常時利用するシステムのためのユーザインタフェース環境の開発
—眺めるだけの置物コンピュータの実現に向けて—

１．背景

機器の高性能・小型化、高速かつ常時接続可能なネットワーク環境の普及、無線通信環境の整備などにより、ユビキタスコンピューティングやウェアラブルコンピュータ、情報家電などが現実のものとなりつつある。また、デジタル放送や電子政府などのネットワークサービスの整備も進められている。このように、日常的にコンピューティングサービスを利用できるようになることが見込まれている。

しかしながら、現在のユーザインタフェースの主流は GUI (Graphical User Interface) による対話型システムである。対話型システムはユーザの操作に応じてシステムが応答するという形態を採るが、別の見方をすると、ユーザが何か操作しない限り何も得られないとも言える。これは、ユーザが明確な目的を持ってシステムを操作する場合はあまり問題にならないが、日常的に利用するシステムの場合には例えば次のような問題が想定される：

- ユーザは操作を行ってみないとどのような結果が得られるのか分からないので、試行錯誤的に操作を行う必要がある。
- 「何か面白いものを見たい」といったような漠然とした要求しか持っていないユーザは、何から手を付けて良いのか分からないかもしれない。
- ユーザがリラックスしているような状況では往々にして積極的に操作することが煩雑に感じられる。
- モバイル機器や情報家電などにおいては GUI が前提としているようなポインティングデバイス（マウスなど）は使いにくいこともある。

２．目的

本プロジェクトの目的は、ユーザが積極的に操作を行わなくてもサービス（従来のアプリケーションやインターネットサービスなど、ユーザに対して何らかの価値を与えるものを総称して「サービス」と呼ぶ）を享受できる、常時利用するためのシステムのためのユーザインタフェース環境 i^2SPEC: Interaction-Inspiring Service Platform for Everyday Computing を開発することにある。

本プロジェクトが対象としている常時使用するシステムとは、例えばつけっぱなしのテレビのようなものである。具体的な装置としては、壁にかけられているコンピュータ、置物のように部屋の片隅に置かれているコンピュータ、ホームサーバの画面を映しているテレビ、などを想定する。概念的には、ユーザのインタラクションの有無に関わらず常になんらかのサービスをユーザに提供するシステムである。このようなシステムによって提供されるサービスは、従来のパーソナルコンピュータで動作するワープロや表計算などのアプリケーションよりも、時計、カレンダー、音楽プレイヤなどの日常的に使用するものや、ニュース、天気予報、株価などの様々

な情報配信サービスといったものが主となる。また、複数のユーザの存在に配慮する必要がある。

i²SPEC は次のような特徴を持つことにより上述の課題の解決を目指す。

- ユーザが操作を行わなくてもシステム側から働きかけてサービスを提示する。
- ユーザはテレビのチャンネルを変えるような簡単な操作で所望のサービスを選択して利用することができる。
- ユーザや外界の状況およびサービスの履歴や状態などのコンテキスト情報に応じてサービスが適切な働きかけを行うことを可能にする。

３．開発の内容

i²SPEC は上記の目的を達成するために下記の機能を実装している：

- サービスの要求に応じて、サービスの内容およびユーザインタフェースを自動的にユーザに提示する機能。
- 複数のサービスからの働きかけの要求を調停して、サービスをユーザに提示するタイミングをスケジューリングする機能。
- サービスが適切な働きかけを決定するためのコンテキスト情報を管理し、サービスに供給する機能。
- ユーザの明示的な操作によってサービスを切り替える機能。
- サービスモジュールの提供するコンテンツおよびユーザインタフェースをレンダリングする機能。

図 1 は i2SPEC のソフトウェア構成図である。

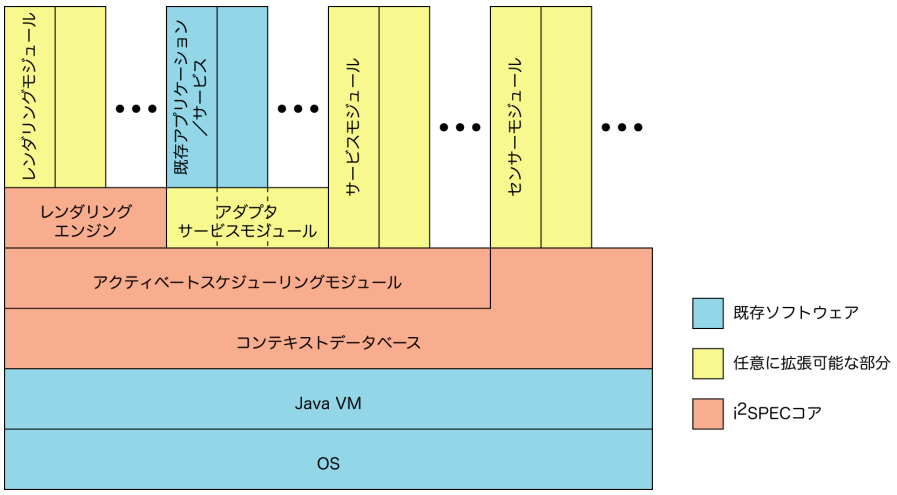

図 1: ソフトウェア構成図

以下、それぞれの構成要素の概要を説明する。

- サービスモジュール

  サービスはアプリケーションとネットワークサービスの二つに大別される。アプリケーションは従来のパーソナルコンピュータ上で動作するアプリケーシ

ョンと同様に、ローカルマシン上で動作する。ネットワークサービスはリモートマシン上で動作し、ローカルマシンは処理の要求と結果の受信を行う。

処理の内容はそれぞれのサービスの具体的な実装であるサービスモジュールに依存し、i²SPEC とはサービスモジュール API で接続される。このため i²SPEC から見ると、アプリケーションとネットワークサービスは同様に扱われる。アダプタとなるサービスモジュールを実装することで既存のアプリケーションやネットワークサービスを i²SPEC 上で利用することもできる。

- センサーモジュール

  コンテキストデータの一部である、ユーザや外界の状況などを収集するためのもの。具体的には、温度計やマイクなどの特殊なセンサーを利用したものや、一般的に用いられるキーボードやマウス等の入力デバイスの使用状況を監視するものなどがある。

- コンテキストデータベース

  ユーザや外界の状況、およびサービスの履歴や状態などの、コンテキストデータを管理する。コンテキストデータは一種のイベントデータとして扱われる。コンテキストデータはサービスモジュール、センサーモジュール、アクティベートスケジューリングモジュール（後述）によって獲得され、コンテキストデータベースに格納される。

  また、アクティベートスケジューリングモジュール（後述）およびサービスはコンテキストデータを参照することができる。この仕組みにより、あるサービスモジュールが別のサービスやセンサーによって獲得されたコンテキストデータを参照して処理やユーザへの働きかけの内容を決定することができる。

- アクティベートスケジューリングモジュール

  サービスおよびユーザからの要求に応じて、複数のサービスを自動的に切り替えて提示するタイミングをスケジューリングする。

  サービスはアクティベートスケジューリングモジュールに対して、重要度や緊急度などのサービス内容のパラメータや、時刻や特定の状況の発生などのコンテキストデータの条件を登録する。アクティベートスケジューリングモジュールはこれらの条件やパラメータと、コンテキストデータベースに蓄積されているコンテキストデータを参照して、提示するサービスを決定する。また、直前に提示していたサービスの内容との関連度にも配慮する。

  一方、ユーザは入力装置を使用して、アクティベートスケジューリングモジュールに対して提示されるサービスを明示的に切り替えることを要求できる。

- レンダリングモジュール

  サービスの内容、すなわちユーザに対して提示する情報や GUI などのユーザインタフェース部品を表示／再生する。このモジュールにより、それぞれのサービスモジュールから描画に関するルーチンを独立させることによって、サービスモジュールの開発を容易にする、画面のカスタマイズを可能にする、オ

ーバーヘッドを削減する、表現力を向上させるといった効果を図る。

レンダリングエンジンは複数のレンダリングモジュールを管理し、サービスモジュールから供給されるサービスの内容を表すデータを適切なレンダリングモジュールに割り当てる処理を行う。具体的には、サービスモジュールから受け取ったグラフィクスや音のデータを、それぞれのデータを処理できるレンダリングモジュールに渡す。

多様な OS に対応するために、i²SPEC は一部のセンサーモジュールやサービスモジュールを除いて Java で実装され、JDK1.4 以上が実行可能な環境で動作する。

４．従来の技術との相違

i²SPEC は以下の３つの新規性のある技術的特徴を有している：

- 確率的なサービス提示スケジューリング

  本プラットフォームの第一の特徴は、サービススケジューリング機能における､確率的なスレッド切り替えのモデルに基づくスケジューリング方式にある。これは、OS が CPU に対して複数のスレッドをスケジューリングすることによってマルチタスクを実現していることのアナロジーである。

- SVG を利用したカスタマイズ可能な提示画面

  SVG (Scalable Vector Graphics)形式のグラフィクスデータを扱うことにより、サービスモジュールが高度な表示形態を扱うことやユーザによるカスタマイズを容易にしている。

- コンテキストデータの表現形式

  コンテキストデータを共通に扱うための枠組みを提供し、コンテキストデータの標準化を試みている。

５．期待される効果

日常的にコンピューティングサービスを利用できるようなユビキタスコンピューティングやウェアラブルコンピュータ、情報家電などの新しい形態の機器、およびこれらの機器を対象とした新しいネットワークサービスの創出が期待される。

また、複雑な操作をしなくても利用可能であるので、デジタルデバイドの問題の解決も見込まれる。

６．普及の見通し

本開発成果は広く一般的に利用可能なものであり、特に各家庭のリビングルームに設置する情報端末や、個人のパーソナル端末のユーザインタフェース環境として適している。

７．開発者名

水口 充 （情報通信研究機構 mmina@acm.org）