

携帯端末からネットワーク機器群を操作するための インターフェースの提案

4U-03

木下信幸* 溝口文雄*
東京理科大学 理工学部†

1 はじめに

ステーションサーバ技術はサービスディレクトリからサービス定義を取得しホームネットワーク内の機器群の遠隔操作を可能にする[1]。サービス定義は機器群やサービス群の操作コマンドを包含するXMLファイルである。操作コマンドはサービスを操作するためのコマンド名およびコマンドの実装であるJavaクラスパス、要求引数等の要素からなるが、これまでは携帯端末からの利用における操作性を考慮していなかったため、携帯端末の操作の際は必ずしも扱いやすいインターフェースを提供することができなかった。そこで本論では携帯端末の画面サイズやテキスト表示数を加味し、携帯端末からの操作をより扱いやすくするインターフェースを提案する。

2 サービスディレクトリ

サービスディレクトリはホームネットワーク内のサービス群や機器群をディレクトリ構造で表したもので頂点をステーションサーバとし、サービスや機器、サービス定義をノードとするサービスネットワークの論理構造を表現する。サービスディレクトリにおいて、あるノードが外部から操作が可能なサービスや機器を表す場合、そのノードは操作ロジックを包含したXMLファイル(サービス定義)の子ノードとしてもつ。

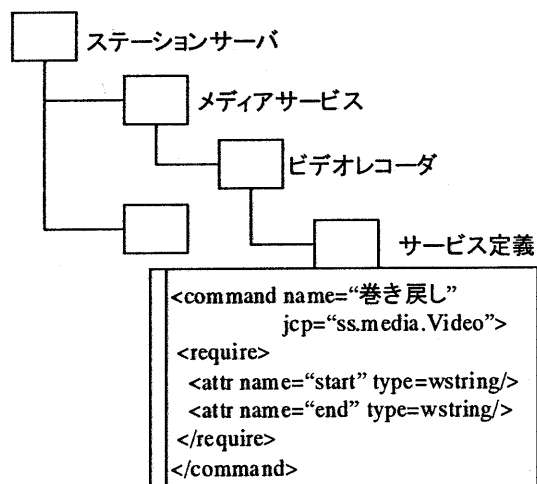


図1: サービスネットワーク

*Nobuyuki Kinoshita, Fumio Mizoguchi

† Faculty of Sci. and Tech. Science University of Tokyo

3 サービス定義

サービス定義はサービス名、サービスXMLファイル名、操作コマンドといったサービスに関連する情報を定義するXMLファイルである。ステーションサーバはユーザエージェントの要求に応じるためにサービス定義を解析し必要な前処理をした後、応答HTMLファイルを出力する。サービス定義にはサービスの対象となる機器群へのAPIを定義したJavaクラスへのパスや操作コマンドの定義などサービスを実行するのに必要なすべての情報を含んでいる。

4 操作コマンド

操作コマンドはサービスを操作するコマンド群を表し、サービス定義内のCOMMANDタグにより定義される。ステーションサーバはユーザエージェントの要求に応じるためにサービス定義から操作コマンドのリストを生成し応答HTMLファイルを出力する。操作コマンドに引数が必要な場合は2つの表現がある。1つは必須の引数の記述であり、<require>タグの子ノードとして<attr>タグを定義するものである。もう1つはオプションの引数の記述であり<command>タグの子ノードとして<attr>タグを定義するものである。ユーザエージェントの要求に必須引数をもつコマンドの実行が伴う場合、ステーションサーバは必須引数が指定されているかをチェックし、必須引数に値が代入されるかあるいはコマンドの実行が取り消されるまでユーザエージェントに必須引数の代入を促す。図2は<command>タグのDTDを表す。図2:コマンドタグのDTD

<pre><!DOCTYPE COMMAND [<!ELEMENT command (attr require) * > <!ATTLIST command name CDATA "na" jcp CDATA "ss.Error" > <!ELEMENT attr (#PCDATA) ></pre>	<pre><!ATTLIST attr name CDATA "na" type CDATA "na" > <!ELEMENT require (attr) * >]></pre>
--	--

5 携帯端末の制限

外部ネットワークから携帯端末を用いて内部ネットワークの機器を快適に扱うことはステーションサーバが携帯端末に特化した応答を返すことを意味する。携帯端末に特化した応答HTMLファイルを出力するためには多様な携帯端末群の1つ1つについてその制限を考慮しなければならないが、本論では最も制限の厳しいi-modeに対応した応答HTMLファイルを出力するようにステーションサーバを更新する。i-modeに対応したHTMLは以下の制限がある。

- 画面のサイズを考慮した文字数、テキスト表示を考える
- イメージは2値のGIFファイル
- 一画面の容量は最大5K
- URLエンコード後の文字長は最大200バイト
- 直接入力可能なURL長はそれぞれ最大100バイト

上記の制限を元に応答HTMLファイルの加工を行い携帯端末に特化したインターフェースを出力できるよう変更を加える。

6 ステーションサーバの更新

ステーションサーバの変更は2段階に大別される。

1段階目はステーションサーバがユーザエージェントが携帯端末かどうかを判断するロジックを組み込む変更である。これはクライアント要求内の環境変数HTTP_USER_AGENTを解釈することにより実現される[3]。この環境変数内に携帯端末の大部分は端末名に関する文字列を含有している。ステーションサーバはこの環境変数内の端末を特定する文字列からクライアントが携帯端末かを特定する。

2段階目はサーバ定義ファイル内の応答HTMLファイルを構成する文字列に携帯端末に特化した新たなタグを追加する変更である。これは関連変数に対して“pda_”というプレフィックスを対応した変数を定義することで実現される。たとえば<command>タグの属性nameは応答HTMLファイルに記述される文字列を含むから、属性pda_nameを新たに定義して携帯端末に特化した記述を加えるといった具合になる。これによりステーションサーバは携帯端末に特化した文字列を得ることができる。図3は上記のユーザエージェントとステーションサーバ、サービスディレクトリの関係を表したものである。

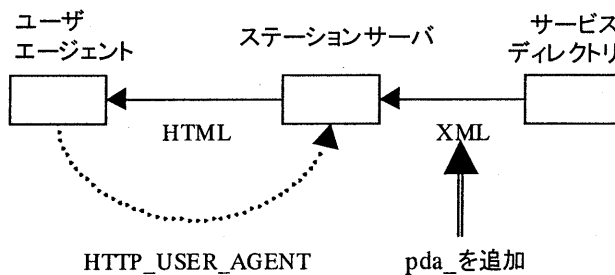


図3:ステーションサーバの変更

7 コマンド入力

図4はコマンド入力時のインターフェース

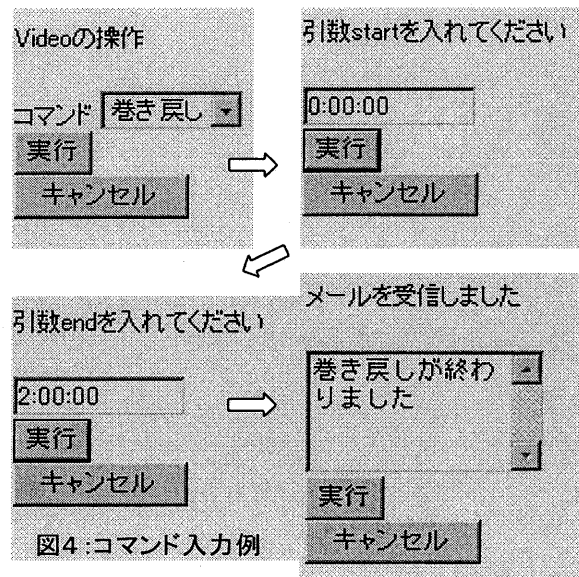


図4:コマンド入力例

の例を表したものである。上記はiModeSS[4]を用いてIE5.0上で得た出力を貼付けている。

8 おわりに

本論ではステーションサーバを用いて外部からサービスネットワーク内のサービスを携帯端末により操作する際に問題となる扱いにくさを解消するインターフェースを提案した。このインターフェースはステーションサーバがユーザエージェントの端末の種類を特定し、その端末により適した応答HTMLファイルを出力することにより実現される。本論ではi-modeに特化した例をあげたがこの方法は他の端末にも応用することが可能である。

参考文献

- [1] 木下信幸、溝口文雄、ステーション・サーバによるネットワークデバイスの操作モデル、日本ソフトウェア科学会 第17回大会
- [2] 木下信幸、溝口文雄、Station サーバの実装、情報処理学会 第61回全国大会
- [3] i モード対応のホームページを作ろう <http://www.watch.impress.co.jp/internet/www/article/1999/0510/imode.htm>
- [4] 川原 千種、iModeSS, <http://www.k-collect.net/imodess/main.htm>