Vol.2010-SLDM-144 No.22
Vol.2010-EMB-16 No.22
Vol.2010-MBL-53 No.22
Vol.2010-UBI-25 No.22
2010/3/26

携帯電話を用いた異種ネットワークデバイス連携システムの開発

加藤 悠一郎[†1] 峰野 博史[†2] 角野 宏光[†3]
石川 憲洋[†3] 水野 忠則[†4]

この論文では，異種ネットワークデバイス連携システムの開発について記述する．ユビキタスネットワーク社会では，様々なデバイスがネットワークに繋がることが想定される．しかし，様々な通信プロトコルの違うネットワークが混在しており，各デバイスが通信することができない．そこで，異種ネットワーク機器のデバイスを統合し，機器間で連携をとれるシステムの開発を行った．また，デバイスの情報の参照や，機器間の連携設定を行う機能を携帯端末に実装した．

The cooperative system using heterogeneous devices

Yuichiro KATO,[†1] Hiroshi MINENO,[†2]
Hiromitsu SUMINO,[†3] Norihiro ISHIKAWA[†3]
and Tadanori MIZUNO[†4]

In this paper, the system which enables flexible cooperative behavior among the devices belonging to different networks. There are a lot of devices connected to a network in the ubiquitous networking society. However many kinds of networks coexist, and they use different protocols. Threfore, We developed the system which intergrates the devices belonging to different networks. With this system, the cooperative behaivior among these device is estabieshed.

†1 静岡大学大学院情報学研究科
Graduate School of Informatics, Shizuoka University
†2 静岡大学情報学
Faculty of Informatics, Shizuoka University
†3 NTT ドコモ サービス＆ソリューション開発部
Service & Solution Development Department, NTT DOCOMO, Inc.
†4 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University

1. はじめに

近年，IrDA，Bluetooth，無線 LAN，Zigbee などの短距離無線技術が発展している．それに伴い，携帯電話や PC はもちろん，情報家電と呼ばれるネットワーク機能搭載の製品も発売され，ホームネットワークやホームオートメーションが注目を集めている．既に，家の状態を外出先から確認するようなサービスも出てきている．しかし各機器は，DLNA，ECHONET，IEEE1934 など様々なプロトコルを利用しており，相互に通信できないことが多い．

センサ技術においても同様に，個々のセンサが無線技術を使いネットワークを形成するのが一般的になっている．様々な情報を得ることのできるセンサネットワーク技術は，様々な分野で注目を集めている．ヘルスケア，ホームセキュリティ，環境モニタリングにおいては，温度，圧力，ガス濃度，振動などを検知できるセンサをさまざまな場所に設置して，得られるセンシング情報を基にサービスの提供を行う．そのため，環境モニタリングや災害防止ではモニタリング対象の場所に，またホームセキュリティでは家にセンサが設置されることになる．今後これらの研究の進展とともに，私たちの身の周りには様々なセンサ機器が常に存在する環境が考えられる．しかし，現在の研究では個々のセンサネットワークは，省電力や通信性能などを考慮して独自の通信・制御プロトコルをつかっている．そのため，災害防止用に設置されたセンサは災害防止以外の用途で利用することはできない．またホームセキュリティ用に設置されたセンサも，ベンダーが提供するサービスのために利用されるだけである．もし，それぞれのセンサネットワークを統一的に扱うことができれば，様々な用途のために設置されたセンサを別の目的にも使う，また扱えるセンサの数も増えることになるので，信頼性の向上，応用アプリケーションの開発などにより，質の高いサービスが可能になると考える．

このような環境では，様々なデバイスを統一的に扱うことでサービスの幅を広げることができる．ホームセキュリティサービスを行うために専用のデバイスを機器しなくても，既に家にある機器やセンサを組み合わせることで同じサービスが実現できるだろう．また，ヘルスケア用のセンサと情報家電を組み合わせるなど，柔軟なサービスが実現できる．

そこで，我々は携帯を用いて異種ネットワークの機器間の連携を設定できるシステムの開発を行った．我々のシステムでは，DLNA や ECHONET へのプロトコル変換を行うので

Vol.2010-SLDM-144 No.22
Vol.2010-EMB-16 No.22
Vol.2010-MBL-53 No.22
Vol.2010-UBI-25 No.22
2010/3/26

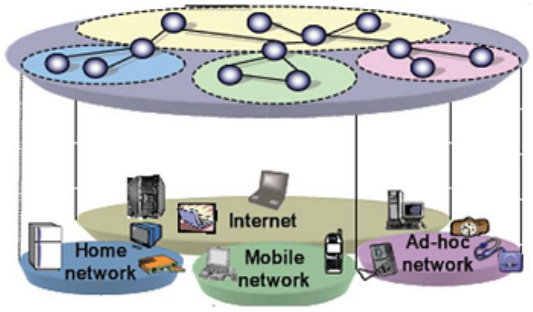

図 1　PUCC アーキテクチャ

はなく，PUCC(Peer-to-Peer Computing Consortium) プロトコルを利用して，異種ネットワークの統合を行っている．PUCC プロトコルは，P2P ネットワークを利用して効率的なノード探索，サービス探索が可能な拡張性の高いプロトコルである．

以下、本論文の構成を示す．第 2 章では，関連研究を紹介し，関連研究が抱える問題を指摘する．第 3 章では，システムの提案を行う．そして，第 4 章で実装の詳細について説明し，第 5 章でまとめを行う．

2. PUCC

現在の情報家電や AV ネットワークとモバイルネットワークは Bluetooth や WiFi などの共通の物理層プロトコルを利用していても，上位層のプロトコル違いから，お互いに通信を行うことはできない．このような問題を解決するために，P2P Universal Computing Consortium (PUCC)[1)] では，DLNA や ECHONET などの既存ネットワークの上に，オーバーレイネットワーク (図 1 参照) を形成することで，様々なネットワークに存在する機器を相互接続・運用可能にしている．PUCC では，オーバーレイネットワークにより異種ネットワークをつなぐため，既存のネットワークを統一する必要はなく，ベンダーの囲い込み競争を引き起こさない．

また，PUCC プロトコルスタックは図 2 のようになっており，IP や Bluetooth，Zigbee，IrDA など下位の通信プロトコルの上に，経路制御やメッセージ制御を行う PUCC コアプロトコルを実装している．これらは PUCC ミドルウェアとして提供されている．このミドルウェアを搭載したノードは PUCC ノードと呼ばれ，PUCC プロトコルを用いてオーバーレイネットワークに参加できる．また，PUCC プロトコルはデバイス発見やサービス実行

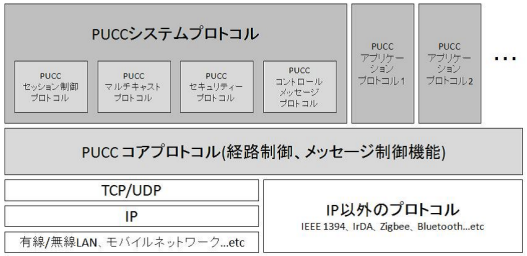

図 2　PUCC プロトコルスタック

などの各種 API も提供しており，携帯電話から家電を操作するサービスや，家にあるビデオを携帯電話で鑑賞するサービスなどが可能になる．

PUCC プロトコルを用いることでユーザデバイスは IP ネットワークなどの下位リンクを左右されず，オーバレイネットワークに参加する限りどこに移動してもサービスが利用可能である．

2.1 メタデータ

PUCC ノードに関する各種情報を記載した XML ファイルである (図 3 参照)．specification 要素にはデバイスの名前や位置，製造番号などのデバイス自体に関する諸情報が記載される．次に，電源や動作モードなどデバイスの状態情報を記した StateVariable 要素がある．そして，イベント検知などのデバイスが提供可能なサービスの内容がリスト上に記載される ServiceList が存在する．また，デバイスがサブデバイスを保持していた場合，デバイスのメタデータはサブデバイスのメタデータを含む．GW が管理する SNW のデバイスはサブデバイスにあたるため，GW のメタデータに SNW の情報が記載される．DVD デッキなどの家電のメタデータには，デバイスの名前や製造メーカー，型番，提供可能なサービスとして，録画，再生，早送りなどの機能が記載される．PUCC ミドルウェアではメタデータの送信・受信及び解析をすることによって，デバイスやサービスの探索を行い，サービスの実行のための API が用意されている．

2.2 PUCC の提供する機能

PUCC ミドルウェアを搭載し，提供される API を利用することで以下の機能が実現できる．

デバイス探索機能　DeviceDiscovery メソッドというものが用意されており，探索条件を

Vol.2010-SLDM-144 No.22
Vol.2010-EMB-16 No.22
Vol.2010-MBL-53 No.22
Vol.2010-UBI-25 No.22
2010/3/26

```
<Device type="URI of device type" id="Device ID" name="Device name">
<Specification>
 <URLBase> Device's URI </URLBase>
 <Manufacuturer> manufacture name </Manufacuturer>
    :
</Specification>
<StateVariableList>
 <StateVariable name="name" datatype="Data type" sendEvents="yes/no">
 <DefaultValue> Default value</DefaultValue>
 <AllowedValueList>
  <AllowedValue> Value </AllowedValue>
 </AllowedValueList>
</StateVariableList>
<ServiceList>
 <Service name="Service name" type="URI of Service" />
  <InputParameterList>
   <Parameter name="Name of Input Parameter" type="Data type" />
  </InputParameterList>
  <OutputParameterList>
   <Parameter name="Name of Output Parameter" type="Data type" />
  </OutputParameterList>
 </Service>
    :
</ServiceList>
<EventConditonList>
 <EventCondion id="Event ID">
  <ConditionExpression> Event  Condition Expression</ConditonExpression>
  <NotificationVariableList>
   <StateVariable name="Name of the function  to be used" />
  </NotificationVariableList>
 </EventCondition>
        :
</EventConditionList>
<PrimitiveDeviceList>
 <PrimitiveDevice type="URI of device type" id="Device ID" name="Device name">
 </PrimitiveDevice>
        :
</PrimitiveDeviceList>
</Device>
```

図 3　PUCC メタデータ

指定することで，検索条件にあったデバイスを探索することができる．デバイス探索を行いたいノードは，PUCC ネットワーク全体にメッセージを送信する．メッセージを受けとった PUCC ノードは，自分が検索条件に合う端末であった場合にメタデータを返す．

イベント登録機能　Subscribe メソッドというものが用意されており，これを利用することで PUCC ノードに対してイベント登録を行える．

イベント通知機能　Notify メソッドはイベントの通知を行う．実際に設定されているイベントが起こった時に，ユーザや他の端末にイベントの通知を行う．

サービス実行機能　Invoke メソッドを利用することで，PUCC ノードが持つサービスを実行することができる．

3. 提案システム

ここでは，本システムの構成要素と要素技術であるイベント検知とユーザインタフェースについて述べる．本システムは，図 4 に示すように，複数のネットワークが混在している環境を想定している．センサネットワーク 1 とセンサネットワーク 2 が存在した場合に，それらは単一のゲートウェイで管理されるとは限らず，図 4 のように複数のゲートウェイで管理される場合もある．このような状況でのイベント検知についても検討を行った．

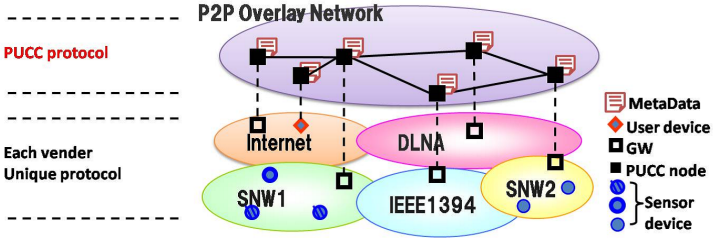

図 4　提案システムにおける想定ネットワーク

3.1 システム構成要素

ユーザデバイス　センサや家電などのデバイス情報の参照や，センサイベントと家電のアクションの関係を設定するための端末である．PUCC プロトコルを採用しているため，PUCC ノードとなる．このため，ユーザアプリケーションは PUCC プロトコルのミドルウェア API の提供する各種メソッドを利用して任意の場所からセンサデバイス発見，イベント検知サービス，家電の操作などが実行可能である．

センサゲートウェイ　超小型センサノードは処理能力が低く，PUCC プラットフォームの実相が困難であるため，センサネットワークの統合にはセンサゲートウェイを用いることで，様々なセンサの処理を行う．センサゲートウェイは，自身の担当するセンサネットワークについてデータをデータベースに蓄積するとともに，複合イベント検知などのセンササービスを提供するのが役割である．また，PUCC プロトコルを採用し，ユーザデバイスと同様 PUCC ノードとなる．このため，ユーザデバイスと同様 PUCC プロトコルのミドルウェア API の提供する各種メソッドを利用可能である．

センサデバイス　温度や湿度などの実環境を定量化するセンシング機能と他のセンサノードや GW との通信機能を備えた小型機器である．

家電ゲートウェイ　役割は家電の管理とサービスの提供である．ビデオの録画・再生や，照明の ON/OFF など，ゲートウェイが管理している家電のサービスを提供する．PUCC

Vol.2010-SLDM-144 No.22
Vol.2010-EMB-16 No.22
Vol.2010-MBL-53 No.22
Vol.2010-UBI-25 No.22
2010/3/26

プロトコルを採用した PUCC ノードであり，PUCC プロトコルのミドルウェア API の提供する各種メソッドを利用可能である．DLNA や ECHONET などのプロトコルを PUCC プロトコルに変換している．家電ゲートウェイとセンサゲートウェイは，同一端末上に実装されていても構わない．

3.2 イベント検知

イベントとは状態の変化であり，単一イベントと複合イベントの 2 種類が存在する．単一イベントは各 1 つずつのセンサ要素と閾値，関係演算子により定義される述語である．例えば室温が 100 ℃以上であればスプリンクラーを起動するという火事検知アプリケーションならば必要なのは「100 ℃を超えた」というイベントの真・偽の情報なので，そのイベントを $P(x)$ とすると $P(x) = temperature > 100$ と表現できる．

一方で複合イベントは単一イベントよりも条件が複雑化したものであり，$E = P_1(x) \wedge P_2(x)$ のように単一イベントと論理演算子で構成される．より複雑なイベントを定義する場合や，情報の精度を向上させる場合に用いられる．例えば，火事検知にしても温度のみを利用すると料理や暖房器具を誤検知するなど，情報の精度が低い可能性がある．そこで，温度と煙を同時に使うを検知制度が上がる．

また，単一のワイヤレスセンサネットワークでは大規模化が難しいというスケーラビリティの課題もある．例えば，我が家と他県に別居中の老いた両親の家の両方にホームセキュリティ用センサネットワークを完備し，イベント情報を一元管理したいというニーズがあるかもしれない．このように複合イベントの検知は特定エリアや特定ネットワーク内のみに限定されるべきではなく，規模に関わらず機能させるためにスケーラビリティが考慮される必要がある．

3.2.1 イベントツリーを用いた複合イベント検知

まず，登録フェーズについて述べる．ユーザアプリケーションはまず任意のノード (ここではセンサ GW を指す) を選択し，Create Composite Device サービスを実行する．このサービスは仮想的なデバイスを作成し，ユーザにその URI を提示する．次にあらかじめ得た任意のセンサネットワークのメタデータの情報を元に単一イベントを組み合わせて複合イベント条件式を作成し，先に選択したデバイスに対して複合イベントの条件式を含めた Subscribe メッセージを送信する．これにより先ほどの仮想デバイスはそのアプリケーションより提示された独自の複合イベントを検知できるデバイスとなると共に，イベントが発生したときにはアプリケーションに対して通知できる．

メッセージを受信したノードは筆者らの開発したセンサゲートウェイ用ライブラリの提供

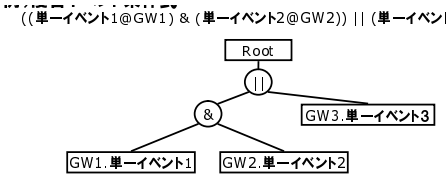

図 5 ユーザインタフェース生成

する API を利用して，構文解析とイベントツリー (図 5 参照) の作成を行う．イベントツリーは複合イベント，単一イベントと論理演算子で構成され，それぞれ根，葉，節に対応する．イベントツリーの全ての葉を調べ，単一イベントが自身の管理するセンサネットワークで検知されるものであれば，PUCC センサ GW 用ライブラリに単一イベントを登録する．もし他のセンサゲートウェイで管理するセンサネットワークで検知されるべきものであれば，そのセンサ GW にサブスクライブメッセージを送信する．(ちなみに Subscribe メッセージを送信したノードが受信したノードの親となる)

次にイベント収集フェーズについて述べる．まず，親ノードは自身の管理するセンサネットワークにおいてデータが通知された場合はまず，各単一イベントの条件を満たすかどうかを判断する．もし満たしていれば，次に複合イベントを満たすかを判定する．子ノードからの通知を受けた場合は，その都度，複合イベントを満たすかを判定する．なお，これらの複合イベント判定は作成されたイベントツリーに沿って行われる．そして複合イベントを満たすと判断された場合は，購読先のアプリケーションに通知する．

3.3 ユーザインタフェース

ここまで，異種ネットワークの統合方法について述べてきた．しかし，統合されたネットワーク上に存在するセンサや家電などのデバイスの情報を参照する，又は連携の設定のためにユーザインタフェースが必要になる．異種ネットワークをシームレスにつなげる技術があっても，ユーザインタフェースがデバイス追加の度に再度プログラミングであっては意味がない．そこで，我々のシステムではメタデータを利用し動的に GUI を生成することで，ネットワーク上のデバイス情報を更新している．

3.3.1 動的な GUI 生成

ユーザインタフェース生成までのフローを図 6 に示す．ユーザデバイスは PUCC ミドルウェアを搭載した，PUCC ノードであることを想定している．まず，ユーザデバイスは PUCC ミドルウェアが提供する機能を用いてネットワーク上のデバイスにメッセージを送

Vol.2010-SLDM-144 No.22
Vol.2010-EMB-16 No.22
Vol.2010-MBL-53 No.22
Vol.2010-UBI-25 No.22
2010/3/26

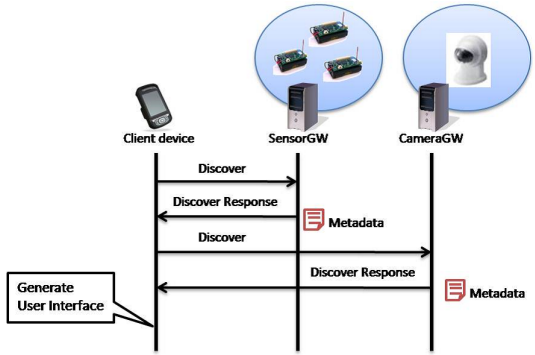

図 6　動的 GUI 生成

信する．メッセージを受け取ったデバイスは，自分の持つメタデータをユーザデバイスに返信する．その後，ユーザデバイスは受け取ったメタデータを解析し，デバイスに合わせてGUI を生成する．非常に単純だが，このような手順でユーザインタフェースを生成することで，ネットワーク上に新たにデバイスが追加されても，ユーザインタフェースの変更が必要になることはない．

4. 実　　装

実装システムの概要を図 7 に示す．ここで実際に利用したデバイスの紹介と，センサゲートウェイ及びユーザデバイスのモジュール構成の説明を行う．

4.1 利用デバイス

センサゲートウェイ　Windows XP 搭載のノート PC を利用した．

センサ　MOTE/Mica2 と NEC/Tokin を利用した．

家電ゲートウェイ　同じく，Windows XP 搭載のノート PC を利用した．

家電　Sony の SNC-P5 ネットワークカメラを利用した．

ユーザデバイス　Android 搭載の HT-03A を利用した．

4.2 センサゲートウェイ

本論文では，PUCC ミドルウェアを搭載し，センサを管理するデバイスをセンサゲートウェイと呼ぶ．センサゲートウェイは様々なセンサネットワークから収集したデータを統合

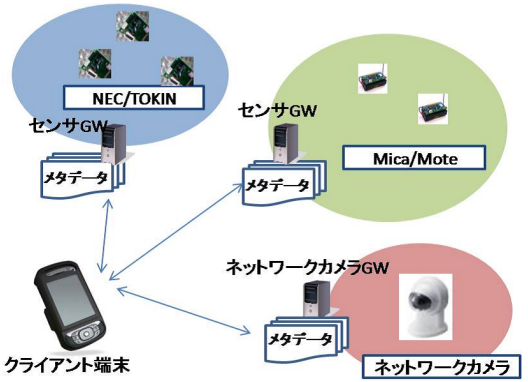

図 7　実装システム

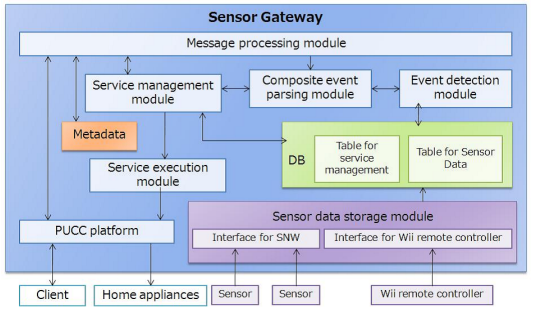

図 8　センサゲートウェイのモジュール図

し，クライアント端末や他のデバイスにセンサデータを送信する．センサゲートウェイのアーキテクチャを図 8 に示す

メッセージ処理モジュール　PUCC プラットフォームを通してクライアント端末などの他の PUCC ノードと通信の際のメッセージ処理をここでおこなう．送られてくるメッセージに応じて必要な処理をおこない，DB のテーブルを参照したりサービス管理部にイベント・サービスの定義を要求したりする．また，サービス実行時にサービス実行部から送られてくるメッセージの送信などにおいても，ここで処理をおこなう．

センサデータ蓄積モジュール　複数の異なる種類のセンサデバイスを統一的に扱うために，リアルタイム性をそれほど要求されないセンシングデータについてはすべて 1 つの統合デー

Vol.2010-SLDM-144 No.22
Vol.2010-EMB-16 No.22
Vol.2010-MBL-53 No.22
Vol.2010-UBI-25 No.22
2010/3/26

タベース（DB）に値を登録する．現在，センサデバイスの値取得部における統一的なインタフェースは存在しないため，センサ GW で対応していないインタフェースを用いたセンサデバイスを新たに利用する場合は，そのセンサデバイスから送られてくる値を統合 DB に格納するための専用のインタフェースの開発をおこなう．同時に，そのセンサデバイスの情報をメタデータに書き込み，統合 DB にセンサデバイス用のテーブルを作成する．センサデータ格納部ではセンサから送られてきたデータを処理し，統合 DB に格納していく．

イベント検知モジュール クライアント端末によって指定されたセンサデバイスを監視し，イベント発生条件に合致するかどうか判定をおこなう．センサの値がイベント発生条件を満たした場合はサービス管理部にその内容を伝える．また，監視していたセンサデバイスがなんらかの理由によりネットワークから離脱した場合はイベント検知が不能となったことをサービス管理部に伝える．

複合イベント判定モジュール 複合イベント検知部において異なるセンサ GW に属する複数のセンサデバイスを用いた複合イベントの検知を可能とする．複合イベント検知部ではクライアント端末から送られてきた複合イベント条件式の構文解析とイベントツリーの生成をおこない，指定されたセンサデバイスの監視をイベント検知部にうながす．もしセンサの値がイベント発生条件に達した場合は，イベントツリーに基づいてサービス実行条件を満たしているかを判定し，満たしていればサービス管理部にその内容を伝える．満たしていなければ，まだイベント発生条件を満たしていないセンサ GW に自身はイベント発生条件を満たしていることを通知する．また逆に，センサの値がイベント発生条件から外れてしまった場合には同様にして監視している複合イベントに関連するセンサ GW にそのことを伝える．

アクション管理モジュール クライアント端末によって指定されたイベント発生時に実行するサービスの管理をおこなう．もしイベント検知部よりイベント検知不能のメッセージを受けた場合は，サービス実行部を通してクライアント端末にそのことを知らせる．

4.3 家電ゲートウェイ

家電機器と携帯端末・センサ GW との仲介をなす家電制御 GW については，PUCC の家電制御 WG にて研究・開発されているものをそのまま利用することととする．この GW によって，インターネットに接続可能なネットワークカメラや DLNA をサポートした AV 機器，ECHONET に準拠した白物家電を操作することが可能となる．本論文では，家電ゲートウェイの実装の詳細については説明を省く．

4.4 ユーザデバイス

本論文では，PUCC ミドルウェアを搭載し，PUCC ノードの情報の参照及び，ノード間の連携の設定を行う端末をユーザデバイスと呼ぶ．PUCC メタデータを利用した動的な GUI 生成と，生成されたインタフェースを利用してデバイスの連携の設定を行うことができる．簡単に分割すると，PUCC のメッセージのやり取りを行うメッセージ処理モジュールと，GUI 生成を行うモジュールとに分かれる．

メッセージ処理モジュール PUCC プラットフォームを通してクライアント端末などの他の PUCC ノードと通信を際のメッセージ処理をここでおこなう．

GUI 生成モジュール 取得したメタデータを解析し，それに従って GUI を生成する．ユーザは生成されたインタフェースを使い，イベントやアクションの設定を行う．

5. ま と め

本論文では，PUCC の技術を利用しユーザが携帯端末を用いて自由にセンサと家電製品を絡めた連携動作を設定することのできる環境の構築について述べた．本論文においては家電製品としてネットワークカメラを用いたが，PUCC のほかの WG にて開発中の GW とも通信をおこなうことによって，本システムの大きな変更なくほかの機器でもセンサと連携環境を組むことが可能である．今後は，センサの種類を増やすことでより汎用的なシステムを目指すと共に，位置情報を考慮したデバイスの探索やサービス定義の拡張などを進めていきたい．

参 考 文 献

1) N. Ishikawa, T. Kato, H. Sumino, S. Murakami, J. Hjelm, *PUCC Architecture, Protocols and Applications*, IEEE Consumer Communications and Networking Conference (CCNC2007), pp.788-792, 2007.
2) H. Sumino, Y. Uchida, N. Ishikawa, H. Tsutsui, H. Ochi, Y. Nakamura, *Home Appliance Control from Mobile Phones*, IEEE Consumer Communications and Networking Conference (CCNC2007), pp.793-797, 2007.
3) A V U Phani Kumar, Adi Mallikarjuna Reddy V, D. Janakiram, *Distributed Collaboration for Event Detection in Wireless Sensor Network*, The 3rd international workshop on Middleware for pervasive and ad-hoc computing(MPAC 2005),December 2005.
4) M. Yu, J. Song, J. Kim, K. Shin, P. Mah, *NanoMon: A Flexible Sensor Network Monitoring Software*, The 9th International Conference on Advanced Communication Technology (ICAST2007), Vol.2, pp.1423-1426, 2007.