

xtel : ユビキタスコンテンツのための  
アジャイルプロトタイプ開発支援環境

徳久 悟<sup>†a)</sup>      石澤 太祥<sup>††</sup>      丹羽 義将<sup>†</sup>      粕谷 貴司<sup>††</sup>  
 勝本雄一朗<sup>††</sup>      石橋 秀一<sup>†</sup>      橋本 悟<sup>†</sup>      植木 淳朗<sup>†</sup>  
 稲蔭 正彦<sup>†</sup>

xtel : A Development Environment  
to Support Agile Prototyping of Ubiquitous Content

Satoru TOKUHISA<sup>†a)</sup>, Takaaki ISHIZAWA<sup>††</sup>, Yoshimasa NIWA<sup>†</sup>, Kenji KASUYA<sup>††</sup>,  
Yuichiro KATSUMOTO<sup>††</sup>, Shuichi ISHIBASHI<sup>†</sup>, Satoru HASHIMOTO<sup>†</sup>, Atsuro UEKI<sup>†</sup>,  
and Masa INAKAGE<sup>†</sup>

あらまし 本論文では、ユビキタスコンテンツ開発支援環境 xtel を提案する。ユビキタスコンテンツとは、生活に密着した生活者のための実空間アプリケーションにおいて、実空間上に存在するヒト、モノ、環境との間で起こるインタラクションを通じて経験されるコンテンツである。xtel は、センサやアクチュエータを接続し、近距離無線通信が可能な MCU(Micro Control Unit) ボード moxa, JavaScript のパーサ、コンパイラ、VM(Virtual Machine)、ライブラリから構成される MCU ボードのための Programming/Runtime 環境 Talktic、センサからの離散情報に加えて、動画・音声などの連続情報を扱うことが可能な P2P ネットワークライブラリ Entity Collaborator の 3 つのツールで構成される。これらのツールは、分散環境下での実空間アプリケーションであるユビキタスコンテンツを開発するにあたり、開発の技術的ハードルを低下させ、開発のスピードを向上させるといった共通の設計思想をもとに開発された。xtel を用いて開発された 3 つのユビキタスコンテンツをサンプルコンテンツとして紹介したのち、ワークショップを通じてツールとしての評価について述べる。

キーワード ユビキタスコンテンツ, xtel, アジャイルプロトタイプリング, MCU, VM, P2P, 開発支援

1. 背景

ユビキタスコンピューティングのための基盤技術が整備される中、アプリケーションのあり方が変容しつつある。無線環境の普及に伴い、ユーザはいつでもどこでもネットワークにつながるができる。さらには、センサデバイスがネットワークやデータベースにつながることによって、実世界で起こるさまざまな現象がデータ化され、共有されつつある。このような環

境の中で、キーボードやマウスといった従来の入力機器、ディスプレイや設置型スピーカといった従来の出力機器に限定されず、実空間上に同等の機能を組み込んだアプリケーションが出現している。すなわち、実世界の現象をセンシングし、その結果を実世界上でアクチュエーションする実空間アプリケーションである。

筆者らは、実空間アプリケーションのうち、生活に密着した生活者のためのアプリケーションをユビキタスコンテンツ [1] と呼んでいる。それは、実空間上に存在するヒトやモノ、環境との間で起こるインタラクションを通じて経験されるコンテンツである。Tangible Bits [2] が情報を触ることをコンセプトに置くのに対し、ユビキタスコンテンツは、情報に触れることができるだけでなく、物理的に生活空間に溶け込み、心理的に日常生活に深く結びつくことをコンセプトに置く。

<sup>†</sup> 慶應義塾大学大学院メディアデザイン研究科, 神奈川県  
Keio University Graduate School of Media Design, Hiyoshi  
4-1-1, Kohoku-ku, Yokohama, Kanagawa, 223-8526 Japan

<sup>††</sup> 慶應義塾大学大学院政策・メディア研究科, 神奈川県  
Keio University Graduate School of Media and Governance,  
Endo 5322, Fujisawa, Kanagawa, 252-8520 Japan

a) E-mail: dangkang@kmd.keio.ac.jp

筆者らは、ユビキタスコンテンツの生活への溶け込み方に応じて、3つの領域を設定した。第1に、「行動的 (behavioral)」ユビキタスコンテンツは、生活の中のある行動や行為に注目したコンテンツである。例えば、料理の上達を支援するためのフライパン [3]、読書という行為に注目し、読書に関連する様々な作業を支援するための道具 [4] などである。第2に、「視覚的 (visual)」ユビキタスコンテンツは、家具や衣類など、既存の生活環境の中に存在する外見の形状を踏襲して、そこに新たなユーザ体験を付与し、従来の体験を拡張するコンテンツである。例えば、植物に流れる電気状態をセンシングし、植物の状態を視覚化するポット [5]、机の表面に表出する影と遊ぶことが可能なデスク [6] などである。第3に、「時間的 (temporal)」ユビキタスコンテンツは、生活における時間感覚に注目したコンテンツである。家族のメンバの年齢に注目し、協調的に動作するフォトビューア [7]、ユーザの行動によって認知地図が形成される地球儀 [8] などである。なお、これらの領域は互いに排他的ではなく、重複するコンテンツもありうる。

このようなユビキタスコンテンツを制作するための開発支援環境を構築するにあたり、実現要件として以下の3点を掲げた。まず、要件1については、PC-PC、PC-デバイス、デバイス-デバイスという3つの分散環境をダイナミックに構築するための仕様である。次に、要件2については、これまで技術的障壁から開発に携わらなかったクリエイティブなデザイナーを開発者として取り込むための仕様である。最後に、要件3については、アジャイルプロトタイピングとユーザテストを繰り返し、コンテンツの質を向上させるための仕様である。

- (1) 分散環境での実空間アプリケーション
- (2) 技術的ハードルの低下
- (3) 開発スピードの向上

本論文では、上記の要件を踏まえて構築されたユビキタスコンテンツのための開発支援環境 xtel について述べる。xtel は、以下の3つのツールで構成され、Eclipse などの IDE(Integrated Development Environment) に組み込むことで、ユビキタスコンテンツのアジャイルプロトタイプを効率的に開発可能なプラットフォームである (図 1)。

- moxa: センサやアクチュエータを接続し、近距離無線通信が可能な MCU(Micro Control Unit) ボード

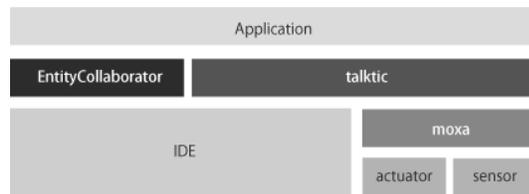


図 1 xtel システム構成  
Fig.1 Structure of xtel

- Talktic: JavaScript のパーサ、コンパイラ、VM(Virtual Machine)、ライブラリから構成される MCU ボードのための Programming/ Runtime 環境
- EntityCollaborator: SIP プロトコルを採用することで、センサからの離散情報に加えて、動画、音声などの連続情報を扱うことが可能な P2P ネットワークライブラリ

本論文では、第2章にてそれぞれのールの先行研究について紹介する。第3章では、開発支援環境としての xtel の特徴について述べたのち、各ツールの詳細について述べる。そして、第4章では、xtel を用いて開発された3つのコンテンツについて説明する。第5章では、3度に渡って開催されたワークショップを通じて、実際に xtel を使用してプロトタイピングを行った参加者たちの意見をもとに xtel の評価を行う。最後に、まとめと今後の展望について述べる。

## 2. 関連研究

本章では、xtel の3つのツールについてそれぞれの関連研究を取り上げ、それらの問題点について述べる。

### 2.1 センサデバイス

ユビキタスコンピューティングにおけるアプリケーション開発を容易にする無線通信機能を有するデバイスは、MOTE [9]、SmartIts [10] が代表的である。MOTE は、アドホックマルチホップネットワークを形成可能な無線センサ端末である。無線規格は、2.4GHz、IEEE802.15.4 規格を採用している。また、TinyOS と呼ばれる OS が搭載されており、開発は nesC と呼ばれる C 言語の拡張を利用する。SmartITs は、様々なセンシング機能を持ったセンサネットワークデバイスである。SmartIts は、Bluetooth 通信機能、あるいは、RFM communication によって無線通信を実現している。これらは、ボード上にセンサの構成とレイアウトが固定されているため、開発者が要求するレイアウトとボード上のセンサのレイアウトが異なる場合、

開発者はセンサ環境をハードウェアのレベルから構築しなければならないという問題点を持つ。

一方、無線通信機能を持たないものの、開発者自身でセンサやアクチュエータを簡易的に接続可能なデバイスには、Phidget [11], Gainer [12] が代表的である。これらは USB 接続でコンピュータから直接制御を行う装置環境である。また、Flash や Max/MSP をはじめとする様々な開発環境で開発を行えるようライブラリが提供されている。これらは、実行環境が PC であるために、小型化が要求されるアプリケーションや、近距離無線により環境情報を共有し協調的に動作するような分散アプリケーションへの適用が難しいという問題点を持つ。

## 2.2 MCU ボードのための開発環境

MCU ボードを含む統合開発環境は、Wiring [13], Arduino [14] が代表的である。Wiring は、I/O ボードとオープンソース開発環境 (Java) からなる統合開発環境である。Java ベースの Processing [15] 言語でプログラム可能である。Arduino は、I/O ボード、および、Wiring と同様のオープンソース開発環境から構成される統合開発環境である。Wiring から派生した独自言語を採用し、C/C++ 風の構文を用いている。なお、両者とも MCU ボードとの連携は開発環境に隠蔽されている。これらのメリットは、スクラッチからハードウェアおよびソフトウェアを設計・実装しない点、簡易なプログラミング言語を採用している点にあり、開発の技術的ハードルを下げている。しかしながら、複数の MCU ボードの連携機能がなく、オブジェクト指向の概念も導入されていないというデメリットを持つ。

開発環境そのものに焦点を当てた場合、オブジェクト指向環境の実現という観点から、Java 仮想マシンを実行環境として利用することができる。Sun SPOT [16] は Java ランタイム環境を搭載した無線センサネットワークデバイスである。Java ベースのプログラマブルデバイスであり、NetBeans に代表される統合開発環境を活用したプログラミングが可能である。2.4GHz IEEE 802.15.4 無線、バッテリーを搭載している。また、NanoVM [17] は、特定の MCU ボード向けではなく Atmel AVR ATmega8 CPU を対象とした仮想 VM であり、Sun JDK を用いて Java 言語でのプログラミングが可能である。1kByte RAM 中の 75 % を Java プログラムで利用できるほど小型化に成功している。

## 2.3 ユビキタスコンピューティングのためのフレームワーク

ユビキタスコンピューティングを用いたアプリケーション開発を支援するためのフレームワークやミドルウェアの研究 [18] が行われているが、既存研究の多くは、センサ情報やコンテキスト情報といったイベント情報の配布を主な目的としているため、動画や音声の利用が困難である。この問題解決手段として SIP [19] や XMPP [20] の利用が検討されている。SIP は、IP 電話やビデオ会議、IM (Instant Messenger) に利用されているプロトコルであり、情報家電や携帯電話への適用も検討されている。XMPP は Google Talk などで利用されている XML ベースのプロトコルであり、IM によるメッセージ交換やログイン状況といったプレゼンス情報の通知に使われている。XMPP は、現在 IM だけに適用されている規格である一方、通信インフラのバックエンドにも使用されている SIP が効果的であると考えられる。

SIP を利用したフレームワークの研究は、CINMA [21], SPLAT [22] が挙げられる。CINMA はマルチメディアコラボレーションのためのソフトウェア基盤であり、様々な SIP サーバから構成されている。サーバ群は SIP プロキシ、プレゼンスサーバ、レジストラなどから構成されており、CPL (Call Processing Language) などによって制御できる。なお、提供されている C, C++ ライブラリによる独自アプリケーションの開発も可能である。SPLAT は SIP によるマルチメディア会議やゲームといったサービスと既存のアプリケーションを連携させるためのプラットフォームである。SIP の機能を提供するための高レベルの API 群と、その運用を可能にする Client-side SIP service, SIP による独自のサービスを定義した Network Infrastructure Blocks から構成されている。低レベルの API も提供しており、SIP のフローごとに細かな設定も可能である。これらは SIP アプリケーション開発のための有用な API を提供しているが、いずれもクライアント・サーバ型のアーキテクチャを採用しており、ホームネットワークやオフィスでの運用を対象とするためスケーラビリティやアドホック性が低い。また、提供されている API は SIP に詳しい技術者以外での習熟、アプリケーションのプロトタイプ開発が困難である。

### 3. xtel

本章では、まず、開発支援環境としての xtel の特徴について述べる。次に、xtel を構成する 3 つのツールについて、それぞれの特徴および構成を説明する。

#### 3.1 xtel の特徴

本開発支援環境の想定する利用者像は、web デザインを行い、JavaScript、ActionScript などのプログラミング経験があり、ハードウェアプログラミング経験のないデザイナーである。xtel は、このような利用者が実空間アプリケーションとしてのユビキタスコンテンツのアジャイルプロトタイピングを実現可能な開発環境として設計された。具体的には、想定するデザイナーの技術的背景を踏まえ、開発の技術的ハードルを低下させる特徴を持たせ、アジャイルプロトタイピングの実現という観点から、開発スピードを向上させる特徴を持たせる。3 つのツールは、これらの特徴を実現すべく設計されている。

開発支援環境としての xtel は、3 つのツールを連携させることで、ユビキタスコンテンツを構築可能である。たとえば、Talktic を通じてプログラミングされた moxa は実空間の情報をセンシングし、無線通信を通じて別の moxa もしくは PC と通信を行うことができる。次に、PC 上の EC は、P2P ネットワークを通じて moxa から渡されたデータを別の PC へ渡すことができる。あるいは、EntityCollaborator は、P2P ネットワーク上のカメラ情報の値を取得し、その結果を当該 PC に接続された moxa を通じて無線通信経由で別の moxa へ渡すことができる。そして、この moxa から光、音などの出力を行うといった連携も想定できる。xtel を用いることで、実空間上の様々なセンサやアクチュエータだけでなく、インターネット上のサービスまで、あらゆる情報を容易に組み合わせ、個人レベルで新しいユビキタスコンテンツを実現することができる(図 2)。

通常の開発環境と異なり、xtel は分散環境下で実行可能なハードウェアプログラミングから P2P ネットワークプログラミングまでをカバーしている。加えて、moxa により 1 対 n の無線通信が容易に実現可能であること、Talktic により ECMAScript ベースでハードウェアプログラミングを容易に実現可能であること、EntityCollaborator により SIP を用いてのカメラ・音声などの連続情報の通信が容易に実現可能であること、など他の開発環境にはない特徴が存在する。次節より

このような各ツールの特徴について詳細を述べる。

#### 3.2 moxa

##### 3.2.1 特徴

関連研究を踏まえた上で、開発する MCU ボードに対し、以下の 3 つの特徴を採用した。

(1) 実空間で起こる事象のコンピュータへの入出力

センサ・アクチュエータの制御を可能とする。

(2) 実空間におけるコンピュータ同士の協調動作空間的な制作の中に分散的に配置されたコンピュータ同士が通信を行い、協調的な動作を実現する。

(3) コンテンツの実現に必要なソフトウェアの実行

より自由にコンテンツを制作するために、組み込みコンピュータ上でのソフトウェアの実行を可能とする。

##### 3.2.2 構成

moxa は Atmel 社の 8bitRISCMCU を主演算装置とし、32kByte の SRAMIC、デジタル無線トランシーバ IC とアンテナ装置、USB 通信 IC とコネクタ等をプリント基板上に搭載した組み込みボードである。外形は 67mm x 34mm x 5mm とコンテンツへの組み込みに適した、十分に小さい実装容積となっている(図 3)(表 1)。

moxa はセンサを利用し、実空間の事象を情報として入力する端子を供えている。すなわち、デジタル入力端子、アナログ入力端子、有線通信による入力端子である。デジタル入力端子、アナログ入力端子はいずれもセンサからの電気信号を受け取る端子であり、それぞれ 8 系統を具備し、最大で 16 系統のセンサが接続できる。有線通信による入力端子は I2C 通信、RS232C 通信がそれぞれ 1 系統ずつ利用可能であり、I2C 通信を利用した場合はより多くのセンサが接続可能である。また、moxa はアクチュエータを利用し、情報を実空間の事象として出力する端子を備えている。これらの端子は入力端子と共通であり、端子毎、入力と出力を切り替えて排他的に利用することができる。

moxa は近距離の無線通信機能を搭載している。2.4GHz 帯を利用する IEEE802.15.4 規格適合の無線トランシーバにより、見通し 100m 程度の距離で最大 100kbps での通信なメッシュネットワークを構成するだけでなく、チャンネル指定が可能である<sup>(注1)</sup>。こうし

(注1): Arduino に Xbee[23] を搭載することで無線機能は実現できるものの、チャンネル指定は不可能である。

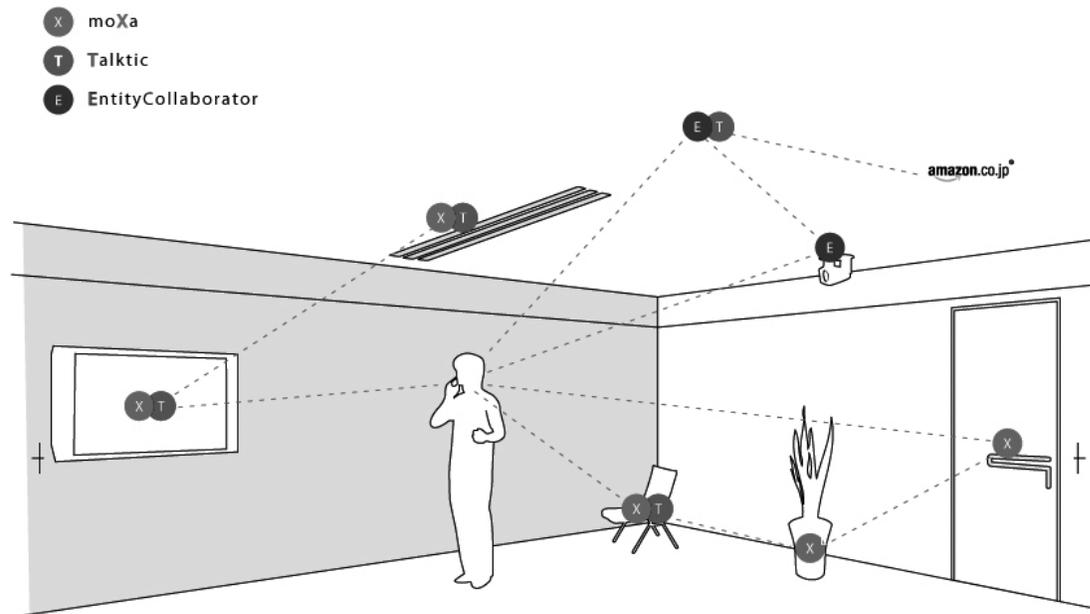


図 2 xtel を利用したユビキタスコンテンツイメージ  
Fig.2 Image of xtel-based Ubiquitous Contents

表 1 moxa スペック表  
Table 1 Specification of moxa

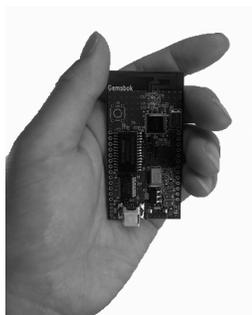


図 3 moxa  
Fig.3 moxa

CPU	8bit RISC CPU ( Atmel ATmega128L )
動作周波数	8MHz
SRAM	32kByte
EEPROM	4K
無線通信	2.4GHz 帯 IEEE802.15.4 規格
外部インターフェース	USB, RS232C, I2C
デジタル入出力端子	8 端子
アナログ入出力端子	8 端子
外径寸法	67mm x 34mm x 5mm

た近距離無線通信機能を有する事で, moxa 同士がセンサ情報を交換し, アクチュエータの制御信号を共有することが可能となり, コンピュータ同士を連携させてコンテンツを制作する事ができる.

moxa は単体でソフトウェアの実行が可能な組み込みコンピュータである. ソフトウェアはコンピュータ上で C 言語を用いて記述され, moxa 用にクロスコンパイルされた実行コードを, USB ケーブルを用いて moxa 上に転送・保存することで任意のソフトウェアを実行することが出来る仕組みを持つ. 開発者は, デ

バイス制御ライブラリ, 無線通信ライブラリ, シリアル通信ライブラリ, リソース管理ライブラリで構成されるソフトウェアライブラリを利用することにより, ハードウェアの機能を容易に利用することができる. しかしながら, C 言語での開発はデザイナーにとってハードルが高いことため, web 開発において馴染み深い ECMA Script に準拠した開発言語を利用できるように Talktic を実装した.

### 3.3 Talktic

#### 3.3.1 特 徴

関連研究を踏まえ, MCU ボード制御のためのプラットフォーム設計にあたり, 以下の 4 つの特徴を採用した.



図 4 Talktic プラットフォームの構成  
Fig. 4 Structure of the Talktic platform

#### (1) 仮想マシンを利用した実行環境

MCU ボードを利用する上で問題となる特定の MCU への依存を排除するために、仮想マシンを利用したバイトコードによるランタイム環境を採用する。

#### (2) ライトウェイトランゲージによるプログラミング環境

オブジェクト指向<sup>(注2)</sup>での記述が可能であり、記述の際に負荷が少ないライトウェイトランゲージを採用する。また独自の言語ではなく、既に普及しているものを選択する。

#### (3) ライブラリによる要素技術の隠蔽

他のデバイスとの通信やセンサ・アクチュエータの取り扱いを、ある程度抽象化した上で隠蔽するライブラリを用意する。

#### (4) 実行規約の明確化

実装規約はドキュメントなどに明確に定めると同時に、プログラミング環境の一部として特定のクラスあるいはメソッドとして実装する。

### 3.3.2 構成

Talktic プラットフォームの構成は実装コンポーネント単位で以下のように分かれている (図 4)。この各実装コンポーネントの組み合わせにより、Talktic プラットフォームとしての各機能が実現される。

Talktic 仮想マシンは NJS JavaScript Interpreter をベースに実装した。このプログラムは JavaScript のコンパイラ、アセンブラ、仮想マシン、ネイティ

(注2): オブジェクト指向言語を採用したことは、ソースコードの再利用性と保守性を高め、開発や保守のコストを低下させる狙いに基づく。また、デザインパタンの導入により、開発ハードルの低下が実現可能となる狙いに基づく。

表 2 代表的なネイティブ API  
Table 2 Representative Native APIs

カテゴリ	API	用途
標準入出力	(なし)	エラー情報, デバッグ
デジタル IO	pinMode	IO ピンの初期化
	digitalWrite	状態設定
	digitalRead	状態読み取り
アナログ IO	analogWrite	PWM 出力
	analogRead	ADC 変換
	soundWrite	指定周波数の PWM 出力
無線通信	radioConnect	無線通信を開始
	radioSend	無線パケットの送信
	onRadioReceive	無線パケットの受信ハンドラ
	radioClose	無線通信を終了
シリアル通信	serialInit	シリアル通信を初期化
	serialAvailable	受信データがあるか
	serialRead	データ受信 (同期)
	onSerialReceive	データ受信ハンドラ (非同期)
	serialSend	データ出力
TCP/IP 通信	tcpipConnect	TCP/IP 通信を初期化
	tcpipAvailable	受信パケットがあるか
	tcpipRead	パケット受信 (同期)
	onTcpipReceive	パケット受信ハンドラ
	tcpipSend	パケット送信
HTTP 通信	tcpipClose	TCP/IP 通信を切断
	httpLoad	GET/POST メソッド
	onHttpRequest	データ受信ハンドラ
	onHttpRequest	データ受信ハンドラ

ブメソッドとクラスを LGPL ライセンスで提供している [24]。Talktic 仮想マシンはこの NJS JavaScript Interpreter 0.2.5 をもとに MCU 上でのランタイム環境に合わせるために様々な改変と修正を加えており、全て C 言語により記述されている。この Talktic 仮想マシンはシングルスレッドで動作し、ガベージコレクションによる動的なメモリ管理、浮動小数点のサポート、仮想割り込みを実装している。

Talktic プラットフォームのコアは、Talktic スクリプトである。Talktic スクリプトは国際標準である Standard ECMA-262 3rd Edition に定める ECMA Script Version 3 にほぼ準拠する [25]。類似の言語として、ActionScript, JavaScript がある。Talktic スクリプトの ECMA スクリプトとしての一般的な特徴として、プロトタイプベースのオブジェクト指向、匿名関数・匿名オブジェクト、標準的な制御構文が挙げられる。ただし Talktic スクリプトは現在の言語仕様として、スコープチェーン、クロージャ、ネイティブクラスのプロトタイプ変更が実装されていない。これは主にベースとなる NJS JavaScript Interpreter の実装に依存している。

Talktic のライブラリは 2 つの側面を持つ。第 1 に、

各 MCU ボードが提供する基本的な入出力と通信機能のドライバとしての役割とともに Talktic スクリプトからの操作を可能にするためのネイティブ API を提供するライブラリがある (表 2) . 第 2 に, Talktic スクリプトのオブジェクト指向の各種デザインパタンによる実装を可能にするための中間ライブラリがある. ネイティブ API はシンプルであり, そのまま利用しても問題ない. 一方で, これらをラップし, デザインパタンとして利用することでスムーズな実装が可能となる.

### 3.4 EntityCollaborator

#### 3.4.1 特徴

関連研究を踏まえ, P2P フレームワーク設計にあたり, 以下の 2 つの特徴を採用した.

##### (1) 離散的・連続的な情報の統合的利用

離散的情報とは, センサ情報から生成されるイベントやコンテキスト, IM といったテキストベースの情報を意味し, 連続的情報とは IP 電話における音声通話やビデオチャットによる動画などのストリーミング情報を意味する. これらの情報を統合的に扱うために, SIP を利用する. イベント駆動型の API を用い, セッション管理の隠蔽とメソッドの抽象化を行うことで, 習熟コストを軽減させる.

(2) アドホック性およびスケーラビリティの確保  
ノード管理において DHT(Distributed Hash Table) を採用することで, アドホック性およびスケーラビリティを実現する. DHT とは, 分散環境中でハッシュテーブルを構築することにより, 情報の分散とデータ探索機能を提供する技術である. ハッシュテーブルの管理を行うのみであるため, 学習が容易である. DHT の特徴は, サーバが存在しないピア P2P であること, 高いスケーラビリティやアドホック性を持つこと, 単一障害点の回避により, ネットワーク中に存在するオブジェクトは高確率で発見できることが挙げられる.

#### 3.4.2 構成

EC は前節 2 点の特徴を持つフレームワークである. ユーザは提供されたライブラリを利用することで離散的情報と連続的な情報を統合的に扱う SIP アプリケーションの開発が可能である. 実装には Java(JDK 5.0) を用いている.

図 5 に EntityCollaborator のシステム構成を示す. システムは 5 つのモジュールで構成されている. 具体的には, EC のシステムはユーザが実装するコンポーネントである Entity, Entity を保持するコンテナであ

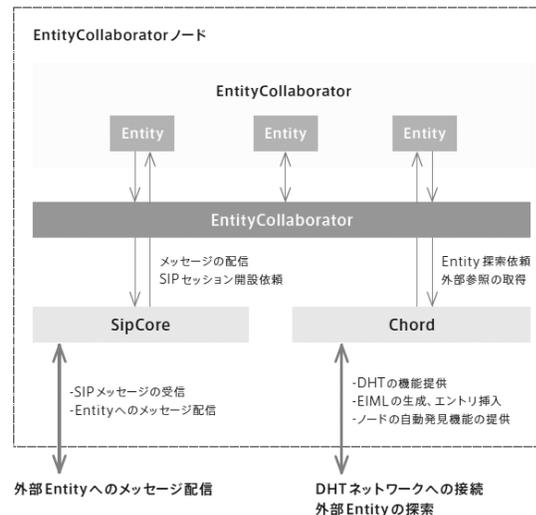


図 5 EntityCollaborator システム構成  
Fig. 5 Structure of EntityCollaborator

る EntityContainer, SIP メッセージの処理モジュールである SipCore, DHT の機能を提供するモジュールである Chord, システムのフロントエンドである EntityCollaborator である. 以下では, それぞれについて説明を行う.

Entity はアプリケーション開発の中心的な要素であり, インタフェースとして定義されている. ユーザは任意の機能を持った Entity を抽象クラスである AbstractEntity を継承することで開発し, それらを組み合わせることでアプリケーションの構成を行うことができる. Entity には SIP による通信を隠蔽, 抽象化したメソッドが定義されており, それによりイベント駆動型のプログラミングが可能である. 表 3 に各 SIP メソッドに対応したメソッドとコールバックメソッドであるイベントハンドラを示す.

Entity は SipCore に対して処理を委譲することで SIP メッセージの送信を行い, 外部からの SIP メッセージを受信することで SIP URI で指定された Entity に対応したイベントハンドラを呼び出す. そのため SIP プロキシのような機能を持っているといえる.

筆者らは DHT を利用した探索方法として, Entity に対して任意のキーワードを付与し, そのキーワードと対応した Entity のメタ情報を記載した XML を DHT のエントリとする手法を考案した. メタ情報を記載した XML である EIML(Entity Information Markup Language) は, Entity の SIP-URI や各種パ

表 3 EntityCollaborator で使用される SIP メソッド  
Table 3 SIP Methods in EntityCollaborator

SIP メソッド	対応するメソッド, ハンドラ
INVITE	sendOffer(), receiveOffer(), receiveAnswer()
BYE	sendBye(), receiveBye()
MESSAGE	sendMessage(), receiveMessage ()
SUBSCRIBE	subscribe(), receiveScribe()
NOTIFY	sendNotify(), receiveNotify()
REGISTE	- ( Entity にメソッドは定義されていない)

ラメータなどを記述したものであり、システムによって自動的に生成される。Chord はそれらのアルゴリズムを実装したクラスである。

EntityCollaborator はシステムのフロントエンドとして機能するクラスであり、システムの起動や Entity の探索のためのメソッドを持つ。システムの起動は基本的に SipCore の初期化、Chord の初期化、ブートストラップノードの指定、自動発見による DHT のネットワークへの参加、Entity の初期化と追加の順に行う。

#### 4. xtel を用いたユビキタスコンテンツ

xtel を用いて構築された数多くのユビキタスコンテンツの中から 3 つのコンテンツをサンプルコンテンツとして紹介する。第 1 に moxa, Talktic を用いて開発された傘状の遊具「雨刀(あまがたな) [26]」、第 2 に moxa, Talktic を用いて開発された新しい遊具「SoundCandy [27]」、第 3 に EntityCollaborator を用いて開発された時空間共有のためのシステム SpaceTracer [28] である。雨刀は視覚的ユビキタスコンテンツ、SoundCandy は行動的ユビキタスコンテンツ、SpaceTracer は時間的ユビキタスコンテンツに分類できる。

雨刀(図 6)は、雨上がりの晴れ晴れとした気分を存分に楽しむための傘状の遊具である。外装は平凡なビニール傘で、雨の日にはそのまま傘としても使用できる。一方、晴れたときには傘を閉じ振り回すことで、刀と刀が衝突する音が耳へと鳴り響く遊具に様変わりする。この音は振り方に応じて 3 種類に変化し、また巧みに振り捌くことで 5 種類の必殺技を繰り出すことができる。これらにより、雨刀のユーザは映画やゲームの主人公のような体験を平凡な日常のうちに体験することができる。

雨刀は、xtel のうち moxa, Talktic を用いて開発された。具体的には、Moxa を 2 枚利用し、一方の moxa は加速度センサを接続し、ビニール傘に埋め込み、無



図 6 雨 刀  
Fig. 6 Amagatana



図 7 SoundCandy  
Fig. 7 SoundCandy

線通信を通じてセンサからの値を他方の moxa へ送信している。他方の moxa は PC に接続され、無線通信を通じて加速度センサからの値を受信し、PC 内のアプリケーションへ送信している。これらの moxa 上のプログラムは Talktic を用いて記述され、実行されている。

SoundCandy (図 7) は、どんな場所でも身の回りの音と動きを利用して、自分たちだけの新しい遊びを作り出すことができる新しい遊具としてデザインされている。従来、遊びは空き地や路地などにおいて、自分たちでさまざまに工夫し、その場を自分たちの遊び場に変えることで行われていた。SoundCandy はそのような遊びの経験を作り出す新たな遊具として位置づけられる。

なお、雨刀と SoundCandy は、コンテンツ同士での



図 8 SpaceTracer  
Fig.8 SpaceTracer

連動が可能である。SoundCandy でユーザが録音した音声は、雨刀の効果音として再利用される。また、雨刀を遊ぶことによって得られたユーザの動きの大きさをもとに、SoundCandy のエフェクトの強弱をその人に合わせて調節する。このコンテンツ間の連動によって、街の中での遊びに一層の深みと広がりが生まれる。

SpaceTracer (図 8) は、プライバシーに配慮しつつ、離れた空間の時間軸を含めた状況の共有によるウェアネスを生むためのシステムである。ネットワークカメラから毎分取得された時空間画像 30 分間をアルファブレンディングにより重ね、さらにその場のイベント発生に合わせたフィルタリング処理を施した映像として合成している。静止画を重ね合わせることで、人物の顔のような細かいものは見えなくなる一方、その場の様子は十分に伝えることができる。また、ある程度の時間分を圧縮して一枚で表示することで、その時間分の状況を一目で確認することができる。

## 5. 評価

本章では、3 度に渡って開催されたワークショップを通じて、実際に xtel を使用してプロトタイピングを行った参加者の意見をもとに xtel の評価を行う。それぞれのワークショップでは、まず xtel について簡単に説明を行ったのち、サンプルプログラムを用いてプログラミングを体験してもらった。その後、チームを決定し、任意のテーマに従ってブレインストーミングを行い、プロトタイプを実装した。ワークショップによって、ブレインストーミングの際、いくつかのセンサ、アクチュエータ、布・粘土・紙などのマテリアル類、家電製品などが配布された。以下ではそれぞれのワー

クショップの概要、タイムテーブル、テーマ、結果について述べたのち、質的評価の結果について述べる。

2007 年 9 月、筆者らは、完成した xtel の最初のバージョンを一般の方にも評価してもらうため、慶應義塾大学 G-SEC Lab にてワークショップを開催した。本ワークショップでは、現在の moxa よりも容積が一回り大きい proto01 ボード、現在のバージョンと比べていくつかのメソッドが少ない Talktic、EntityCollaborator の 3 つのシステムを用いた。すでに研究関係者の中では数回のワークショップを通して利用感の確認を行い、改良を施した状態となっていた。参加者は、ゲーム会社の方や、インタラクション設計で著名なプロダクトデザイナー 20 名であった。テーマは「目の前にある素材を組み合わせ、新しい製品やサービスのコンセプトを体験することができるようなプロトタイプの制作」であった。

タイムテーブルは、全体約 6 時間の構成であった。前半 2 時間で xtel の概要およびシステム構成について説明し、参加者に簡単なプログラミングを体験してもらった。休憩後、後半 2.5 時間で、チーム決定ののち、ブレインストーミング、プロトタイプの実装、発表準備までを行った。残りの 1 時間弱で発表および講評を行った。実装後の発表では、2 人で同じタイミングで腰を振ることで親密になれるエクササイズ器具、3 人が呼吸を合わせて特殊なコントローラを操作することで PC の画面の色を狙った色に近づけるゲーム、団扇を仰ぐ行為に合わせて動き出す扇風機などの魅力的な商品プロトタイプ群が提案された。

同じく 2007 年 9 月に、デザイン誌『AXIS』の主宰した第 2 回金の卵学校選抜オールスターデザインショーケース<sup>(注3)</sup>において xtel を用いたデザインワークショップを行う機会を得た。参加者は、ショーケースに参加したデザイン学校の学生や企業のデザイナー 14 名であったが、参加者の多くは既存の製品のプロトタイピング過程についてはすでに親しんでいたものの、プログラムや電子回路などに直接触れるのが初めてであった。テーマは「新しい家電製品のコンセプトのプロトタイピング」であった。

タイムテーブルは、主催者側の依頼で全体約 2 時間という構成であった。したがって、プログラムの詳細なチュートリアルを省略し、各チームにスタッフを技

(注3): <http://www.axisinc.co.jp/publishing/exhibition/200708.html>

術サポートとして参加させ、コンパクト化を図った。ワークショップ開始直後に xtel の簡単な説明を行い、実働するプロトタイプを用いてセンサやアクチュエータの活用方法を具体的に説明した。その後、チームを決定し、ブレインストーミング、プロトタイプの実装ののち、発表および講評を行った。実装後の発表では、座ることで自動的に体重を量りさりげなくダイエットを支援する椅子、複数の扇風機が向かい合った時のみ動きを止め電力を節約するシステム、複数の家電製品がドミノのように連動して楽しめる仕掛けなどが提案された。

なお、実装段階にて参加者であるデザイナーは、それぞれの作品のフォルムや機能を総合的に考えながら、技術スタッフと協力して作品を作り上げていた。通常の製品開発プロセスでは、機能部品が完全に設計されたのちにデザイナーへ渡されるケース、あるいは、先にデザインを決定して、製品の回路や機構をデザインにあわせて設計するケースが多数である。この場合、開発フローは不可逆かつ一意的な流れに基づくため、プロトタイピングを繰り返し製品の質を高めるというアプローチは実質的に困難である。これに対して、アジャイルプロトタイプ開発支援環境としての xtel を利用することで、製品の機構や機能、フォルムや製品コンセプトのデザインを同時に行うことができる。参加者のデザイナーの方々からも、非常に貴重な経験ができたとの意見をいただいた。

2009 年 11 月に行ったワークショップは、RT 開発支援セミナー「オープンソースハードウェアで広がるロボットビジネス<sup>(注4)</sup>」の一部として開催された。このワークショップでは、小型化された現在の moxa、現在のバージョンの Talktic を用いた。これまでのワークショップと異なり、moxa が小型化されていることで、より柔軟な実装が可能になった。参加者は、既存のウェブサービスからハードウェアとネットが融合した新しいサービスへの展開を狙う地元の意欲的な企業の方々13名であった。テーマは「身体を使って楽しく健康になるものやサービス」という身近なテーマを用意した。

タイムテーブルは、全体約4時間の構成であった。前半2時間で xtel の概要およびシステム構成について説明し、チーム決定、および、ブレインストーミング

を行った。続いて30分で、プログラムに関するチュートリアルを行った。すでに実現するものがイメージできている段階でのチュートリアルを行うことで、理解を促進する効果を狙った。後半1.5時間で、プロトタイプの実装、発表および講評を行った。実装後の発表では、加速度センサを用いて見えないボールをキャッチボールするボールとグローブのセット、手や頭などいろいろな場所をタッチすることで反応する人形、通勤時につま先立ちの姿勢を維持させるために踵が地面につくとアラームが鳴る靴など、非常にユニークな作品が並んだ。

これらのワークショップのような2-4時間程度の短い時間でも、xtel を用いたアジャイルプロトタイプの開発を体験することができる。従来の製品企画では実体験まで持ち込むことが不可能だった製品コンセプトも、インタラクションのスケッチを行うツールとして xtel を用いることで、短時間で実装し、具体的な体験として検証することが可能になる。また、プログラマーとデザイナーが入り混じって迅速なプロトタイピングを行うことで、アジャイルな web 開発のプロセスをプロダクト開発へと持ち込むことができるだけでなく、商品開発などの経験のないメンバでもプロトタイピングに参加し、貢献することが可能になる。これらのワークショップの結果から、分散環境での実空間アプリケーションを開発する際、技術的ハードルの低下、および、開発のスピードの向上を実現するために xtel は有効な環境であることがわかる。

最後に、参加者の個別の意見を表4に示す。これらの意見から、開発の技術的ハードルの低下、および、開発のスピードの向上について一定の評価を得ていることがわかる。しかしながら、機能面において、グラフィカルな表現を実現するための機能として Flash や Processing などとの連携機能を強化すべきとの意見を得た。また、Java プログラミングおよびネットワークプログラミング経験のないユーザにとって、EntityCollaborator を用いた開発は難解であるとの意見も目立った。この点については Flash や Processing 上で動作するラッパークラスを提供することで解決したい。

## 6. ま と め

本論文では、ユビキタスコンテンツ開発支援環境 xtel を提案した。xtel は、分散環境での実空間アプリケーションとしてのユビキタスコンテンツの開発ス

(注4): [http://www.sansokan.jp/events/eve\\_detail\\_san?H\\_A\\_NO=09920](http://www.sansokan.jp/events/eve_detail_san?H_A_NO=09920)

表 4 ワークショップ参加者の自由意見  
Table 4 Free Opinion from Attendee of the Workshops

肯定的な意見
web との連携が面白い 無線が容易に使える . 既存技術でできたことがより簡単にできるようになった . 初心者になじみやすい . 越えなければならぬハードルがクリアされていて思考を直接反映できる . 無線が分かりやすいのが良い . プログラムの簡易さがうれしい . 初心者にとってこのようなシステムがあると導入しやすい .
否定的な意見
グラフィカルな表現を行なおうとすると Flash などとの連携が必要となるので、これが同じ環境へ組み込まれると幅が広がる . Processing のように簡単にビジュアルなものが作れないので、プログラムに馴染みのない人は使わないと思われる . Flash にラッピングしてくれるようなライブラリがあるとうれしい . もっとサンプルを用意してほしい .

ビードを向上だけでなく、技術的ハードルの低下を目的としている . xtel は 3 つのツールで構成され、それぞれがこの目的を達成するための特徴を持つ . moxa では、センサ・アクチュエータの制御、および、実空間におけるコンピュータ同士の協調動作を実現し、コンテンツの実行に必要なソフトウェアの実行を可能とした . Talktic では、デザイナーにとって馴染み深い ECMAScript を採用した . EntityCollaborator では、離散情報に加えて連続情報を P2P ネットワークで扱えるだけでなく、学習コストを軽減させる仕様を採用した .

今後の開発方針として、グラフィックス機能の連携を含めた開発プロセスのさらなる簡易化を行う . 現状、開発環境は Eclipse を利用し、JavaScript、Java ベースでの開発を必要としている . しかしながら、特に Java を用いた開発はデザイナーにとって依然としてハードルが高いとの意見が目立った . 今後は、Flash、Processing、Max/MSP などでも利用可能なラッパークラスを提供し、デザイナー側でのプログラミング部分を隠蔽する . これにより、多くのユーザがプロトタイプングに際し、xtel を利用可能となる . 同時に、これらのラッパークラスを提供することは、これらの開発環境におけるグラフィックス機能との連携が必然的に可能となることを意味する . なお、moxa、Talktic、EntityCollaborator のいずれも LGPL ライセンスにてソースコードを公開している [29] .

## 7. 謝 辞

本研究は、科学技術振興機構 (JST) 戦略的創造研究推進事業 (CREST) の支援によって行われた .

### 文 献

- [1] Inakage, Masa, et al. "Designing Ubiquitous Content for Daily Lifestyle." International Journal of Cognitive Informatics and Natural Intelligence (IJCINI) 5.1 (2008): 35-40.
- [2] Ishii, Hiroshi, and Brygg Ullmer. "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms." The SIGCHI conference on Human factors in computing systems. Atlanta, Georgia, 1997. 234 - 41
- [3] 生井みづき, et al. "Panavi: センサ・アクチュエータ・無線通信機能を内蔵するフライパンを中心とした料理スキルの習得を支援するシステム." 電子情報通信学会 マルチメディア・仮想環境基礎研究会. 大阪, 2009. 41-46.
- [4] Shibata, Itsuki, and Naohito Okude. "Biblioroll." The 8th International Conference of Ubiquitous Computing (UbiComp). Orange County, California, 2006.
- [5] Kuribayashi, Satoshi, et al. "Plantio: An Interactive Pot to Augment Plants' Expressions." International Conference on Advances in Computer Entertainment Technology. Salzburg, 2007. 139-42.
- [6] Uchida, Yu. "Kage No Sekai: Interactive Animation of Shadow Based on Physical Action." International Conference on Advances in Computer Entertainment Technology(ACE). Salzburg, 2007. 274-75.
- [7] Uriu, Daisuke, et al. "Caraclock: An Interactive Photo Viewer Designed for Family Memories." Ed. The 27th international conference extended abstracts on Human factors in computing systems. Boston, MA, USA 2009. 3205-10.
- [8] Ozaki, Fumitaka, et al. "Myglobe: Cognitive Map as Communication Media." SIGGRAPH '09: Posters New Orleans, Louisiana 2009.
- [9] Polastre, Joseph, et al. "The Mote Revolution: Low Power Wireless Sensor Network Devices." Proceedings of Hot Chips 16: A Symposium on High Performance Chips., 2004.
- [10] Beigl, Michael, and Hans Gellersen. "Smart-Its: An Embedded Platform for Smart Objects." Smart Objects Conference (SOC 2003). Grenoble, France, 2003.
- [11] Greenberg, Saul, and Chester Fitchett. "Phidgets: Easy Development of Physical Interfaces through Physical Widgets." Symposium on User Interface Software and Technology(UIST). Orlando, Florida, 2001. 209 - 18.
- [12] Gainer.cc. "Gainer". 2006. 21 December 2009. <<http://gainer.cc/Main/HomePage?userlang=ja>>.
- [13] Barragan, Hernando. "Wiring". 2006. 21 December 2009. <<http://barraganstudio.com/>>.

- [14] Arduino. "Arduino". 2005. 21 December 2009. <<http://www.arduino.cc/>>.
- [15] Reas, Casey, and Benjamin Fry. "Processing". 2001. 30 June 2008. <<http://processing.org/>>.
- [16] Simon, Doug, et al. "Java on the Bare Metal of Wireless Sensor Devices: The Squawk Java Virtual Machine." ACM/Usenix International Conference On Virtual Execution Environments. Ottawa, Ontario, Canada, 2006. 78 - 88
- [17] Harbaum, Till. "Nanovm". 2005. 21 December 2009. <<http://www.harbaum.org/till/nanovm/index.shtml>>.
- [18] Endres, Christoph, Andreas Butz, and Asa MacWilliams. "A Survey of Software Infrastructures and Frameworks for Ubiquitous Computing." Mobile Information System 1.1 (2005): 41 - 80.
- [19] J. Rosenberg, et al. "Sip: Session Initiation Protocol". 2002. <<http://www.ietf.org/rfc/rfc3261.txt>>.
- [20] Jabber Software Foundation. Extensible Messaging and Presence Protocol (Xmpp): Core, 2004.
- [21] Berger, Stefan, et al. "Ubiquitous Computing Using Sip." International Workshop on Network and Operating System Support for Digital Audio and Video. Monterey, CA, USA, 2003. 82 - 89.
- [22] Singh, Aameek, et al. "Splat: A Unified Sip Services Platform for Voip Applications: Research Articles." International Journal of Communication Systems 19.4 (2006): 425 - 44.
- [23] MaxStream. "Xbee/Xbee-Pro". 2007. 21 December 2009. <<http://www.digi.com/products/wireless/point-multipoint/>>.
- [24] New Generation Software. "Njs Javascript Interpreter." 1998. 21 December 2009. <<http://sourceforge.net/projects/njs/>>.
- [25] ECMA International. "Standard Ecma-262, 3rd Edition." 1999. 21 December 2009. <<http://www.ecma-international.org/publications/standards/Ecma-262.htm>>.
- [26] Katsumoto, Yuichiro, and Masa Inakage. "Amagatana." The 15th international conference on Multimedia. Augsburg, Germany 2007. 361-62
- [27] Ishibashi, Shuichi, Daisuke Uriu, and Naohito Okude. "Sound Candy." The international conference on Advances in computer entertainment technology(ACE). Salzburg, 2007.
- [28] 橋本悟, and 中西泰人. "Spacetracer: ネットワークカメラ画像の合成による空間型コミュニケーションシステムの提案." 情報処理学会シンポジウム インタラクシオン 2006. 東京, 2006. 181-82.
- [29] Ubiquitous Content Project. "Xtel Technical Information". 2008. 21 December 2009. <<http://uc.sfc.keio.ac.jp/xtel/>>.

(平成 xx 年 xx 月 xx 日受付)

## 徳久 悟

2002 慶應義塾大学法学部政治学科卒業 .  
2004 同大学大学院政策・メディア研究科  
修士課程修了 . 2007 同大学大学院政策・メ  
ディア研究科博士課程修了 . 現在慶應義塾  
大学大学院メディアデザイン研究科 講師兼  
しさを媒介としたインタラクシオンデザイ  
ンの研究に従事 . 情報処理学会会員 , VR 学会会員 .

## 石澤 太祥 (正員)

2001 慶應義塾大学環境情報学部卒業 .  
2003 同大学大学院 政策・メディア研究科  
修士課程修了 . 2006 同大学大学院 政策・  
メディア研究科 博士課程入学 . 情報処理学  
会会員 .

## 丹羽 義将

2005 慶應義塾大学環境情報学部卒 . 2007  
同大学大学院政策メディア研究科修士課程  
了 . 点スイッチ有限会社を共同設立 , 同社  
代表取締役 .

## 粕谷 貴司

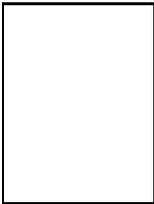
2005 東京農工大学情報コミュニケーション  
工学科卒 . 2007 慶應義塾大学大学院政  
策メディア研究科修士課程了 . 2008 株式  
会社竹中工務店入社 . ワークプレイス、お  
よびデータセンターの IT コンサルタント  
業務に従事 .

## 勝本雄一朗

2004 慶應義塾大学環境情報学部卒業 ,  
2006 同大学大学院政策・メディア研究科  
修士課程修了 . 現在同大学大学院政策・メ  
ディア研究科後期博士課程在学中 .

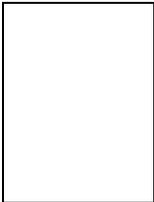
## 石橋 秀一

2004 武蔵野美術大学造形学部卒 . 2008  
慶應義塾大学大学院政策メディア研究科  
修士課程了 . 2008 同大学院メディアデザ  
イン研究科後期博士課程入学 .



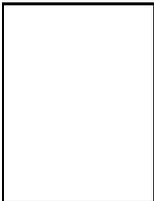
橋本 悟

2006 慶應義塾大学環境情報学部卒 . 2008 同大学大学院政策・メディア研究科修士課程 中途退学 . 現在ベンチャー企業にて携帯電話向け技術開発に従事 .



植木 淳朗

2001 東京大学経済学部経済学科卒業 . 2003 慶應義塾大学大学院政策・メディア研究科 修士課程修了 . 2007 同大学大学院政策・メディア研究科 後期博士課程単位取得退学 . 2009 慶應義塾大学大学院メディアデザイン研究科 講師 . 情報処理学会会員 .



稲蔭 正彦

1982 米国・オベルリン大学卒 . 1983 米国・カリフォルニア芸術工芸大学大学院芸術修士修了 . 1990 メディアスタジオ株式会社代表取締役 . 1999 慶應義塾大学環境情報学部教授 . 2008 慶應義塾大学大学院メディアデザイン研究科委員長兼教授 . ACM SIGGRAPH 会員 . Visual Effects Society 会員 .

**Abstract** This paper describes the "xtel" development environment for "Ubiquitous Content". Ubiquitous contents are real space applications that are embedded in day-to-day life and intended for use by consumers. This is content that is experienced through interaction with people, objects and environments that exist in real space. Xtel comprises three tools: the "moxa" MCU board that connects to sensors and actuators and is capable of short-distance wireless communications; the "Talktic" programming/runtime environment for the MCU board that contains a JavaScript parser, compiler, VM and library; and the "EntityCollaborator" P2P network library that is capable of handling continuous information such as video and audio in addition to the discrete information from sensors. The authors had a common purpose to make development easier and to accelerate development faster when designer or engineer use this development environment in order to develop ubiquitous contents. After introducing three ubiquitous contents developed using xtel as sample contents, this paper evaluates xtel as tool through some workshops.

**Key words** Ubiquitous Content , xtel, Agile Prototyping , MCU, VM, P2P , Development Environment