状態遷移記述によるユーザインタフェースの動的合成機構

3V−03

門田 昌哉[1] 高橋 元[1] 中澤 仁[2] 徳田 英幸[1,2]

[1] 慶應義塾大学 環境情報学部 [2] 慶應義塾大学大学院 政策・メディア研究科

1 はじめに

ホームネットワーク上に多様なサービスが存在する環境において，任意の入出力機器を用いてサービスを制御するためのユーザインタフェース (User Interface:UI) 構築機構が必要不可欠である．サービスとは，家電機器やセンサー等のデバイスが持つ機能やアプリケーションを指し，入出力機器とは，ユーザが制御機器として用いる携帯電話や PDA 等の携帯端末を指す．

従来のサービス制御方式では，サービスと UI が強く結合しているため，個々に異なる UI を持つサービスを集中制御できない．複数のサービスを制御するためには，物理的に管理ユニットを設置したり，UI を個々の入出力機器に対して静的に記述しなければならない．

本稿では，実行時にユーザの要求に従ってサービスをグループ化し，複数のサービスに対する UI を動的に合成する機構を提案する．UI の記述には，入出力機器に非依存な状態遷移記述を用いる．複数の状態遷移記述を合成することにより，任意の入出力機器上に統合 UI を構築し，複数のサービスの集中制御を実現する．

2 研究の概要

本機構は，先行研究によって実現された機構 Universal Interface[1] を拡張する形で，同一入出力機器上に複数のサービスに対する UI を動的に合成し，サービスの集中制御を実現するものである．Universal Interface において，個別のサービスに対する UI は状態遷移記述を用いて生成される．本機構は，実行時にユーザがサービスを集約，並列，直列にグループ化することにより，サービスごとに異なる UI の合成を実現する．以下に，状態遷移記述の概念を説明すると共に，それを用いたサービスのグループ化および UI の合成機構について述べる．

2.1 状態遷移記述

状態遷移記述とは，XML を用いたドキュメントベースの UI 記述であり，ユーザとサービス間のインタラクションを個々の状態に分割し，それらを遷移させることによってサービスを制御するという概念に基づく．状態は，サービスに対するユーザの入力を示す設定状態と，サービスの実行状態に分類される．状態の遷移は，ユーザの入力またはサービスからのフィードバックによって行う．図 1(a) にこれを概念的に示す．○で表されるものが設定状態であり，●で表されるものが実行状態である．個々の設定状態は，それぞれ一つ以上の実行状態に関連づけられて記述される．

たとえば，ビデオの予約録画を行う時，ユーザは日付を設定し，時間を設定し，チャンネルを設定し，最後にそれらの設定を実行する．ユーザはいくつかの設定状態を経て，実行状態に遷移する．実行状態は，サービスからのフィードバックによって初期状態に遷移する．

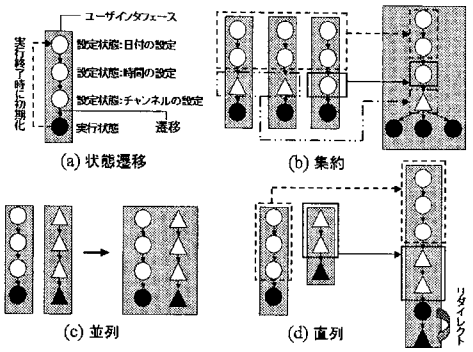

図 1: 状態遷移記述と合成規則

2.2 ユーザインタフェースの合成

UI の合成とは，複数のサービスが個々に持つ UI を，ある合成規則に従って一つの UI として同一入出力機器上に構築することである．合成を行うことにより，複数のサービスを集中制御する統合 UI をユーザに提供し，サービス制御におけるユーザビリティを向上させられる．

本機構では，UI の合成規則を集約，並列化，直列化によるサービスのグループ化によって定義する．UI の合成規則は，実行時にユーザが選択する．ユーザの要求を反映して UI を動的に合成することにより，ユーザがサービスを実行する文脈に即した統合 UI を構築できる．以下に，個々の合成規則の目的と実現方法を述べる．

集約

UI の集約は，同種類のサービス制御におけるユーザの状態遷移を一つにまとめ，複数の UI における同一操作の繰り返しを省く．ディスプレイ，スピーカ，ライト等のデバイスは環境に多数存在しており，それらのデバイスが提供するサービスを一回の操作で制御できると便利である．同種類のサービスを集約した UI を構築すると，既存の静的な管理ユニット等は不要になり，動的にユーザの要求に合わせたサービスの集中制御を実現できる．これを図 1(b) に示す．集約できない設定状態が存在する場合，個別に状態遷移に連結される．実行状態への遷移は，全ての設定が終了した後，同時に行う．

並列

UI の並列化は，同一入出力機器上で複数のサービスを並列に制御する UI を構築する．たとえば，プレゼンテーションを行うとき，ライトコントロールとスライドショーを並列に制御できると便利である．これにより，個々のサービスに対する UI を個別に起動する必要がな

Dynamic User Interface Composition with State Transition Definition

Masaya Kadota[1], Gen Takahashi[1], Jin Nakazawa[2], Hideyuki Tokuda[1,2]

[1]Faculty of Environmental Information, Keio University

[2]Graduate School of Media and Governance, Keio University

E-Mail: masaya@ht.sfc.keio.ac.jp

く，複数のサービスを同一のUIを用いて制御することが可能になる．これを，図1(c)に示す．複数の状態遷移は個別のUIとして同一入出力機器上に生成され，ユーザの入力を並列に受け付ける．

直列

UIの直列化は，複数のサービスを逐次的に実行するUIを構築する．ユーザは複数のサービスに対する設定をまとめて行い，サービスはユーザが指定した順番に従って自動的に実行される．この時，なんらかの出力を伴うサービスが実行された場合，その出力は次のサービスの入力にリダイレクトされる．これを，図1(d)に示す．概念的には，Unixシェルにおけるパイプ機能に似ている．例えば，ペットの状態を監視するカメラで画像を取り込み，それをディスプレイに表示するサービスの入力へ逐次的にリダイレクトする場合等が想定される．出力されたデータが入力に妥当な形式でない場合や入力を必要としない場合，本機構は例外を発生して実行を終了するか，データを棄却して次のサービスを実行する．

3 システムの設計

本機構の構成を図2に示す．本機構は，各サービスごとに配置されるUI記述，各入出力機器に配置されるモジュール群，ホームネットワーク上に配置される共有スペースから構成される．

- **UI記述**
  UI記述は状態遷移記述と，通信規約記述で構成される．通信規約記述とは，サービスとUI間の通信プロトコルを記述するものであり，既存のプロトコルやサービスに固有な通信プロトコルを記述できる．
- **モジュール群**
  各入出力機器上のモジュール群は，サービスと通信を行う通信部，状態遷移記述を合成する合成部，入出力機器の機能に合わせた形態のUIを生成するUI生成子から構成される．UI生成子は，その入出力機器の機能に合わせた表現形態のUIを生成するツールキットからなり，インストール時に入出力機器の機能定義文書によって利用するツールキットが設定される．ツールキットとは，HTMLやSwing等のGUIや，音声UIを生成するためのモジュールである．
- **共有スペース**
  共有スペースは，直列合成において出力を他の入力にリダイレクトする際，出力を一時的に保存するネットワーク上の記憶領域である．記憶領域は，複数の入出力機器およびサービス間で共有される．

4 実装

本機構のプロトタイプ実装を，Java言語を用いて仮想情報家電機器を実現するミドルウェアVNA[2]上で行った．サービスの検索及びサービスとUI間の通信は，同ミドルウェアが提供する機能を利用する．

プロトタイプ実装では，入出力機器としてPDAを用いて，ライトコントロールの集約，DVDプレイヤーとAVアンプの並列化，デジタルカメラとディスプレイの直列化によるUIの合成を行った．PDAという身近な入出力機器上で動的にUIを合成することにより，複数のライトを制御する電源管理ユニットや，複数のサービスを制御するための静的なUI記述が必要なくなる．これにより，ユーザがサービスを利用する状況に適応した統合UIを構築し、ユーザビリティを向上させられる．

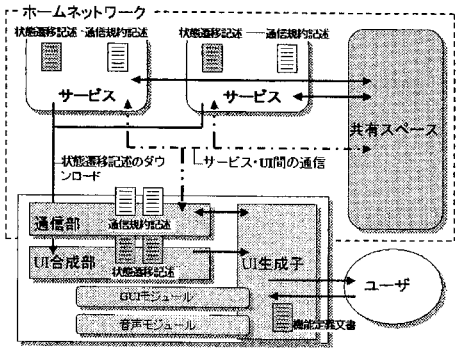

図2: システム構成図

5 関連研究

動的にUIを構築する機構として，Document-based Framework[3] や ICrafter[4] 等がある．Document-based FrameworkではISLというXMLを用いたUI記述を用いるが，GUIに偏っており，実現されているUIの合成は，同一画面に複数UIを並列に表示する機能だけである．これに対して，ICrafterではSDLというUI記述を用いて，全てのサービスの電源を切る等，本稿におけるUIの集約および並列化を実現している．しかし，サービスを合成するUIを静的に記述しなければならないため，実行時にユーザの要求を反映してUIを合成することができない．これに対して，本機構はUIの直列化による入出力のリダイレクト，およびUIの集約・並列化を実現しており，それらのUIの合成はユーザの選択に基づいて動的に行われるため，静的なUIの記述を必要としない．

6 まとめと今後の課題

本稿では，状態遷移記述によるUIの動的合成機構の概念と，その設計及び実装について述べた．本機構を用いることにより，実行時のユーザの要求を反映した形で，サービスを集中制御するUIを同一入出力機器上に構築できる．その反面で現状では，ユーザが異なる入出力機器を用いる度に，同様のサービスのグループ化が必要となり，ユーザビリティを損なう恐れがある．今後の課題として，ユーザが合成したUIを異なる入出力機器間で共有する機構を組み込む必要がある．

参考文献

[1] 高橋元，門田昌哉，中澤仁，徳田英幸: 入出力デバイスに非依存なユーザインタフェースの動的生成機構，システムソフトウェアとオペレーションシステム研究会，情報処理学会 (2002).

[2] Nakazawa,J., Tobe.,Y, and Tokuda,H.,: On Dynamic Service Integration in VNA Architecture, Vol.E84-A, No.7, 電子情報通信学会論文誌 (2001)

[3] Hodes,T.D., and Katz,R.H.: A Document-based Framework for Internet Application Control, 2nd USENIX Symposium on Internet Technologies and Systems(1999.10)

[4] Ponnenkanti,S.R., Lee,B., Fox,A., Hanrahan,P., and Wingnograd,T.: ICrafter:A Service Framework for Ubiquitous Computing Environments, Ubicomp(2001)