?

どのように UI 機能に対応するウィジェットを選択するか

SEi



Which?

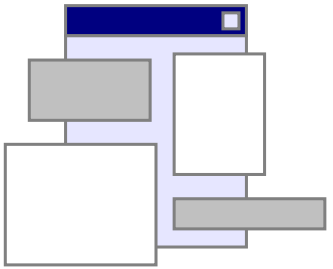
UI 機能（選択要素）　　ウィジェット候補

[32] S. Lok and S. Feiner. A survey of automated layout techniques for information presentations. In Proc. of the 1st Int. Symp. on Smart Graphics, pp. 61–68, USA, 2001. ACM.

課題(2/3)

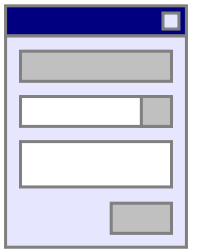
- 望ましさの観点
  - 一般的なユーザビリティ(ガイドライン)
  - ユーザ(環境)への適応(好み，使いやすさ)
- 選択の戦略

✖戦略１(望ましさだけで選択)



  - 大きいウィジェットは使いやすい傾向

- 戦略２(程々に望ましいものを選択)



  - 全体が収まる範囲で良い感じに

課題(3/3)

- モデル・ベース UI 設計における GUI 生成
  - 「程々に望ましいウィジェットを全体が収まるよう選択」

  - 一般的なユーザビリティ(ガイドライン)
  - ユーザ(環境)への適応(好み，使いやすさ)

あらかじめ用意しておくことが困難

しかも，ICLS ではクライアントにおいて動的に生成 → スピードが重要

提案手法

- Flexible Widget Layout 問題
  - ウィジェットの選択
  - レイアウトの（主観的）望ましさ

組合せ探索問題
ファジィな制約

- ファジィ制約充足問題（ FCSP ）[33] として定式化
  - 変数間の制約を全て満たすような変数への割当てを決定する組合せ探索問題
  - 不完全に充足される解を求め，現実の問題解決への有用な情報を提供

[33] Z. Ruttkay. Fuzzy constraint satisfaction. In Proc. of the 3rd IEEE Conf. on Fuzzy Systems, pp. 1263–1268, USA, 1994. IEEE.

レイアウト問題の概要(1/4)

- Flexible Widget Layout 問題
  1. ウィジェット候補集合の決定
     - 選択行為モデルの各要素と対応
  2. それぞれの候補集合からのウィジェットの選択
     - ウィジェットの組合せ探索問題

  - ウィジェットの特性
    - 個々に最小サイズ $ms_w = \langle ms.width_w, ms.height_w \rangle$
    - 種類ごとに望ましさ $0 \leq \alpha \leq 1$

↳ ユーザ毎に定義可能(カスタマイズ性)

レイアウト問題の概要(2/4)

- レイアウトの制約
  - レイアウトの実現可能性
    - ダイアログ・ボックスに全てのウィジェットが収まるかどうか

      → 必ず収める
  - レイアウトの望ましさ
    - 選択されたウィジェット各々の望ましさの最小値

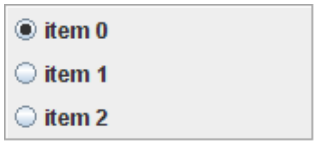
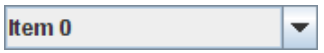
      → 出来るだけ最大化する

「レイアウト可能で，出来るだけ望ましい解」

ウィジェットの組合せ

レイアウト問題の概要（3/4）

- レイアウトしやすさと使いやすさの関係 [46, 47, 60]

	ラジオ・ボタン	ドロップダウン・リスト・ボックス
外見		
UI機能	同じ	同じ
レイアウトしやすさ	より悪い	より良い
使いやすさ	より良い	より悪い

[46] S. L. Fowler. GUI Design Handbook. Mcgraw-Hill Companies, Inc., 1997.
[47] Apple Inc. Apple human interface guidelines, 2008. Available at http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines/OSXHIGuidelines.pdf.
[60] J. Tidwell. Designing Interfaces O'Reilly Media, Inc., 2005.

レイアウト問題の概要(4/4)

- 本論文におけるレイアウト・ルール(制約)
  - 親子間の包含関係
  - ウィジェットの望ましさ

- 一対比較法などによる事前カスタマイズ
- 利用中の学習

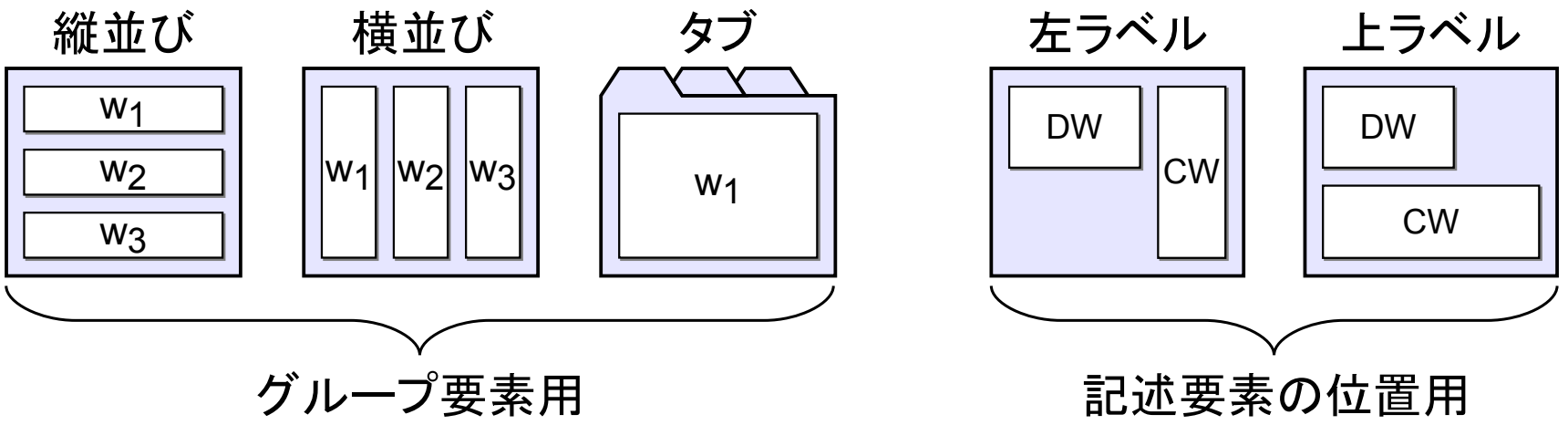
- その他に考えられるルール
  - ウィジェットの兄弟要素間の関係
    - 種類の統一
    - 揃え方
  - モデルの意味とウィジェット選択

… さまざまなレイアウト・ルールの(ファジィ)制約による表現

ウィジェット（1/2）

- コンテナ・ウィジェット
  - 配置の選択を「コンテナ・ウィジェットの選択」として表現
  - グループ要素と「記述要素の位置」

$\Leftrightarrow$ コンテナ・ウィジェット候補集合 $W_i \subset W_C$

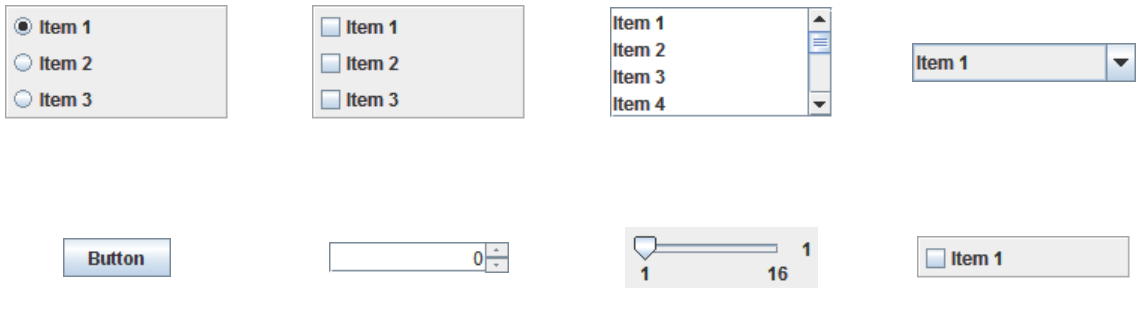


ウィジェット(2/2)

- ノーマル・ウィジェット
  - 多くのツールキットで採用されている 8+2 種類
  - 選択要素と記述要素

⟷ ノーマル・ウィジェット候補集合 $W_i \subset W_N$

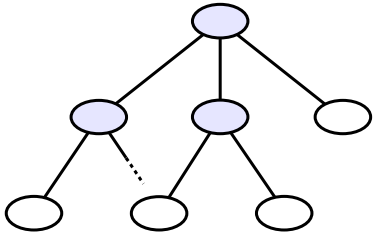
選択要素用



記述要素用 { キャプション，略称ラベル

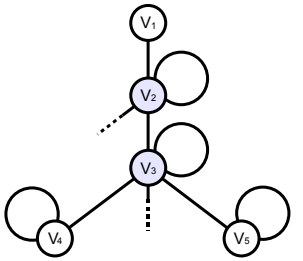
レイアウトの流れ

1. UI モデルから
   FCSP を生成



UI モデル

フェーズ 1



FCSP

フェーズ 2

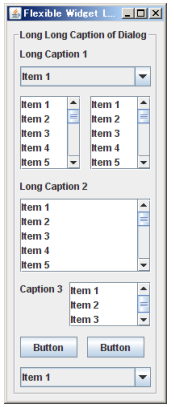
2. FCSP を解き,
   ウィジェットの組合せを取得

3. レイアウトの実行

$V_1 = 3$
$V_2 = 1$
$V_3 = 0$
$V_4 = 3$
$V_5 = 2$
⋮

変数割り当て

フェーズ 3



レイアウト

ファジィ制約充足問題(FCSP)

- FCSP の構成要素
  - 変数集合 : $X = \{x_1, \ldots, x_m\}$
  - ドメイン集合 : $D = \{D_1, \ldots, D_m\}$
  - ファジィ制約集合 : $C = \{c_1, \ldots, c_r\}$
    - メンバーシップ関数 : $\mu R_h(v[S_h])$ → $0 \leq$ 制約充足度 ≤ 1

      S_h: 制約の範囲，v: 全変数への割り当て
- FCSP の充足度
  - 全制約充足度の最小値 : $Cmin(v) = \min(\mu R_h(v[S_h]))$
    - $Cmin(v) > 0$ のとき，割り当て v は解

定式化

- ウィジェットのサイズと位置を変数とし**ない**
  - 変数 :　　割り当てで候補集合からの選択を表現
  - ドメイン :　ウィジェット候補集合と対応
  - 制約 :　　望ましさと親子の包含関係を表現

→ ドメインの規模縮小に有効

- サイズと位置を変数とした場合（初期段階）
  - 十分な速度を得られず
    - 探索空間の拡大が原因と見られる

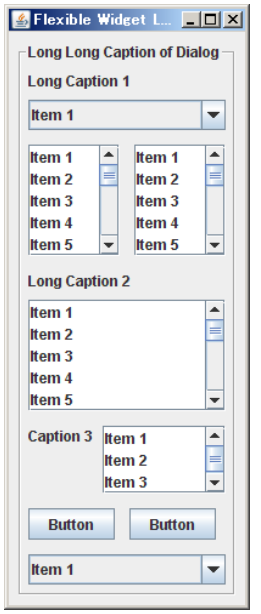
デモンストレーション

Javaによる実装

※ Java アプレットによるデモ公開ページ
http://aiwww.main.ist.hokudai.ac.jp/~takty/demo/fwl.html

レイアウトのスピード(1/2)

- 予備実験
  - {モデルの複雑さ，ウィンドウ・サイズ} ⇔ {時間，望ましさ}
    （十分な高速性は得られるか？）
  - モデルの複雑さ⇔時間
    - 環境
      - Java6
      - Windows XP
      - Turion 64 (2.0 GHz)
      - 望ましさを経験的に定義
    - 条件
      - 複雑さの変更(+1 ～ +3)
        (Sample の元となるモデルに選択要素を追加)

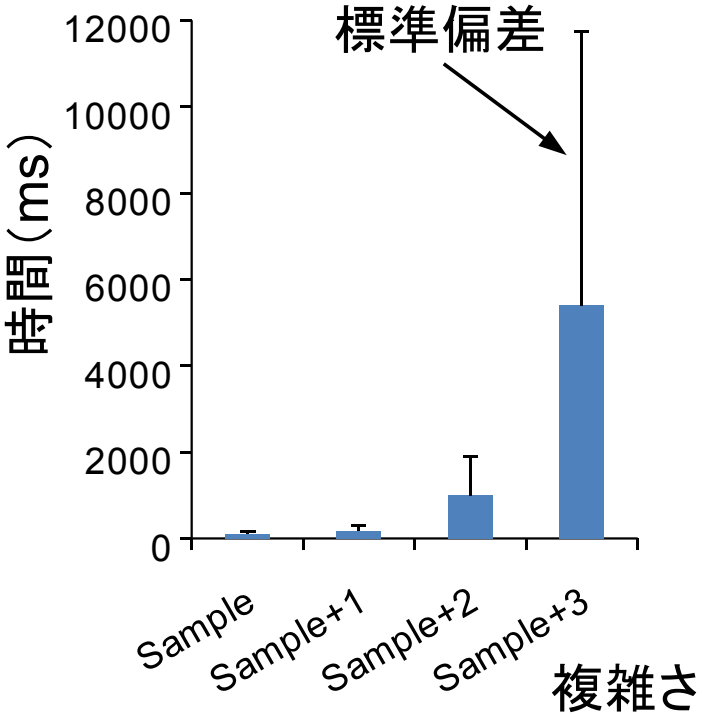


Sample

レイアウトのスピード(2/2)

- 結果(モデルの複雑さ⇔時間)
  - レイアウト時間の依存関係
    - モデルの複雑さ
    - ウィンドウ・サイズに依存
  - 例題の規模で平均 103ms
    - UI の生成時間として十分高速

      (一般に 1000ms が待たされていると感じない目安※)



… 提案手法で既存のダイアログ・ボックスを再構成し有効性を評価

※ S. K. Card, G. G. Robertson, and J. D. Mackinlay. The Information Visualizer, an Information Workspace. In Proc. of CHI '91, pp. 181–188, USA, 1991.

まとめ

- モデル・ベース UI 設計における課題
  - 生成される UI の具体性向上 & UI マイグレーション
    - Interface client/logic server (ICLS)
    - 実現可能性を検証
  - モデルからの（Ｇ）UI 生成手法
    - Flexible widget layout (FWL)
    - ある程度の規模の問題は十分高速に解けることを検証

今後の課題

- Interface client/logic server (ICLS)
  - モダリティを越える情報の出力
  - 意味を活用するUI生成
  - GUI以外のUIと評価
- Flexible widget layout (FWL)
  - 望ましさ決定手法
  - レイアウト・ルールの追加
  - 提案手法の検証