

# 携帯端末上でコンテキスト依存プログラムを 記述するためのビジュアルプログラミング環境

Mobile Visual Programming Environment for Context-Aware Program

西本 裕貴    志築 文太郎    田中 二郎\*

**Summary.** 携帯端末上でコンテキスト依存プログラムを記述するためのビジュアルプログラミング環境である MoCoPro の開発を行った。ビジュアルプログラミング手法を用いることで、プログラムを視覚的に表現し、エンドユーザであっても容易にプログラムを記述することを可能にする。また、MoCoPro が動作するタッチパネルを搭載した携帯端末上での操作手法についても述べる。

## 1 はじめに

携帯端末に搭載された様々なセンサからユーザのコンテキスト情報を取得し、そのコンテキストに合わせた動作を携帯端末に行わせることが可能になってきている [1][2]。例えば、「本屋の前を通る時に、買いたい本のリストを表示する」が考えられるが、このようなコンテキスト依存プログラムはユーザによって多種多様な要求があるため、エンドユーザが自分自身で記述できることが必要となる。

本研究では、携帯端末上でコンテキスト依存プログラムを記述する環境である MoCoPro (Mobile Context Programming) の開発を行った。MoCoPro は、ビジュアルプログラミングの手法を導入し、プログラムを視覚的に表現することで、エンドユーザの理解を助け、容易にプログラムを記述することを可能にする。MoCoPro は、タッチパネル搭載の携帯端末を対象としており、本稿ではそれに適した操作についても述べる。

## 2 MoCoPro

図 1 は、図 3 のプログラムを作成している際の MoCoPro の画面の外観を示している。画面は複数のセルから成っており、1 つのセルに 1 つのコンテキスト定義やアクション定義を記述する。定義の記述は、コンポーネントを配置することにより行う。プログラムの実行は左側のセルから順に行われる。

### 2.1 ECA ルール

ECA ルール [3] と呼ばれるイベント駆動ルールに従い、センサ入力やメール受信などの携帯端末に起こるイベント (Event)、動作を実行するための制約条件 (Condition)、制約条件が満たされた際に実行する動作 (Action) を定義することによりプログラムを記述する。

Copyright is held by the author(s).

\* Hiroki Nishimoto, Buntarou Shizuki and Jiro Tanaka, 筑波大学 コンピュータサイエンス専攻

本システムでは、コンテキスト定義として、利用するイベントと動作を実行するために満たすべき制約条件を記述し、アクション定義として、制約条件が満たされた場合に実行する動作を記述する。

### 2.2 コンポーネント・コンポーネントセレクト

コンポーネントは、0 個以上の入力と、0 または 1 つの出力を持ち、アイコンで象徴される動作を行う要素である。例を挙げると、入力された文字列を画面に表示する Dialog コンポーネントや、入力された 2 つ文字列を連結した文字列を出力する Plus コンポーネントなどがある。コンポーネントの配置は、図 1 に示すコンポーネントセレクトから利用したいコンポーネントを選択することにより行う。コンポーネントの種類によって入力数は決まっており、コンポーネントを配置すると、入力数に応じたブランクコンポーネント (? のアイコン) が接続された状態で配置される。ユーザは、このブランクコンポーネントを実際のコンポーネントで埋めていくことで、プログラミングを進めていく。

コンポーネントセレクトは、コンポーネントの階層構造リストであり、コンポーネントを配置する際に使用する。コンポーネントセレクトは、コンポーネント上をタップすることで出現し、タップしたコンポーネントに接続されている入出力の型から、そこに配置できるコンポーネントのみを表示する。

### 2.3 フォルディングによる関数化

コンポーネント上をダブルタップすることで、それより右側 (入力, 引数に相当する) のコンポーネントを重ねて表示 (フォルディング) する。この際、あらかじめフォルディングを無効にしておいたコンポーネントや、ブランクコンポーネントは、フォルディングされない。例えば、図 2 左の一番左側のコンポーネントをフォルディングすると、図 2 右のようになる。また、フォルディングを利用して関数を作成することもでき、フォルディングしたコンポー

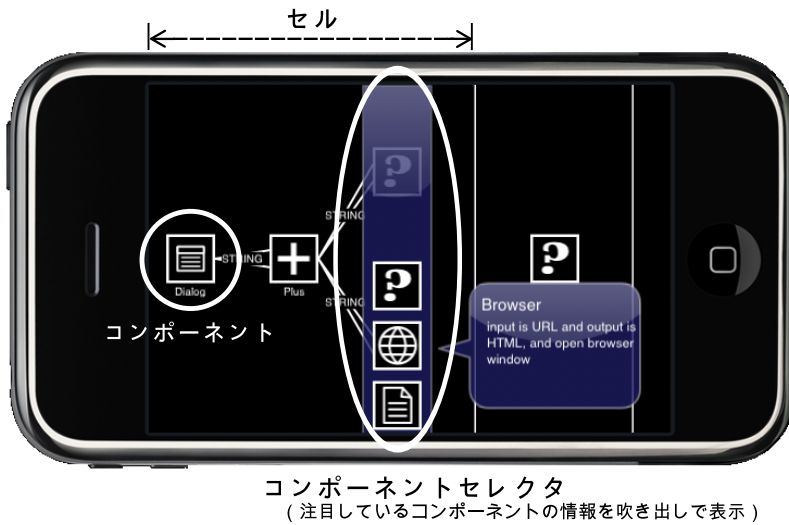


図 1. MoCoProの外観と各部名称

ネットに接続されているコンポーネントの型がそのまま関数の入出力となる。図2左は、メールを送信する動作の例であり、状況に応じて変更が考えられる送信先や本文に引用するメモのタイトルをブランクコンポーネントにしている。この後、フォルディングし関数化することで、再利用が可能となる。

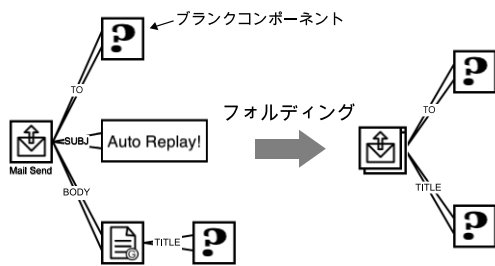


図 2. フォルディング

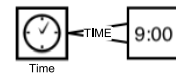
## 2.4 クロッシング

携帯端末画面の辺を外側から内側に向かって指で横切る(クロッシング)操作を行うことによりコマンドを呼び出す。横切る辺や指を離れた場所によってコマンドは異なり、下辺を横切り、コンポーネントセレクタ上で指を離れた場合は、コンポーネントセレクタの階層の移動を行い、下辺や上辺を横切り、エディタ上で指を離れた場合は、コンテキスト定義とアクション定義の切り替えを行う。

## 2.5 プログラム例

プログラム例として、「朝9時に、今日の予定とTODOを表示する」を記述したものを図3に示す。図3上は、コンテキスト定義の「朝9時に」に当たる記述である。図では、Timeコンポーネントが、TIMEと書かれたコネクタで、9:00と書かれたコンポーネントと接続されている。コネクタ上の文字列は、そこを流れるデータの内容を示しており、9:00のように、数字や文字列で表わされたコンポーネ

## コンテキスト定義



## アクション定義

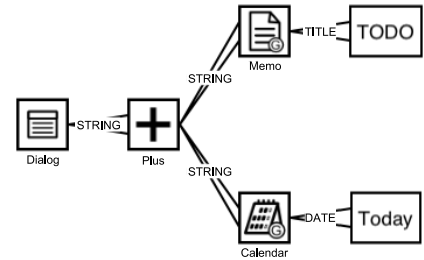


図 3. プログラム例

ントは、そのデータを出力するコンポーネントである。よって図では、入力された時刻がどうかを監視する Time コンポーネントに、9:00 という時刻を表すデータが入力されていることを表している。

図3下は、アクション定義の「今日の予定とTODOを表示する」に当たる記述である。Memo コンポーネントは、メモアプリケーションと連携し、入力されたタイトルのメモ本文を文字列として取得する。Calendar コンポーネントも同様であり、カレンダーアプリケーションから、指定した日付の予定を文字列として取得する。得られた2つの文字列は、Plus コンポーネントで結合し、Dialog コンポーネントで画面に出力する。

## 3 実装

iPod touch, iPhone 3G 上で動作するアプリケーションとして、iPhone SDK を使い、Objective-C で実装した。GPS や加速度センサなどのセンサ類は、端末に搭載されているものを利用した。

## 4 まとめと今後の課題

本稿では、携帯端末上でエンドユーザがコンテキストに依存したプログラムを記述する環境である MoCoPro を示した。今後は、複数人でのプログラミングに対応できるように機能拡張をする予定である。

## 参考文献

- [1] 寺田, 他. その場プログラミング環境の実現に向けて. 情報処理学会研究報告, 2007-UBI-46, pp.1-8, 2007.
- [2] Panu Korpipaa, et al. Context Management for End user Development of Context-Aware Applications. MDM 2005, pp. 304-308. 2005.
- [3] J.Widom, et al. Active Database Systems: Triggers and Rules for Advanced Database Processing. Morgan Kaufmann, 1996.