

# モバイル環境下におけるサービス利用のための タスク知識記述および実行環境

菊地 悠<sup>\*1</sup> 津田 雅之<sup>\*2</sup> 富岡 淳樹<sup>\*1</sup> 長沼 武史<sup>\*1</sup> 倉掛 正治<sup>\*1</sup>

<sup>\*1</sup> 株式会社 NTT ドコモ マルチメディア研究所

<sup>\*2</sup> 株式会社 NTT ドコモ 移動機開発部

あらまし

様々なネットワークに多様なサービスや情報が提供され、このような情報を携帯電話などの移動端末を介して利用することが可能になってきたが、数多くの情報・サービスを利用するためにはユーザ自身のノウハウによるところが大きい。本稿ではユーザがモバイルサービスを利用する際に必要となる知識をタスクオントロジの枠組みで表現し、簡便にサービスを提供するための記述要件および利用方法の検討内容について述べる。

## 1. 背景

携帯電話に代表されるモバイルインターネット環境の普及により、時間や場所を問わずにネットワーク上の様々なサービスを利用できるようになってきた。Web によるブラウジングに限らず、プログラムをインターネット上からダウンロードしてアプリケーションを利用したり、アプリケーションを介して実世界上での商品購入、電子チケットとしての利用など、ユビキタス社会の萌芽とも捉えられるサービス形態も出現するようになった。

将来的には、携帯端末上での Web サービスの利用、無線 LAN や Bluetooth、RFID などの近距離無線、GPS などによる位置情報取得、あるいは Felica による電子決済などを利用・組み合わせた本格的なユビキタスサービスの利用が可能になることが予想される（Fig.1 参照）。

本来このようなサービスは、我々が日常生活で直面する様々な問題解決を支援するものであり、これらを利用することにより効率的に目的（タスク）を実現することが可能になると考えられる。現在でも、屋外で終電の時刻を調べるとか、外出先で食事をするためのレストラン検索などといったモバイルサービスの利用方法はかなり一般的に見受けられる。

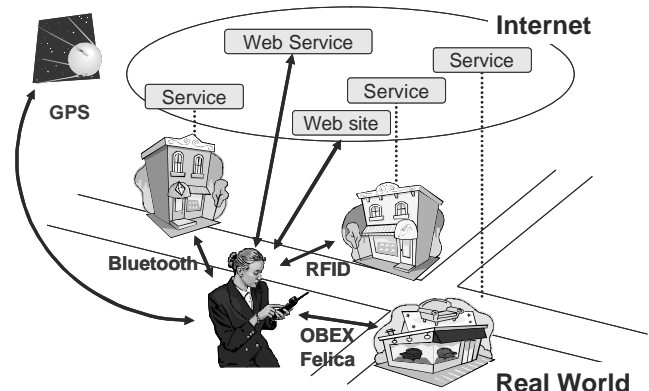


Fig.1 携帯端末を通して将来提供されるサービス

一方でこうしたサービスを検索するための手段として提供されるのは、依然としてディレクトリ型のメニューもしくはキーワードによる検索といったものが主流であり、目的のサービスを見つけ出すのに経験やスキルを要する。また今後のユビキタスサービスの実現に向けて、携帯端末がサポートする機能が多様化することは、先の例で挙げたモバイルインターネット環境におけるサービスの検索のみならず、サポートされる様々なデバイスの利用方法をユーザが理解することを強いることとなり、利便性の享受のためにユーザは今以上の労力を要することになる。

このような背景を受けて、我々はモバイル環境におけるサービスの利用方法を“タスク知識”と

して記述・表現し、サービス利用者であるエンドユーザが簡単にサービスを利用できる枠組みとして、以下に示す 3 つの目的を実現することを目指している。

- 1) タスク知識記述言語により表現されたオントロジをユーザが参照し、ユーザがサービスを簡単に検索する。
- 2) 上記オントロジを用い、タスク実現にあたって関連付けられる複数のサービス・様々な形態のサービスを組み合わせて利用する。
- 3) ユーザ独自のサービス利用方法をタスク知識として構造化する環境を提供し、ユーザ自身が生み出したタスク知識を交換・流通させる。

1)に関しては、先のモバイルインターネット環境において、実際にユーザがサービスを利用するシーンを想定した事例ベースのオントロジ構築[1]を行っている。ここで構築されるオントロジは、オントロジ開発者により管理・メンテナンスされることを前提としている。

2)は、モバイル環境におけるサービス利用という観点から、タスク知識記述として構成されたオントロジをどのように利用するか、またそのためにはどのような要件が記述・実行環境に求められるかということに焦点を当てている。

3)の背景としてあるのは、オントロジを利用するユーザ自身がタスク知識を生み出すという考え方であり、Semantic Web 的な世界観に近い。インターネットの世界に流通する情報・サービスの多様性にタスク知識が応えるためには、オントロジとして一元的に管理・運営される知識に加え、ユーザ自身の多様性による自由な知識の構築と流通が不可欠であると我々は考えている。

本稿では 2)に焦点を当て、オントロジの構築そのものではなく、“モバイルサービスをユーザに使いやすく提供する”ために、オントロジの枠組みの中でサービスの利用をどのように記述するかという問題、および記述された情報からどのようにユーザへサービスを提供するかという実行環境

の 2 点にフォーカスする。

## 2. 関連技術・研究

Web サービスの連携に焦点を当てたものとして BPEL[2]、DAML-S[3]などが存在する。BPEL は既存の Web サービスコンポーネントを連携することに焦点を当てており、BPEL によるプロセス記述が第 3 者的な表現をとる点が特徴として挙げられ、これにより記述のポータビリティが確保されている (BPEL の解釈エンジンは、サーバでも中間サーバでも、あるいはクライアント上で動作しても構わない) ほか、記述そのものがインタープリタ方処理をそのまま反映している、といった特徴を備えている。また DAML-S では、Web サービスコンポーネントの連携におけるプロセス表現に対し、DAML をベースにした制御構造・状態表現の枠組みを提示している。

BPEL、DAML-S はいずれも Web サービスに焦点を当てた技術である。すなわちこれらはサーバサイドの技術であると同時に、サービス指向の記述言語である。

本研究ではサービスを利用するエンドユーザを中心に据え、エンドユーザが実現したいタスクをタスク知識として表現するタスク指向のアプローチを採用する。またサービス提供形態を Web サービスに限定しない点、クライアントである携帯端末上での実行を前提としている点などが大きく異なる。

## 3. アプローチ

タスク知識記述表現に基づくモバイルサービスの利用にあたって、タスク知識による記述の要件を明らかにする必要がある。3.1 節において、タスク知識記述における前提条件を述べる。また、3.2 節においてタスク知識記述を用いたモバイルサービス利用の具体例を示し、サービスを利用するという観点から要件を明らかにする。

### 3.1 前提条件

本研究では、ネットワーク上で提供される様々なサービスを、ユーザが実世界で直面する問題

(タスク)を解決するための具体的手段と位置付ける。このため、着メロのダウンロードやゲームなどのコンテンツは対象外とし、ネットワーク上の情報やサービスを利用することによって解決可能な問題のみを取り扱う。

対象とするサービスの種類は、HTML による Web コンテンツ、Web アプリケーション、端末上で実行するアプリケーション、Web サービスなどであり、タスク知識の記述者とサービス提供者は異なるものとする。

タスク知識記述は、ユーザが実際に解決したいタスクをタスク・サブタスクの関係で階層構造として表現する。また、タスクの階層構造を構成するノードは、記述されたタスクと実際のサービスと粒度が一致する場合にリンクを張る。“映画を見る”というタスクにおけるサブタスクおよびリンクされたサービスの例を Fig.2 に示す。

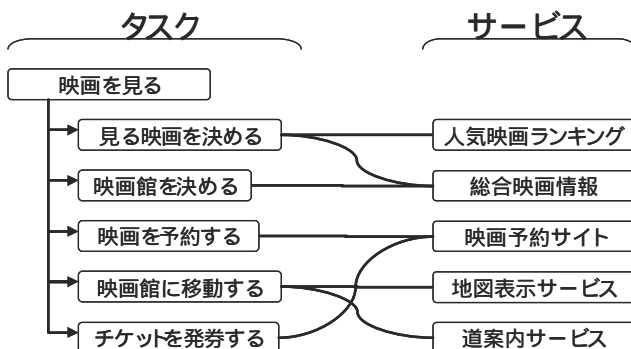


Fig.2 タスク階層構造とサービスへのリンク

タスク知識の記述とはサービスに対して“目的”という観点から付与されたメタデータであり、またサービスとはタスク実現を直接的に支援する“手段”として捉えることができる。なお、ここまで述べてきた前提条件に関する議論の詳細は[1]を参照されたい。

モバイルという視点でサービスを捉えたとき、今後ますます実世界とのインタラクションが重要になってくる。携帯電話で自販機から飲み物が買えるように、提供されるサービスは多くが実世界と関連付けられ、必要に応じて携帯電話等の端末が有する局所通信機能などを用いてやり取りを行う。実際に利用する端末機能の具体例としては、

GPS・OBEX（赤外線通信）・RFID（無線 ID タグ）・Felica（電子マネー）・Bluetooth などが挙げられる。我々はこうした、端末に依存する機能を利用するサービスについても取り扱う。サポートされる機能は端末ごとに異なるものとする。

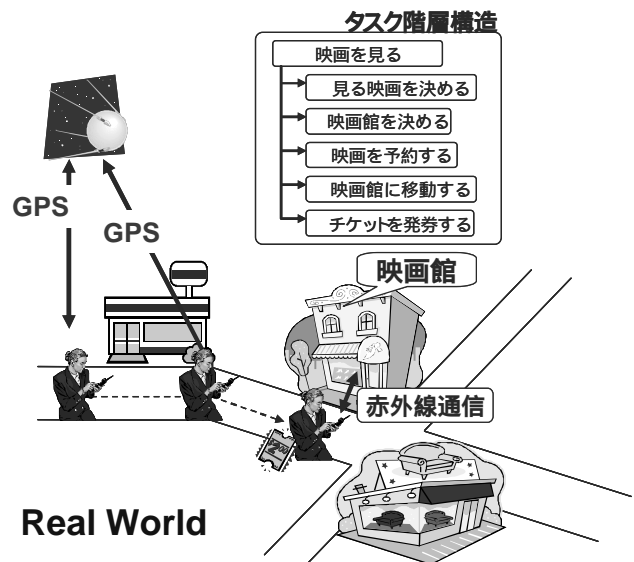


Fig.3 複数タスクの連携によるユーザ支援

### 3.2 具体的なサービス事例

Fig.3 はタスク知識記述言語を用いて実際にユーザが様々なサービスを組み合わせて利用するシーンを表している。ここでのタスクは Fig.2 に示した“映画を見る”という事例に基づくものであり、本研究により実現を目指す具体的なユースケースである。

Fig.3 内に示した ~ の番号は、実世界上でのユーザの状態と、ユーザによるタスクの実行手順の関連を表すためのものである。なおここでのタスク実行手順はタスク連携を説明する際の便宜的なものであり、4 節で述べるように実際の実行手順にはある程度の自由度が存在する。

以下、Fig.3 について説明する。

#### 見る映画を決める

ユーザは“映画を見る”というタスクを解決するため、まず“見る映画を決める”タスクを選択する。タスクからは総合映画情報・人気映画ランキングといったネットワーク上のホーム

ページにリンクが張られており、ユーザはこれらの情報を参照して、見る映画を決定する。

映画館を決める

映画を決定した後、ユーザはその映画を見る“映画館を決める”タスクを選択する。

映画を予約する

見る映画・映画館を決定した後、ユーザはその映画を予約するため、映画予約サイトを参照し、予約する。

映画館へ移動する

予約の後、ユーザは映画館の場所をよく知らなかったため、“映画館へ移動する”タスクを選択する。所持する端末がGPSの機能を有すると仮定し、ユーザは映画館までの道案内をするサービスを利用し、実際に映画館へと移動する。

チケットを発券する

映画館へ移動すると、あらかじめ予約時に取得した予約番号などの情報を利用して、映画館に設置された発券機から、チケットを発券する。この際、端末の赤外線通信機能あるいは端末画面上にバーコードを表示するなどの認証作業を行う。

このように、“映画を見る”タスクを実現するために、関連する様々なタスクにリンクされたサービスを利用することが必要となる。

#### 4. タスク連携の要件

3.2 節でサービス事例を簡単に説明したが、様々な形で提供されるサービスをモバイル環境で連携させるためには、以下のような制約を考慮する必要がある。

##### 4.1 実行手順の制約

3.2 節では便宜的に実行手順を から の手順に固定して説明した。実際には“映画を予約する”というサブタスクを実行する前には、事前に“見る映画を決める”と、“映画館を決める”という二つのタスクを実行しておかなければならない、というような実行手順の制約というものが存在する。

タスクまたはサービスの実行手順は、先の制約

を満たす範囲であればユーザ自身によって実行手順を選択することが可能である。

またタスクの解答をあらかじめユーザが持っている場合には、タスクの実行を割愛することも可能である。例えば“見る映画”が事前に決まっている場合などがこれにあたる。

##### 4.2 端末デバイスによる制約

端末上で実行が可能なサービスというのは、端末がサポートする機能によって実際には異なる。

3.2 節の例では、“映画館へ移動する”タスクの支援として端末のGPS機能を利用した道案内サービスを利用した。このように、モバイルサービスにおいては端末がサポートする機能によって受けられるサービスの種類に制約が生じる。

端末デバイスによる制約は、実際にはタスクではなく、リンクされたサービスに依存する。GPS機能を端末が有していない場合であっても、“映画館へ移動する”タスクに対して、地図を表示するといった代替サービスがリンクされていれば、端末の多様性に非依存な形で、様々な形態のサービスに対応することができる。

##### 4.3 実世界のユーザ状態に基づく制約

3.2 節の例で、予約した映画のチケットを映画館内に設置されたチケット発券機から取得するというサービス事例を示した。この場合、ユーザが発券機の前に移動したことを受けてタスクが遷移する必要がある。すなわちタスクの連携には実世界におけるユーザの状態に基づいた遷移の制約が存在する。

遷移のタイミングは、実世界上のあるデバイスと、ユーザが所持する携帯端末とのインタラクションを通して検知することができる。これには、赤外線通信やBluetooth、RFIDといった局所通信機能や、ディスプレイ上に表示したバーコードを発券機が搭載するカメラを用いて認識するといった方法が考えられるが、いずれの場合においても端末の持つデバイスに依存するため、ユーザ状態に関する制約は常に端末デバイスの制約を包含することになる。

## 5. タスク知識の記述

前節でのモバイルサービスに関する要件を基に、タスク知識の記述方法を述べる。

Fig.4 にタスク知識表現のレイヤ構成を、実世界のユーザ要求と対応付けて示す。タスク知識表現は大きく3つの要素から構成される。

- ・ タスクに関する記述 (Task Representation)
  - タスクそのものの意味的な記述 (タスクの対象・タスク自身の目的・処理による意味的な入出力関係)
  - 他のタスクとの関係の記述 (処理の実行手順に関する明示的な制約)
- ・ タスクとサービスのマッピング (Task-Service Grounding)
  - 実際に提供されるサービスに対して、タスクとのリンク
  - サービスの形態 (e.g. HTML、Web アプリケーション、端末上で実行するアプリケーション、Web サービス .etc)
- ・ サービスに固有の情報の記述 (Service Representation)
  - 端末デバイスの利用に関する情報
  - プロトコル・バインドなど

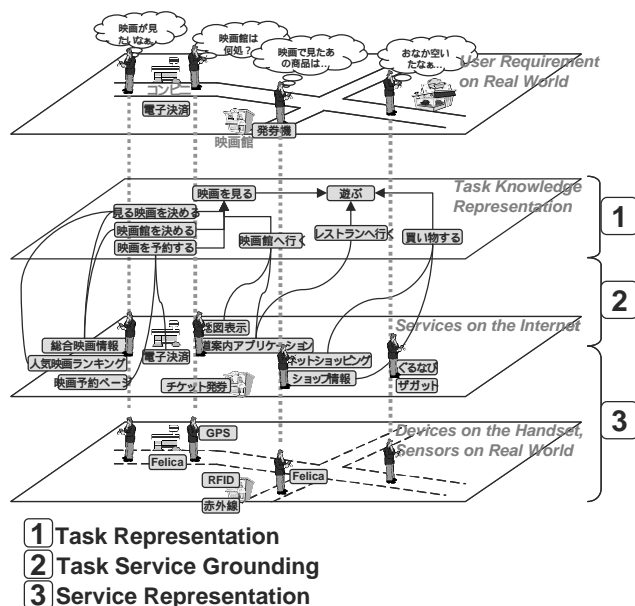


Fig.4 タスク知識表現のレイヤ構成

なおタスク知識の記述には携帯端末上での処理の負荷も考え、Semantic Web 技術として提案されている OWL のようなオントロジ記述言語ではなく、通常の XML をベースとした。

## 5.1 タスクに関する記述

タスクに関する記述の具体例を Fig.5 に示す。これは Fig.3 に示した複数タスクの連携例に従ったものとなっている。タスクに関する記述は、“タスクそのものの意味的記述”と、“タスク間の関係の記述”に分かれる。

```
<task ID="映画を見る">
  <subTaskOf name="デートする"/>
  <taskObject name="映画"/>
  <taskPredicate name="見る"/>
  <inputObject name="チケット"/>
</task>

<task ID="見る映画を決める">
  <preTaskOf name="映画を見る"/>
  <taskObject name="映画"/>
  <taskPredicate name="決める"/>
  <outputObject name="映画タイトル"/>
</task>

<task ID="映画館を決める">
  <preTaskOf name="映画を見る"/>
  <taskObject name="映画館"/>
  <taskPredicate name="決める"/>
  <outputObject name="映画館"/>
</task>

<task ID="映画を予約する">
  <preTaskOf name="映画を見る">
  <taskObject name="映画"/>
  <taskPredicate name="予約する"/>
  <inputObject name="映画タイトル"/>
  <inputObject name="映画館"/>
  <outputObject name="予約番号"/>
</task>

<task ID="映画館へ移動する">
  <preTaskOf name="映画を見る"/>
  <postTaskOf name="映画館を決める"/>
  <inputObject name="映画館"/>
</task>

<task ID="チケットを発券する">
  <preTaskOf name="映画を見る"/>
  <taskObject name="チケット"/>
  <taskPredicate name="発券する"/>
  <inputObject name="予約番号"/>
  <outputObject name="チケット"/>
</task>
```

Fig.5 タスクに関する記述例

“ タスクそのものの意味的記述 ” は、  
<taskObject>および<taskPredicate>により行う。  
これは<task>を主語と捉えた場合、それぞれ目的  
語、述語に相当する概念で説明したものになる。  
この二つの要素は<task>タグの子要素として必須  
である。

“ タスク間関係の記述 ” は、親子関係によっ  
て表現されるものと、意味的入出力関係によって  
表現されるものがある。

親子関係には、具体的に以下の 3 種類を定義す  
る。DAML-S ではプロセス表現と状態表現をそれ  
ぞれ<process>クラスと<condition>クラスで区別  
しているが、本記述手法では、タスク表現自体が  
状態表現を兼ねるものとみなしている。

- <subTaskOf>
  - 上位タスクを具体化したタスク
  - e.g. “ デートする ” (subTaskOf) “ 映  
画を見る ”
- <preTaskOf>
  - 上位タスク実現に関連付けられて、事前  
に行う必要があるタスク
  - e.g. “ 映画を見る ” (preTaskOf) “ 映  
画を予約する ”
- <postTaskOf>
  - 上位タスク実現に関連付けられて、事後  
に行う必要があるタスク
  - e.g. “ ネットショッピングをする ”  
(postTaskOf) “ 商品を受け取る ”

また、意味的な入出力関係は、<inputObject>、  
<outputObject>の二つのパラメータ表現を用いる。  
意味的な入出力パラメータの対応関係を見ること  
によって、タスクを実行する手順の制約が表現さ  
れる。

Fig.5 に示した記述例の入出力関係を基にすると、  
Fig.6 に示す処理手順の制約を導くことができる。  
複数のタスクが並行して並んでいる部分が存在す  
るが、ユーザはサービスを利用したタスクの実行

時にいずれのタスクからも選択が可能である。

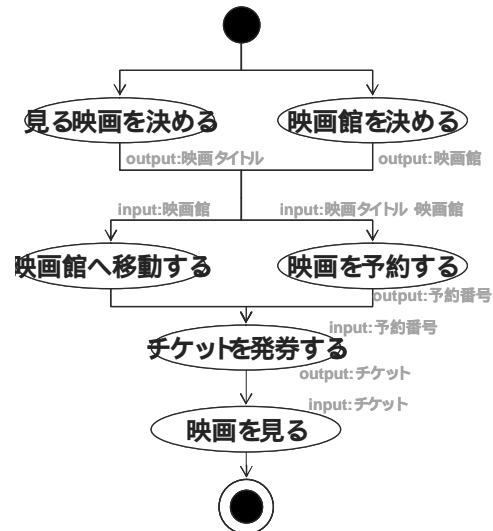


Fig.6 タスク処理手順の制約

## 5.2 タスクとサービスのマッピング ( Grounding )

タスクとサービスのマッピングでは、サービス  
の種類によってマッピングの仕方が異なる。ここ  
では通常の HTML によるコンテンツおよび XML  
により記述されたコンテンツの例を Fig.7 に示す。

```
<grounding name="見る映画を決める">
  <resource type="html">
    <uriReference uri="http://some_url1.html"/>
  </resource>
  <resource type="xml">
    <uriReference uri="http://some_url2.xml">
      <parameterMap>
        <inputObject name="映画タイトル"/>
        <parameterObject name="movieTitle"/>
      </parameterMap>
    </resource>
  </grounding>
```

Fig.7 タスクとサービスのリンク

対象となるコンテンツが html の記述の場合、  
意味的な入出力に対応付けて特定のデータを指定  
することはできないが、XML 形式のデータが提供  
される場合には、意味的な入出力パラメータに対  
し直接タグの名前を指定してやる  
( <parameterObject>タグで記載 ) ことで、パラ  
メータに対してもマッピングを行うことが可能で  
ある。また、マッピングされたパラメータのデー  
タ型が一致する場合には、タスク知識の処理・実

行環境が対応することでデータの受け渡しをすることもできる。

### 5.3 サービスに固有の情報の記述

サービスに固有の情報としては、サービスの利用に必要な端末側の機能や、データのやり取りを行うプロトコルなどに関する記述が必要である。携帯電話の赤外線機能を利用したデータのやり取りを想定した記述を Fig.8 に示す。

```
<service ID="http://some_uri">
  <useDevice type="赤外線">
    <communication>
      <parameterPost name="reservationNo"/>
      <parameterGet name="result"/>
    </communication>
  </useDevice>
</service>
```

Fig.8 サービスが利用するデバイス情報の記述

Fig.8 の例では、サービスを利用する端末を携帯電話に限定しているため、記述の内容は非常にシンプルであるが、Web サービスなどを対象とした場合には、DAML-S における Grounding 記述のように、複雑な表記が必要となる。

## 6. 実行環境

Fig.9 にシステム全体の構成を示す。

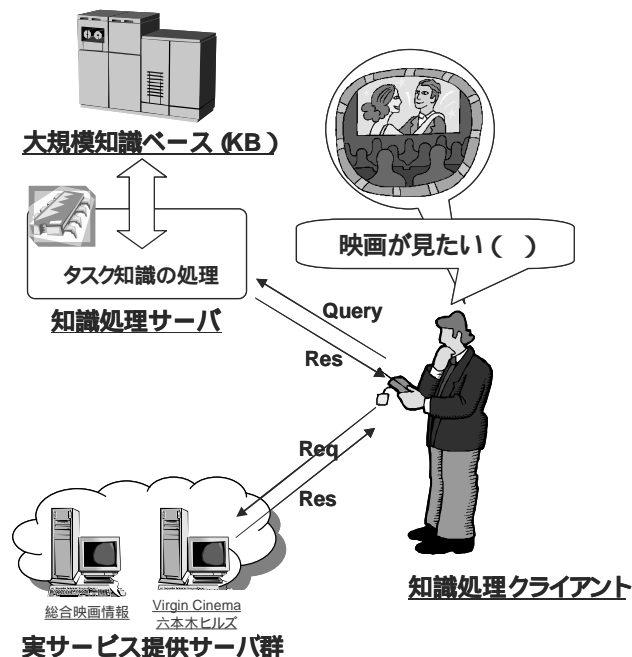


Fig.9 システム全体の構成

NW 上に数多く存在するサービスのメタデータとして、タスク知識は記述される。またタスク知識の記述は大規模知識ベース (KB) として意 h 資源的に管理される。ユーザは携帯端末 (Fig.9 では知識処理クライアントと表記) を利用してこれらのサービスを選択、実行するものとする。

知識処理サーバは主にタスクやサービスの検索に関する処理を行い、知識処理クライアントは、UI および実際のサービスの利用を行うための最低限の処理を行うものとする。

Fig.10 にクライアント端末が備える機能を示した。実行環境は主に以下の 4 つのモジュールから構成される。

Task Knowledge Description Parser : タスク知識の記述を解析するパーサ

Device Manager : クライアント端末の有するデバイスに関する情報の管理

Transition Manager : 複数のタスクを連携して実行するためのトランザクション管理

Application Manager : クライアント上で実行するサービスの管理。付随して Web の閲覧に際してのブラウザや、アプリケーション、モバイルコードのダウンロードなどの処理に加え、アプリケーション間のデータ受け渡しのためのパラメータコンテナを備える。

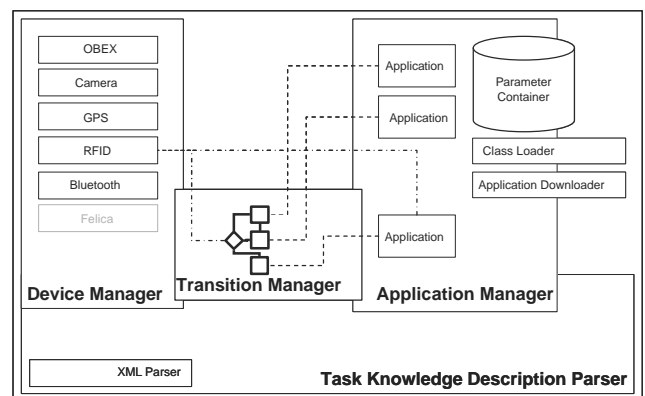


Fig.10 知識処理クライアント実行環境

## 7. 結論

本稿ではモバイル環境においてユーザが様々な情

報・サービス・クライアント端末の機能を利用し、複合されたサービスをシームレスに利用可能にするフロー記述言語の設計および実行環境の提案を行った。

今後は言語の実行環境の整備によるフィージビリティ評価を行うとともに、ユーザが独自でタスク知識を構造化できるツールの構築を進めていく予定である。

## References

- [1]：長沼、磯田、倉掛、“人間行動に基づく実世界セマンティクスの構築”：情報処理学会研究報告「情報学基礎」、No.72-008
- [2]：The DAML Service Coalition, DAML-S: Semantic markup for Web Services,  
<http://www.daml.org/services/>
- [3]Business Process Execution Language for Web Services Version 1.1 :  
<http://www106.ibm.com/developerworks/webservices/library/ws-bpel/>