# FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

# FIPA Device Ontology Specification

| | | | |
|---|---|---|---|
| **Document title** | FIPA Device Ontology Specification | | |
| **Document number** | XC00091C | **Document source** | FIPA Gateways TC |
| **Document status** | Experimental | **Date of this status** | 2002/05/10 |
| **Supersedes** | None | | |
| **Contact** | fab@fipa.org | | |
| **Change history** | | | |
| 2002/05/10 | Approved for Experimental | | |



*Geneva, Switzerland*

## Foreword

The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent-based applications. This occurs through open collaboration among its member organizations, which are companies and universities that are active in the field of agents. FIPA makes the results of its activities available to all interested parties and intends to contribute its results to the appropriate formal standards bodies.

The members of FIPA are individually and collectively committed to open competition in the development of agent-based applications, services and equipment. Membership in FIPA is open to any corporation and individual firm, partnership, governmental body or international organization without restriction. In particular, members are not bound to implement or use specific agent-based standards, recommendations and FIPA specifications by virtue of their participation in FIPA.

The FIPA specifications are developed through direct involvement of the FIPA membership. The status of a specification can be either Preliminary, Experimental, Standard, Deprecated or Obsolete. More detail about the process of specification may be found in the FIPA Procedures for Technical Work. A complete overview of the FIPA specifications and their current status may be found in the FIPA List of Specifications. A list of terms and abbreviations used in the FIPA specifications may be found in the FIPA Glossary.

FIPA is a non-profit association registered in Geneva, Switzerland. As of January 2000, the 56 members of FIPA represented 17 countries worldwide. Further information about FIPA as an organization, membership information, FIPA specifications and upcoming meetings may be found at http://www.fipa.org/.

# Contents

# 1 Scope

This document is part of the FIPA specifications and deals with device ontology. This document contains specifications for properties of devices. Additionally, the document provides an example to illustrate the usage of the ontology via a profile of a hypothetical smartphone, an example of using the ontology through CC/PP, and other informative examples.

# 2 Overview

The capabilities of different devices are best expressed using some ontology, against which the profiles of those devices are validated. This document contains specifications for a device ontology.

Provided that two devices D1 and D2 have a connection, they may exchange device profiles (either directly or through a brokering agency) and acquire a list of services provided by the other device. The list of services may include both hardware and software services, for example: a software component that provides access to a hardware component of the device (such as microphone, headset or GPS service). The profile needs to support the identification of services for various input and output capabilities, such as audio input and output. An informative example of a profile for a hypothetical device is given in Annex A.

The `Fipa-Device` ontology can be used by agents when communicating about devices. Agents pass profiles of devices to each other and validate them against the `Fipa-Device` ontology. The profiles come in handy for example in a situation where memory- or processing-intensive actions take place; agent A1 can ask agent A2 whether device D has enough capabilities to handle some task A1 has in mind. Annex B gives a set of informative examples showing how profiles based on `Fipa-Device` ontology can be exploited.

Related work is done both in W3C [CC/PP] and WAP Forum [UAProf]. There is an overlap between the definitions found in those documents and this specification. However, direct references to those specifications are not used here. That is because, unlike the ontology presented in this specification, they rely on specific frameworks and languages, namely RDF and XML. Annex C gives an informative example on how to use the `Fipa-Device` ontology via CC/PP descriptions.

# 3 Device Ontology

## 3.1 Object Descriptions

This section describes a set of frames that represent the classes of objects in the domain of discourse within the framework of the `Fipa-Device` ontology.

The following terms are used to describe the objects of the domain:

**Frame**. This is the mandatory name of this entity that must be used to represent each instance of this class.

**Ontology**. This is the name of the ontology, whose domain of discourse includes the parameters described in the table.

**Parameter**. This is the mandatory name of a parameter of this frame.

**Description**. This is a natural language description of the semantics of each parameter.

**Presence**. This indicates whether each parameter is mandatory or optional.

**Type**. This is the type of the values of the parameter: Integer, Word, String, URL, Term, Set or Sequence.

**Reserved Values**. This is a list of FIPA-defined constants that can assume values for this parameter.

### 3.1.1 Relationships between Frames

*Figure 1* depicts the frames used in this ontology with associations among them.

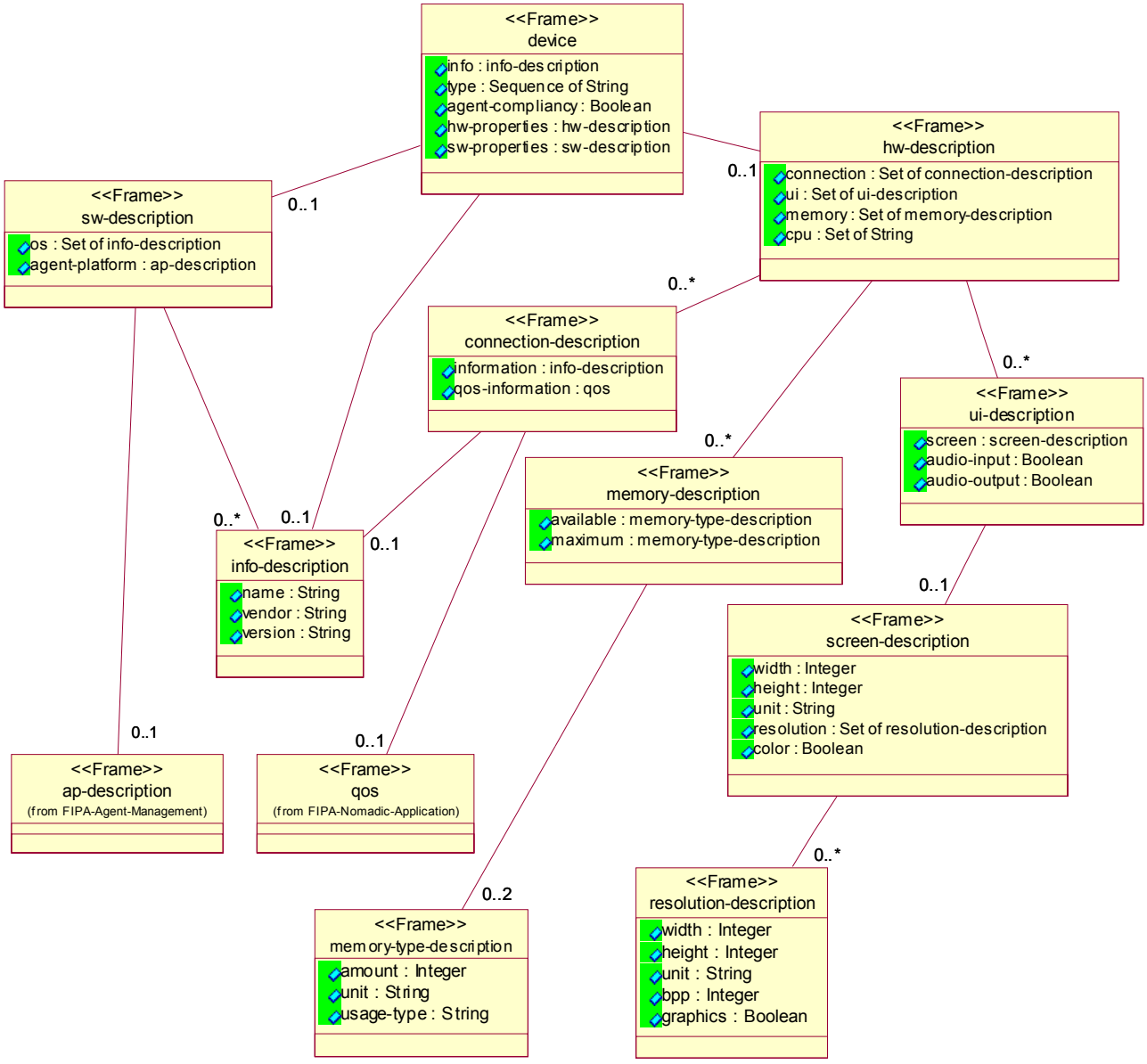



**Figure 1:** Relationships between Frames in the `Fipa-Device` ontology

### 3.1.2 Device Description

This type of object represents the description that can be used to define the device with its most general properties.

| **Frame Ontology** | `device`<br>`Fipa-Device` | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Presence** | **Type** | **Reserved Values** |
| `info` | General information for the device. | Mandatory | `info-description` | |
| `type` | The type(s) of the device. General type(s) of devices like 3G phones, PDA's etc. To be used as a sequence from general to more specific types. | Optional | Sequence of `String` | |
| `agent-compliancy` | Capability to host a FIPA-agent platform or participate in a distributed one. | Optional | `Boolean` | `true`<br>`false` |
| `hw-properties` | List of properties describing the hardware features of the device in question. | Optional | `hw-description` | |
| `sw-properties` | List of properties describing the software features of the device in question. | Optional | `sw-description` | |

### 3.1.3 Product Info Description

This type of object represents the description that can be used to define the name, vendor and version of some product.

| **Frame Ontology** | `info-description`<br>`Fipa-Device` | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Presence**[1] | **Type** | **Reserved Values** |
| `name` | The name of the product in question. | Optional | `String` | |
| `vendor` | The vendor of the product in question. | Optional | `String` | |
| `version` | The version of the product in question. | Optional | `String` | |

[1] While all of these parameters are optional, a valid `info-description` object will contain at least one parameter.

### 3.1.4 Hardware Description

This type of object represents the description that can be used to define the hardware capabilities of a given device.

| **Frame Ontology** | `hw-description` `Fipa-Device` |
|---|---|

| **Parameter** | **Description** | **Presence**[2] | **Type** | **Reserved Values** |
|---|---|---|---|---|
| `connection` | The type of the connection the device uses. | Optional | Set of `connection-description` | |
| `ui` | List of the user interfaces that the device offers. | Optional | Set of `ui-description` | |
| `memory` | The amount of memory that the device has. | Optional | Set of `memory-description` | |
| `cpu` | The type of the central processing unit that the device has. | Optional | Set of `String` | |

### 3.1.5 Connection Type Description

This type of object represents the description that can be used to define the connection-related details of a given device.

| **Frame Ontology** | `connection-description` `Fipa-Device` |
|---|---|

| **Parameter** | **Description** | **Presence**[3] | **Type** | **Reserved Values** |
|---|---|---|---|---|
| `information` | General information for the connection. | Optional | `info-description` | |
| `qos-information` | Detailed information about the Quality of Service of this connection type | Optional | `qos`[4] | |

[2] While all of these parameters are optional, a valid `hw-properties` object will contain at least one parameter.
[3] While all of these parameters are optional, a valid `connection-description` object will contain at least one parameter.
[4] The frame for `qos` is found in [FIPA00014].

### 3.1.6 User Interface Description

This type of object represents the description that can be used to define the user interface(s) of a given device.

| **Frame Ontology** | `ui-description` `Fipa-Device` | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Presence**[5] | **Type** | **Reserved Values** |
| `screen` | Information characterizing the screen of the device. | Optional | `screen-description` | |
| `audio-input` | Specifies whether the device in question is capable of receiving audio input. | Optional | `Boolean` | `true` `false` |
| `audio-output` | Specifies whether the device in question is capable of producing audio output. | Optional | `Boolean` | `true` `false` |

### 3.1.7 Screen Description

This type of object represents the description that can be used to define the screen of a given device.

| **Frame Ontology** | `screen-description` `Fipa-Device` | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Presence**[6] | **Type** | **Reserved Values** |
| `width` | The width of the screen. This value must be positive. | Optional | `Integer` | |
| `height` | The height of the screen. This value must be positive. | Optional | `Integer` | |
| `unit` | The unit for the `width` and `height` parameters of this frame. | Optional | `String` | `mm` `cm` `inch`[7] |
| `resolution` | The resolution description for the screen. | Optional | Set of `resolution-description` | |
| `color` | Has the value `true` if the device has a color screen; `false` if it has a monochrome screen. | Optional | `Boolean` | `true` `false` |

[5] While all of these parameters are optional, a valid `ui-description` object will contain at least one parameter.
[6] While all of these parameters are optional, a valid `user-interface` object will contain at least one parameter.
[7] 1mm = 0,1cm. 1mm = .03937inch. 1cm = 10mm. 1cm = . 3937inch. 1inch = 25.4mm. 1inch = 2.54cm.

### 3.1.8 Resolution Description

This type of object represents the description that can be used to define the resolution-details of a given display.

| Frame Ontology | resolution-description<br>Fipa-Device | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Presence**[8] | **Type** | **Reserved Values** |
| width | Number of resolution units horizontally. This value must be positive. | Optional | Integer | |
| height | Number of resolution units vertically. This value must be positive. | Optional | Integer | |
| unit | The unit for the resolution. | Optional | String | pixels<br>characters |
| bpp | Bits per pixel. | Optional | Integer | |
| graphics | Has the value true if the device is capable of displaying graphics; false if the device is capable of displaying only characters. | Optional | Boolean | true<br>false |

### 3.1.9 Memory Description

This type of object represents the description that can be used to define the maximum memory of a given device, as well as the memory available at the time of query.

| Frame Ontology | memory-description<br>Fipa-Device | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Presence**[9] | **Type** | **Reserved Values** |
| available | The amount of memory available. | Optional | memory-type-description | |
| maximum | The maximum amount of memory. | Optional | memory-type-description | |

### 3.1.10 Memory Type Description

This type of object represents the description that can be used to define the amount, unit, and usage type of some memory.

| Frame Ontology | memory-type-description<br>Fipa-Device | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Presence**[10] | **Type** | **Reserved Values** |
| amount | The amount of memory. This value must not be negative. | Optional | Integer | |
| unit | The unit used to express the amount of memory. | Optional | String | B<br>KB<br>MB |
| usage-type | The usage type of the memory. Either application, storage, or both. | Optional | Set of String | application<br>storage |

[8] While all of these parameters are optional, a valid user-interface object will contain at least one parameter.
[9] While all of these parameters are optional, a valid memory-description object will contain at least one parameter.
[10] While all of these parameters are optional, a valid user-interface object will contain at least one parameter.

### 3.1.11 Software Properties Description

This type of object represents the description that can be used to define the software capabilities of a given device.

| **Frame Ontology** | `sw-description` `Fipa-Device` |
|---|---|

| **Parameter** | **Description** | **Presence**[11] | **Type** | **Reserved Values** |
|---|---|---|---|---|
| `os` | Details of the operating system that the device has. | Optional | Set of `info-description` | |
| `agent-platform` | Description of the agent platform the device in question has. Can be used only if agent-compliancy of device level is either `true` or unspecified. | Optional | Set of `ap-description`[12] | |

[11] While all of these parameters are optional, a valid `sw-properties` object will contain at least one parameter.
[12] The frame for `ap-description` is found in [FIPA00023].

# 4 References

[CC/PP] Composite Capabilities / Preference Profiles. http://www.w3.org/Mobile/CCPP/

[FIPA00014] FIPA Nomadic Application Support Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00014/

[FIPA00023] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00023/

[UAProf] User Agent Profile Specification. Wireless Application Protocol Forum Ltd., 1999. http://www.wapforum.org/

# 5 Informative Annex A — Profile of a Hypothetical Smart Phone

## 5.1 Profile Description

This section describes a profile that represents the hypothetical smart phone. The validation of this profile is based on the `Fipa-Devices` ontology.

The following terms are used to describe the objects of the domain:

**Profile**. This is the mandatory name of this entity that must be used to represent each instance of this class.

**Ontology**. This is the name of the ontology, whose domain of discourse includes the parameters described in the table.

**Parameter**. This is the mandatory name of a parameter of this profile.

**Value.** This is the value given to a parameter.

### 5.1.1 SmartPhone xyz

Here the profile of the hypothetical SmartPhone xyz is presented.

| **Profile Ontology** | fipa.profiles.device.smartphonexyz<br>Fipa-Device |
|---|---|

| **Parameter** | | | | | | **Value** |
|---|---|---|---|---|---|---|
| info-description | name | | | | | SmartPhone |
| | vendor | | | | | Smartphones Ltd. |
| | version | | | | | xyz |
| type | | | | | | mobile-phone<br>PDA<br>GPS |
| agent-compliancy | | | | | | true |
| hw-description | connection-description | info-description | name | | | Bluetooth |
| | | | version | | | x.x |
| | connection-description | info-description | name | | | Infrared Data Association |
| | | | version | | | y.y |
| | connection-description | info-description | name | | | High Speed Circuit Switched Data |
| | | | version | | | z.z |
| | ui-description | screen-description | width | | | 500 |
| | | | height | | | 800 |
| | | | unit | | | mm |
| | | | resolution-description | width | | 1024 |
| | | | | height | | 768 |
| | | | | unit | | pixels |
| | | | | bpp | | 32 |
| | | | | graphics | | true |
| | | | color | | | true |
| | | audio-input | | | | true |
| | | audio-output | | | | true |
| | memory-description | memory-type-description | amount | | | 8 |
| | | | unit | | | MB |
| | | | usage-type | | | storage |
| | | memory-type-description | amount | | | 3856 |
| | | | unit | | | KB |
| | | | usage-type | | | storage |
| | cpu | | | | | 64-bit ARM9-based RISC |
| sw-description | info-description | | name | | | SmartOS abc |
| | | | vendor | | | ABCVendor Corp. |
| | | | version | | | 8.1 |
| | agent-platform[13] | | name | | | FIPA-OS v2.1.1 |
| | | | dynamic | | | true |
| | | | mobility | | | true |

The values on the rightmost column can change at any time. For example, if extra memory is inserted to the device or if another version of operating system is installed, the values for those parameters change. The parameters themselves, however, are more static. They stay the same despite the changes in single device profiles, since they are defined in the `Fipa-Device` ontology that is independent of them.

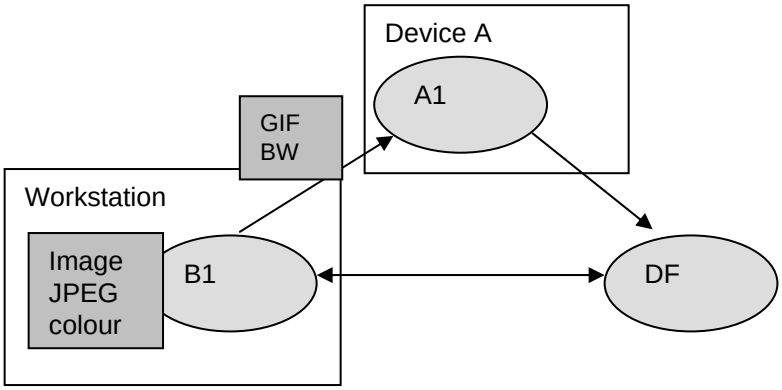
The values for parameters can be further divided into static and dynamic depending on the ability to change them in runtime. For example agent-compliancy and memory-type-description describing the memory available can change without booting the device. Hence they are dynamic information. On the other hand, screen-description and CPU are static information; they cannot change while the machine is running.

[13] The ontology against which this parameter is validated is found in [FIPA00023].

# 6 Informative Annex B — Examples

Annex B presents examples and use cases for device profiles based on the device ontology. The term agent is used to depict any software entity capable of reasoning over the profile, and the term DF or Directory Facilitator is used to depict a general directory service.

## 6.1 Content Adaptation I




Agent A1 sends its device profile to DF and registers to the system. Agent B1 interacts with agent A1 residing on device A. Agent B1 queries A's device profile either from the DF or directly from device A. Agent B1, which aims to send an image (640x480x24bits) to the user, analyses the device profile user interface capabilities:

| | | | | | |
|---|---|---|---|---|---|
| hw-description | ui-description | screen-description | width | | 2.26 |
| | | | height | | 3.02 |
| | | | unit | | inch |
| | | | resolution-description | width | 320 |
| | | | | height | 240 |
| | | | | unit | pixels |
| | | | | bpp | 4 |
| | | | color | | false |
| | | audio-input | | | true |
| | | audio-output | | | true |

| | | |
|---|---|---|
| sw-description | supported-mime-types | text/html<br>image/gif<br>image/wbmp<br>text/ascii |

The device operating system (or browser) is capable of handling ACSII text, html and also supports the GIF and Windows BMP mime-types. The agent reads from the device profile that the target device has a greyscale display and reduces the colours of the image to 4 greyscales (dithering), because it is not reasonable to send large images with excess unusable bits. The image size is reduced from 640x480 to 320x240 to fit the device's small screen.

In order to adapt the dialogue between agents, the dialogue service needs knowledge about the human-agent interface, especially information about the input and output capabilities of devices. For instance, if the user is using pen based input or touch-screen, the service may rely more on image maps to trigger actions, and if the user is interacting with keyboard, the service might use more text based input.

Now the same example is presented in more detail and using FIPA ACL. However, mime-type treatment is excluded.

1. The agent residing at a mobile device named *dummy* (A1 in the picture above) registers with the DF:

```
(request
  :sender
    (agent-identifier
      :name dummy@foo.com :addresses (sequence iiop://foo.com/acc))
  :receiver (set
    (agent-identifier
      :name df@foo.com :addresses (sequence iiop://foo.com/acc)))
  :language FIPA-SL0
  :protocol FIPA-Request
  :ontology FIPA-Agent-Management
  :content
    (action
      (agent-identifier
         :name df@foo.com :addresses (sequence iiop://foo.com/acc))
      (register
        (df-agent-description
          :name
           (agent-identifier
            :name dummy@foo.com
            :addresses (sequence iiop://foo.com/acc))
          :protocol (set FIPA-Request FIPA-Query)
          :ontology (set FIPA-Device)
          :language (set FIPA-SL0 KIF)
          :services (set
            (service-description
              :name device
              :type device-stuff
              :ontology (set FIPA-Device)))))))
```

2. Then, the agent *velmu* (B1 in the picture above) searches with the DF for a list of agents that support `Fipa-Device` ontology:

```
(request
  :sender
    (agent-identifier
      :name dummy@helluli.com
      :addresses (sequence iiop://helluli.com/acc))
  :receiver (set
    (agent-identifier
      :name df@foo.com
      :addresses (sequence iiop://foo.com/acc)))
  :language FIPA-SL0
  :protocol FIPA-Request
  :ontology FIPA-Agent-Management
  :content
    (action
      (agent-identifier
        :name df@foo.com
        :addresses (sequence iiop://foo.com/acc))
      (search
        (df-agent-description
          :ontology (set FIPA-Device)
          :language (set FIPA-SL0))
        (search-constraint :min-depth 2))))
```

3. *velmu* gets an answer, that dummy at foo.com supports `Fipa-Device` ontology:

```
(inform
  :sender
    (agent-identifier
      :name df@foo.com
      :addresses (sequence iiop://foo.com/acc))
```

```
      :receiver (set
        (agent-identifier
          :name velmu@foo.com
          :addresses (sequence iiop://foo.com/acc)))
      :language FIPA-SL0
      :protocol FIPA-Request
      :ontology FIPA-Agent-Management
      :content
        (result
          (action
            (agent-identifier
              :name df@foo.com
              :addresses (sequence iiop://foo.com/acc))
            (search
              (df-agent-description
                :ontology (set FIPA-Device)
                :language (set FIPA-SL0))
              (search-constraint :min-depth 2))))
           (set
            (df-agent-description
              :name
                (agent-identifier
                  :name dummy@foo.com
                  :addresses (sequence iiop://foo.com/acc))
              :ontology (set FIPA-Device)
              :languages (set FIPA-SL0 KIF)
              :protocol (set FIPA-Request FIPA-Query)
              :services (set
                 (service-description
                   :name device
                   :type device-stuff
                   :ontology (set FIPA-Device)))))))))
```

4. *velmu* aims to send an image (640 x 480 x 24 bit) to the device where dummy is located: *velmu* queries the dummy in order to find out the capabilities of device in which dummy is located:

```
(query-ref
  :sender
    (agent-identifier
      :name velmu@foo.com
      :addresses (sequence iiop://helluli.com/acc))
  :receiver (set
    (agent-identifier
      :name dummy@foo.com
      :addresses (sequence iiop://foo.com/acc)))
  :language FIPA-SL0
  :protocol FIPA-Query
  :ontology FIPA-Device
  :content
    (iota ?x (FIPA-Device :hw-description ?x)))
```

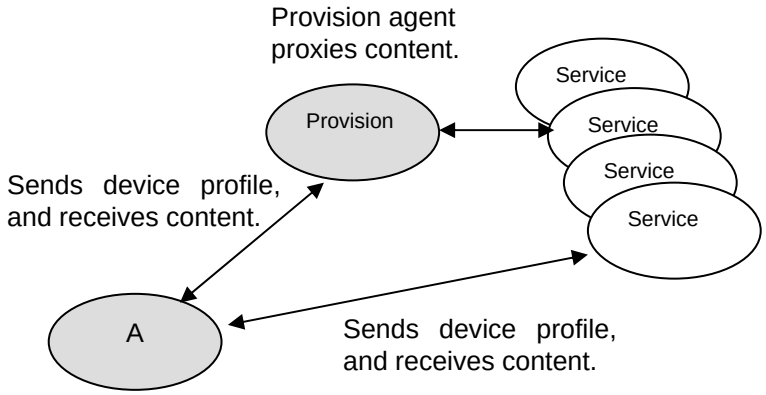
5. *dummy* sends appropriate information:

```
(inform
  :sender
    (agent-identifier
      :name dummy@foo.com
      :addresses (sequence iiop://foo.com/acc))
  :receiver (set
    (agent-identifier
      :name velmu@foo.com
      :addresses (sequence iiop://helluli.com/acc)))
  :language FIPA-SL0
```

```
      :protocol FIPA-Query
      :ontology FIPA-Device
      :content
        (= (iota ?x (FIPA-Device :hw-description ?x))
          (hw-description
            :cpu “i286”
            :ui (set
              (ui-description
                :screen
                  (screen-description
                    :width 57
                    :height 78
                    :unit mm
                    :color false
                    :resolution (set
                      (resolution-description
                        :width 320
                        :height 240
                        :unit pixels
                        :bpp 4
                        :graphics true))
                :audio-input true
                :audio-output true)))))
```

*velmu* analyses the information, and finds that the target device has a greyscale display and reduces the colours of the image to four greyscales (dithering), because it is not reasonable to send large images with excess unusable bits. Furthermore, the image size is reduced from 640 x 480 to 320 x 240 to fit the device’s screen.

## 6.2 Content Adaptation II

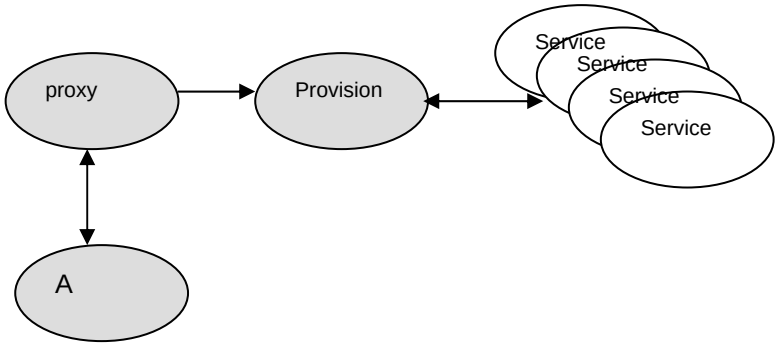



A new client logs in to an agent service domain providing tourism services. The service provision agent receives the device profile from the device software system accessing the agent-based services using ACL. The provision agent first stores the profile into a local cache (for example, CC/PP caching) and then checks the services available for this particular type of client. The device profile indicates that the device is part of an agent platform, which makes it eligible to access directly all of the agent based services, depending on whether or not it hosts or is capable of hosting the correct interface agents or layers. The agent on the device may contact the service agents directly and send the device profile for adaptation.

| | | | | |
|---|---|---|---|---|
| type | | | | PDA<br>GPS |
| agent-compliancy | | | | true |
| hw-description | connection-description | info-description | name | GPRS |
| | | | version | x.x |
| | memory-description | memory-type-description | amount | 8000 |
| | | | unit | KB |
| | | | usage-type | application |
| | | memory-type-description | amount | 4000 |
| | | | unit | KB |
| | | | usage-type | application |
| sw-description | agent-platform | | name | FIPA-OS v2.0 |
| | | | dynamic | false |
| | | | mobility | false |

However, the client profile does not specify any streaming codecs in the sw-description frame that the services support, so the provision agent excludes all streaming services from the service list when the client requests it.

## 6.3 Content Adaptation III




Another client is not capable of hosting an agent platform or being a part of an existing platform, but hosts browser software that supports html content with streaming audio. The specific output capabilities of the browser are extracted from the sw-description extension fields.

The client contacts the provision agent through a proxy that, using some proprietary format, accepts the device profile. Now, the provision agent has to exclude those services that cannot be accessed using proxies that mediate between non-agent and agent based resources.

## 6.4 Service Advertisement and Software Updates

The Provision agent may detect that a new service, which is compatible with a new XYZ Communicator, has become available. The new product is based on Java Midlet technology, and supports the downloading of new software (jar-files). Now, when clients using the XYZ device log into the system, they are displayed (if their user profile allows it) information about the new service. The system checks the sw-description frame extension fields for Java environment and the device name and version from the info-description frame.

| | | |
|---|---|---|
| info-description | name | XYZ Communicator |
| | vendor | Smartphones Ltd. |
| | version | xyz |

| | | | |
|---|---|---|---|
| sw-description | java-env | configuration | CLDC-1.0 |
| | | profile | MIDP-1.0 |
| | | locale | en-US |
| | supported-mime-types | | text/vnd.sun.j2me.app-descriptor |

# 7 Informative Annex C — Usage of FIPA Device Ontology through CC/PP

A technology called CC/PP (Composite Capabilities/Preference Profiles) is developed in W3C [CC/PP]. The frames in this specification received some of their concepts from CC/PP specifications. There are, however, differences and this is mainly due to the different goals of FIPA and W3C.

For example, in CC/PP the ontology is divided into three following categories at the highest level: Terminal Hardware, Terminal Software and Terminal Browser. Of these only Terminal Hardware and Terminal Software were adopted here. Terminal Browser was left out because FIPA is not as focused to www as W3C is. On the other hand, in this specification there is a parameter called agent-compliancy that is not found in CC/PP specifications [CC/PP]. The value of agent-compliancy parameter informs whether the device in question is capable of hosting one or more FIPA agents or not.

Despite the differences between the approaches the FIPA-device ontology could be used in a CC/PP profile. This can be accomplished in a similar fashion as with UAProf (see [CC/PP]). So, if a developer wants to inform that some device is FIPA-compliant, then it can be achieved with a CC/PP profile as follows:

```
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:ccpp="http://www.w3.org/2000/07/04-ccpp#"
     xmlns:fipa="http://www.fipa.org/profiles/device-20010202#">
     xmlns:uaprof="http://www.wapforum.org/UAPROF/ccppschema-19991014#">

  <Description about="http://www.foo.com/profiles/ProfileX">
    <ccpp:component>
      <Description about="http://www.foo.com/TerminalHardware">
        <type resource="http://www.foo.com/Schema#HardwarePlatform" />
        <ccpp:Defaults rdf:resource="http://www.foo.com/profiles/hwproperties" />
        <fipa:compliancy>true</fipa:compliancy>
      </Description>
    </ccpp:component>

    <ccpp:component>
      <Description about="http://www.foo.com/TerminalSoftware">
        <type resource="http://www.foo.com/Schema#SoftwarePlatform" />
        <ccpp:Defaults rdf:resource="http://www.foo.com/profiles/swproperties" />
        <fipa:ap-description>FIPA-OS v2.1.1</fipa:ap-description>
      </Description>
    </ccpp:component>

    <ccpp:component>
      <Description about="http://www.foo.com/Browser">
        <type resource="http://www.foo.com/Schema#BrowserUA" />
        <ccpp:Defaults rdf:resource="http://www.foo.com/profiles/browserproperties" />
        <uaprof:BrowserName>Internet Explorer</uaprof:BrowserName>
        <uaprof:BrowserVersion>5.0</uaprof:BrowserVersion>
      </Description>
    </ccpp:component>
  </Description>
</RDF>
```

Here the `fipa-namespace` is used to refer that the device characterized in ProfileX is FIPA-compliant and that the agent platform it has is the same FIPA-OS v2.1.1 used earlier as an example. Other CC/PP –defined properties are (supposedly) found in the URI's declared in `rdf:resource` attributes of the `ccpp:Defaults` elements. Agent compliancy seems to be the property that most clearly distinguishes the ontology and profiles presented in this paper from the comparable ones defined in W3C and WAP Forum.

The namespace declaration in the fourth row defines a URI that should contain a CC/PP schema (`http://www.fipa.org/profiles/device-20010202#`). The schema in that location corresponds to the

ontology presented in this paper, but in CC/PP terms. More specifically, there are specified only those elements that are not found in CC/PP schema itself. FIPA Agent-compliancy is naturally an example of these.