

即興的な分散スマート環境間連携機構

藪下直哉^{†1} 宇佐美則明^{†2}
新井イスマイル^{†3} 西尾信彦^{†4}

近年、ネットワークを介して様々な情報機器やセンサなどを連携させることによって、利用者に利便性の高いサービスを提供する空間、スマート環境に関する研究が盛んである。また、単一のスマート環境内でのサービス利用におさまらず、複数のスマート環境を相互利用することが出来れば、より便利である。しかし、分散したスマート環境間連携を実現するためには、利用者は様々な事前設定を強いられる。さらに、連携先のスマート環境内のサービスを IP アドレスなどのサービスが動作するノードを特定できる情報なしで指定するのは難しく、利用者にとって容易にサービス連携を実現するには至らなかった。本稿では、分散スマート環境間連携のための要件を示し、それを満たすような分散スマート環境間連携機構を提案する。本稿の提案する連携機構では、利用者は、分散スマート環境間で、利用したい状況・環境を抽象的に記述する。この抽象的な記述を本稿ではシナリオと呼ぶ。本システムではシナリオを解析し、利用者の記述した通りのサービス連携を行うことを可能にする機構を構築する。

Extemporaneous Collaboration Mechanism for Distributed Smart Spaces

NAOYA YABUSHITA,^{†1} NORIAKI USAMI,^{†2}
ISMAIL ARAI^{†3} and NOBUHIKO NISHIO^{†4}

Recently, there has been significant amount of research on Smart Spaces in which information devices and sensors are linked together through a network to provide useful services to users. It is useful if one could use services not only in a single Smart Space, but also in multiple Smart Spaces. Typically, A user has to preconfigure the services before the collaborative services are ready to use. However, it is difficult to do configurations without knowing the details of each individual devices, such as IP address. In this paper, we propose a collaboration mechanism for distributed smart spaces. The user writes an abstract description of the collaborative service they want. Given the description, or a "scenario," the proposed system reads it and organizes the environment accordingly.

1. はじめに

近年、ネットワークを介して様々な情報機器やセンサなどを連携させることによって、利用者に利便性の高いサービスを提供する空間、スマート環境が実現されてきており、今後は一般家庭など、様々なところでスマート環境が構築されることが想定される。それに伴い単一のスマート環境内でのサービス利用だけでなく、複数のスマート環境間に跨って、相互にサービスを利用することができることが期待される。例えば、複数の分散スマート環境間でテレビ会議等の協調作業を行う際、自動的にプレゼンテーション画面が会議に参加しているユーザのノート PC 等の画面に表示されることや、会議の流れに合わせてそれぞれのスマート環境上のライトの電源やスピーカのミキサーを制御したりできれば便利である。よって、複数のスマート環境内で構築されたサービスを、相互に利用・連携させて、状況に応じて自動的に構成することが出来ればより利便性が高まると言える。

一方で、一般にスマート環境は、安全性の面から NAT やファイアウォールで隔離されたプライベートネットワーク上に構築されることが多いため、外部からの通信を制限した接続性の問題がある。よって、状況に応じて分散したスマート環境間でサービスを連携させるためには、利用したいサービスに対し、自動的に遠隔プライベートネットワーク間における通信を確保する必要がある。さらに状況に応じて分散スマート環境上のサービスが動的に動作することが望まれる。

そこで、本稿の提案する連携機構では、利用者は、分散スマート環境間で、利用したい状況・環境を抽象的に記述する。この記述したものを本稿ではシナリオと呼ぶ。本システムではシナリオを解析し、利用者の記述した通りのサービス連携を行うことが可能である。

本稿では 2 章で現在の分散スマート環境間におけるサービス利用の問題と要件を述べ、3 章で要件を満たす分散スマート環境間連携機構の提案をする。4 章で本稿が提案する手法を

^{†1} 立命館大学 理工学研究科 情報理工学専攻
Department of Computer Science, Ritsumeikan University

^{†2} トヨタ自動車 (株)
Toyota Motor Corporation

^{†3} 立命館大学総合理工学研究機構
The Research Organization of Science and Engineering, Ritsumeikan University

^{†4} 立命館大学 情報理工学部
College of Information Science and Engineering, Ritsumeikan University

実現する為の設計を述べ、5章で実装を述べる。1章で評価を述べ、7章で本研究との関連研究を述べる。最後に8章で本稿のまとめと今後を述べる。

2. 分散スマート環境間連携の問題と要件

スマート環境は、プライベートネットワーク上に構築されるため、NATやファイアウォールによって外部からの通信が制限される等の問題があり、利用者は、利用したいサービスに対し、接続性を確保する為に、NATやファイアウォールを超えてネットワーク間を接続する設定を強いられる。また、連携先のスマート環境内での利用したいサービスの有無やIPアドレスなど具体的な情報を事前に把握しておく必要があり、これらの作業は利用者にとって大きな負担である。

以上の問題を解決する為に必要であると考える要件を述べる。

要件1 利用したいサービスへの自動的な接続性の確保

連携先のスマート環境のサービスに対し、接続性を確保するためには、NATやファイアウォールを越える必要がある。しかし、利用したいサービスの数だけ逐一接続性を確保する設定をするのは利用者にとって負担である。そこで、利用したい状況に応じて、自動的にサービスへの接続性を確保することが必要である。

要件2 利用したい状況に応じてサービスが動作

例えば、複数の分散スマート環境間でテレビ会議を行う際、自動的にプレゼンテーション画面が会議に参加しているユーザのノートPCの画面やプロジェクトを通してスクリーンに表示されることや、会議の流れに合わせてそれぞれのスマート環境上のライトの電源やスピーカのミキサーを制御したりできれば便利である。ビデオ会議が始まる度に細かい設定をすることなく利用したい状況に応じてサービスが動作する必要がある。

要件3 利用したいサービスの特定

連携先のスマート環境のどのサービスを利用したいのかを、IPアドレスなど具体的な情報を知らなくとも容易に特定できる必要がある。

要件4 利用したい環境や状況の設定の簡易化

仮に要件1~3を満たすことができても、とても時間のかかるものであったり、利用者にとって作業が煩雑になれば有用性のあるものとは言えない。要件1~3をいかに利用者にとって煩雑な作業にならないようにするか、また、分散スマート環境間で、迅速に利用したいサービス連携を実現できるかといったことが重要である。

3. 提案手法

2章で述べた要件を満たす為に、実現したい状況・環境をシナリオに記述し、そのシナリオを読み込ませるだけで、記述内容通りのサービス連携が実現できる手法を提案する。以下に本手法で提案するシナリオについて述べる。

シナリオに記述する内容は以下の項目が挙げられる。

・ネットワーク環境

分散スマート環境間以外にもシナリオベースでサービスを利用することも想定される。ここでは、どのような環境でシナリオを利用するかを記述することで、単一スマート環境内でのみ利用であるのか、分散スマート環境間でサービスを相互利用するのかをシステムに伝える項目である。

・連携先スマート環境

どのスマート環境と連携するかを記述しておくことで、どのスマート環境に対し、接続性を確保する処理をするのか、どのサービスをどのように動作させるのかといった要求が行える。

・利用するサービス指定

どのサービスを利用するかを指定する必要がある。もし、ネットワーク環境が分散スマート環境間であれば、利用したいサービスに対し接続性を与える処理を行う必要もある。さらに、利用者が利用したいサービスのIPアドレスなど、サービスを特定する詳細な情報を知ることなくサービスを指定できることが望まれる。そこで、本研究では、抽象的なサービス名の記述のみでサービスを指定する機構を構築する。

・サービスの動作する条件とその動作内容

どのサービスがどのような条件のもと、どのような動作をするのかを記述する必要がある。ただし、シナリオは時間系列に従って記述される必要がある。例えば、サービスAがサービスBの終了と同時に動作させたいと記述する。そのような状況がシナリオの中で複数回実現される場合、サービスAの動作内容が毎回異なると、どの動作を実現させるべきかを判断することが困難である。

・シナリオの終了する条件

シナリオが終了する条件を記述しておくことで、その条件を満たしたときに動作させていたサービスの停止、または連携先スマート環境内のサービスへの接続性の解除などの処理を行う。

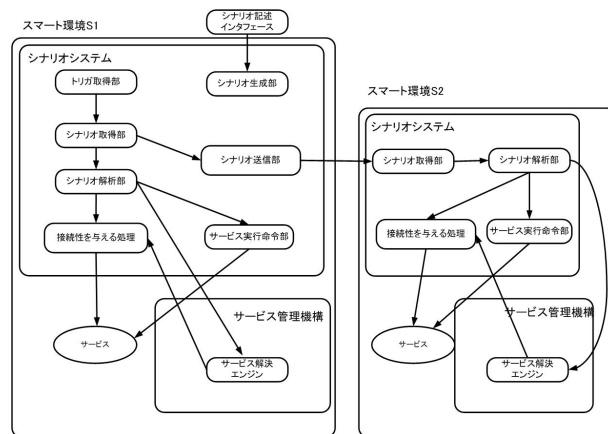


図 1 システム設計

本稿で提案するシナリオにより、必要なサービスに対し、分散スマート環境間で接続性を与える設定を自動的に実行し、利用したいサービスが状況に応じて動作する。また、シナリオは状況に応じて記述することが可能であり、複数のシナリオを常に準備することが可能である。さらに、複数のシナリオの中からどのシナリオに記述された状況・環境を実現させるのかを判断することは非常に重要であり、例えば、スマート環境 S1 に属するユーザ A とユーザ B、スマート環境 S2 に属するユーザ C とユーザ D がそれぞれの属するスマート環境の会議室の席に座るとテレビ会議が始まるというような条件を定義し、その条件を満たすとどのシナリオを実現するのかを送信するようなアプリケーションを構築することで、後は自動的に分散スマート環境間で利用したい環境を構築することが可能である。また、IP アドレスなどの具体的な情報を事前に知らずに抽象的なサービス名で指定したとしても、どのサービスを動作させるべきかを判断するサービス解決エンジンを実装することで、容易に連携先のスマート環境内のサービスを利用可能にする。

4. 設 計

本研究のシステム設計図を図 1 に示す。

4.1 トリガ取得部

本システムでは、利用者が直接シナリオをシステムに送信するのではなく、複数あるシナリオの中からシステムがどのシナリオを実現するのかを判断する。そこで、シナリオを実現

させるトリガを別アプリケーションからの通知を受け取ることで可能にしている。別アプリケーションは、本システムにトリガとなるキーワードを送信し、そのキーワードにそってシナリオが決定される。

4.2 シナリオ取得部

トリガ取得部からどのシナリオを実現するのかを受け取りシナリオを取得する。取得したシナリオはシナリオ解析に渡す。また、別アプリケーションからの通知以外にも、利用者の利用したい時にシナリオをシステムに送信することも可能である。その際は、シナリオ送信用のインタフェースから、選択したシナリオを送信すると、シナリオ取得部がシナリオを受け取る。

4.3 シナリオ解析部

トリガ取得部がアプリケーションから通知を受けると、シナリオ取得部に取得すべきシナリオを伝える。シナリオ取得部はシナリオ解析に取得したシナリオを渡す。シナリオ解析で受け取ったシナリオを項目別に分類する(シナリオ解析処理の詳細を図 2 に示す)。大きく項目は 5 つに分類することができ、詳細は 4 節にて述べた通りである。

まずネットワーク環境であるが、ネットワーク環境要素の値が分散スマート環境である場合、シナリオに記述されているサービスに対してのみ、連携先スマート環境との接続性を与える処理をする必要がある。また、その際に、連携先のスマート環境にもシナリオを送信する必要がある。そこで、連携先スマート環境の宛先にシナリオを送信する。

次に、シナリオに記述されているサービス名を、サービス解決エンジンに問い合わせ、もっとも近いサービスを判断し、以降の処理に利用する。また、サービス解決エンジンに、各サービスに対する入出力関係などを登録しておくことで、必要なサービスが欠けている場合でも、利用可能なサービスを代替する機能を備えている。サービスを特定できる IP アドレスなどの情報を管理しており、シナリオ記述時には IP アドレスなどを意識せずに記述することが可能である。

続いてサービスの動作であるが、シナリオに記述された条件を要素に分類し、その要素に適した処理を行う。条件の要素の値をもとに、比較を行い、条件を満たしていた場合は記述されているサービスを、どのように動作させるのかをサービス実行命令部に渡す。

最後に、シナリオの終了する条件であるが、シナリオのサービス動作の部分は時間系列にそって記述する必要がある。例えば、シナリオの終了する条件がサービス A の動作停止後であるとする。ここで、サービス B の動作開始条件もサービス A の動作停止後であれば時間系列にそって記述しない限りシステムがどちらを優先させるべきかを判断することは困難

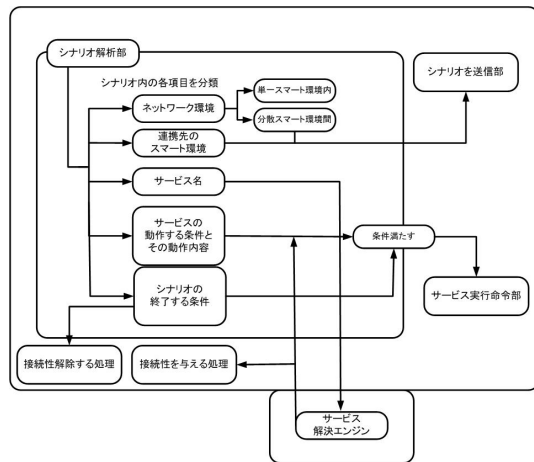


図 2 シナリオ解析処理

である。よって、シナリオの終了条件は、サービス動作の条件とその動作内容の最後に記述されている要素とその値をもとにシナリオを自動的に終了させる。ここで、シナリオの終了とは、シナリオに記述されているサービスの動作終了や、分散スマート環境間における接続性の解除を意味する。

4.4 サービス解決エンジン

スマート環境内のサービスを一括管理し、シナリオに記述されているサービス名に対して、どのサービスを動作させるべきかを判断する。また、各サービス上で動作しているホストの IP アドレスも管理しており、連携先のスマート環境との接続性を与える処理を行う際にこの IP アドレスを利用する。このサービス解決エンジンにより、シナリオ記述時に、抽象的なサービス名のみでのサービス指定が可能になる。この機能により、容易にサービスを特定でき、要件 3 が満たされるといえる。

4.5 サービス実行司令部

どのサービスをどのように動作させるのかを受け取り、各サービスに対して命令を送る処理を行う。サービス解決エンジンにより指定されたサービスに対し、操作命令を行う。サービス実行司令部では、シナリオに記述されているサービス動作条件を満たした場合に、動作内容に適した実行命令を出すので、シナリオに記述するだけで、後はシステムがどのタイミングでサービスを動作するのかを判断し、状況に応じたサービス動作を可能とする。この機

能により要件 2 を満たすことが可能である。

4.6 連携先のスマート環境内に存在するサービスとの接続

サービス解析とサービス解決エンジンにより、シナリオに記述されているサービスの特定や、そのサービスが動作するノードの IP アドレスを取得することが可能である。連携先のスマート環境内に存在するサービスとの接続性を確保する処理を行う際には、この IP アドレスとサービス名をもとに接続性を確保する設定を行う。この機能によりシナリオにサービスを記述するだけで、必要なサービスのみ接続性を与えることが可能になり、要件 1 を満たす。

5. 実 装

提案したシステムの実現可能性を示すために、Java 1.6 と、HTTP サーバである Jetty 6.1¹⁾ を用いてプロトタイプシステムを実装した。本システムの全体図を図 3 に示す。

・ Pool GW

本研究では、遠隔スマート環境間での接続性を与える為に、Peer Pool²⁾ を利用した。Peer Pool とは DNS クエリをインタフェースとして、各スマート環境に設置された Pool Gateway (Pool GW) と呼ばれるホスト間で構築される IP レイヤのオーバーレイネットワーク (共有空間ネットワーク) を、即時制御・構成するシステムである。Pool GW と同一ネットワークセグメントに存する任意のノードは、一定様式の DNS クエリを Pool GW に送ることによって共有空間ネットワークに参加することができる。また、共有空間ネットワークに参加した各ノードには、仮想的な IP アドレス (Pool アドレス) と .pool で終わるホスト名 (Pool ノード名) が割り当てられる。Peer Pool を利用することにより我々は NAT の背後にあるノード同士の通信を可能とした。よって、Pool GW 上で本システムを動作させることで、容易に連携先のスマート環境への接続性を与える処理をすると共に、シナリオを連携先のスマート環境に送信する役割もなすことが可能である。分散スマート環境間の Pool GW は OpenVPN³⁾ を利用し、分散スマート環境間の Pool GW 間のシナリオ送受信は、VPN アドレスに対し HTTP 通信を行うことで成立させている。

・ REST サーバ

また、我々は異なるミドルウェアによって構築されたサービスの非互換性の問題を、REST⁴⁾ に基づいてラップし、共通仕様の Web サービスとすることによって解決している⁵⁾。よって、シナリオ解析後に、動作条件を満たしたサービスは、どのようなサービス動作をするのかを REST サーバに送信する必要がある。ここでは、REST サーバで管理している WADL

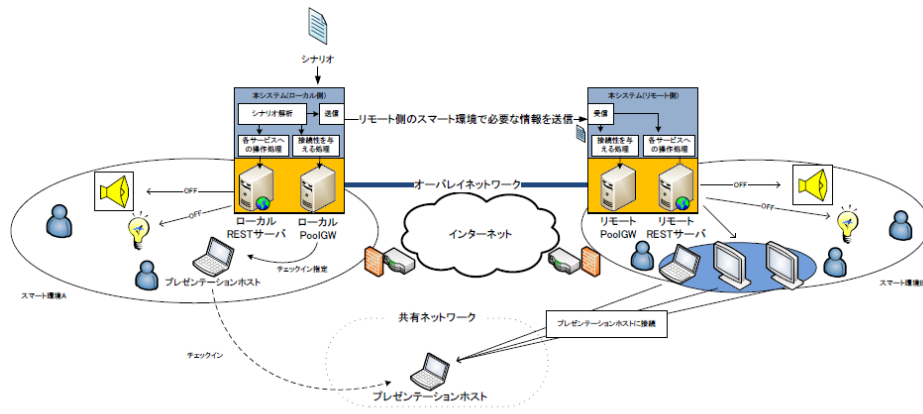


図 3 システム全体図

(Web Application Description Language)⁶⁾ からサービスのリソースパスとコマンドを取得し、動作内容に適したコマンドをリソースパスに対して PUT もしくは POST メソッドを送ることにより、サービスを動作させることが可能である。

また、サービス名を抽象的な名前だけで利用できるように、シナリオに記述されているサービス名に対し、管理しているサービスと近いであろうと判断したサービスを選択するサービス解決エンジン機能を持つ。この機能により、抽象的なサービス指定に対しても、期待通りのサービスの動作を保証する。

実際に分散スマート環境間でのサービスの動作を実現するために、スマート環境内の RESTful サービスとして、VNC を対象として実装した。VNC サービスを提供する各機器には RESTful サービスとしての HTTP リクエストを待ち受けるスタブを導入し、各機器には URL が割り当てられ、URL に対して POST リクエストを行なうと、機器の状態を変化させる (VNC Server を起動させている機器の画面を VNC Viewer の ListenMode を起動している機器の画面に VNC Reverse Connection 機能によって押し付けるなど) ことができる。

・シナリオ

現在シナリオは、柔軟に独自のタグを利用でき、表現の容易さや、扱いやすさなどの理由から XML 形式で記述している。図 4 にシナリオの記述フォーマットを示す。

```

<?xml version="1.0" ??
<scenario>
  <network_environment>ネットワーク環境</network_environment>
  <remotePoolGW>連携先のPoolGWアドレス</remotePoolGW>
  <service>
    <id>1</id>
    <name>サービス名</name>
  </service>
  <service>
    <id>2</id>
    <name>サービス名</name>
  </service>
  <service>
    <id>3</id>
    <name>サービス名</name>
  </service>
  <service_collaboration>
    <サービスの動作する条件と動作内容>
      <条件>サービスの動作条件</条件>
      <action>サービスの動作</action>
    </サービスの動作する条件と動作内容>
    <サービスの動作する条件と動作内容>
      <条件>サービスの動作条件</条件>
      <action>サービスの動作</action>
    </サービスの動作する条件と動作内容>
    <シナリオの終了する条件>
      <条件>シナリオの終了条件</条件>
    </シナリオの終了する条件>
  </service_collaboration>
</scenario>

```

図 4 シナリオ記述フォーマット

・シナリオ取得部インタフェース

本システムがシナリオを取得するタイミングは、別アプリケーションからの通知を受け取る以外に、利用者が好きなタイミングで連携することも可能である。そこで、本システムでは、シナリオ取得用に図 5 に示す WEB インタフェースを実装した。シナリオをファイルアップロードすると、シナリオ取得部に選択したシナリオが送られる。あとは、設計で述べた手順でシステムが動き、選択したシナリオ通りのサービス連携が実現される。

6. 評価と考察

6.1 評価

本章では、本研究で実装した機能について、各サービスへの接続性を与える操作負担の緩和や各サービス利用の際の操作負担の緩和についての成果を定性的に行った。また実装した



図 5 web インタフェース

機能の動作について定量的に評価を行った。その結果について述べる。

6.1.1 利用したいサービスへの接続性の確保

シナリオに分散スマート環境間連携であることを記述した際は、連携先のスマート環境内のサービスに対して接続性を与えたい。本研究では分散スマート環境間における接続性を確保するために Peer Pool を利用している。Peer Pool では、一定様式の DNS クエリを Pool GW に送ることによって共有空間ネットワークにチェックインすることができるが、必要なサービス数だけ毎度作業を行わなければならない。そこで、状況ごとのシナリオを用意することで、各利用者が状況に応じて逐一各ノードを共有空間ネットワークにチェックインする操作の手間がなくなった。

6.1.2 各サービス利用の際の操作負担の緩和

本システムでは、シナリオに利用したいサービスを抽象的に記述し、またそのサービスの動作条件と動作内容を記述することにより、逐一操作をする必要なく、期待通りのサービス連携が行える。また、サービス解決エンジンにより、利用したいサービスを特定する IP アドレスなどの情報を知らなくとも容易にシナリオに記述することが可能となった。しかし、現在のサービス解決エンジンでは、プロトタイプ実装での想定するサービスのバリエーションが乏しいため、よほどの外れなサービス名を記述しない限り、期待通りのサービスを動作することは必然に近い。よって、今後分散スマート環境間でのサービス連携の様々な可能性を考慮し、設計・実装を改めて試みる必要がある。

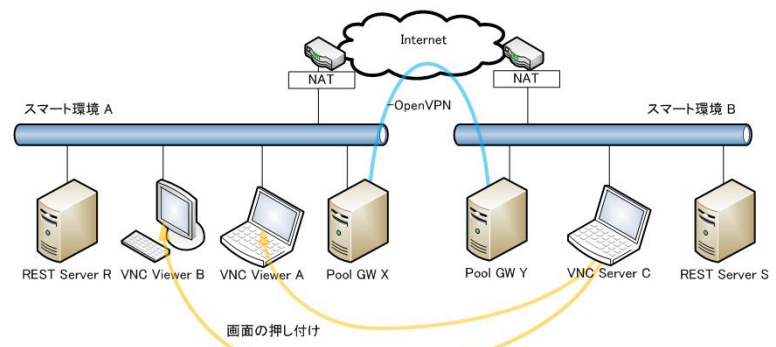


図 6 実験環境

6.1.3 各機能の処理時間および、シナリオを満たす処理時間

今回は、スマート環境 A に 2 台の VNC サービスとスマート環境 B に 1 台の VNC サービスを利用し、実験を行った。実験環境を図 6 に示す。ここで、スマート環境 B を連携先のスマート環境とする。スマート環境 B で VNC サービスが動作しているノードの画面をスマート環境 A で VNC サービスが動作しているノードの画面に押し付けるというシナリオを行った。以下に実験を 20 回行い、各機能の平均処理時間および平均総処理時間を表 1 に示す。

6.2 考 察

今回実装したプロトタイプでは、特定のノード間における分散スマート環境間での状況に応じたサービス連携を実現することができた。連携が行われるまでの時間も十数秒程度で構築できるため、十分に実用にたれる性能が出せ、利便性は高いと言える。今後は、より様々なサービスや条件のもと、期待通りにシナリオが実現できるかに焦点をおいて実験を行う必要がある。

7. 関連研究

本章では、本研究の関連研究・関連技術について述べる

- ・XML 形式のシナリオを用いたサービス合成手法の提案⁷⁾

この研究は、インターネット上に存在する、さまざまな Web サービスを合成するもので

表 1 機能の処理時間

項目	平均処理時間 (ms)	割合 (%)
トリガを取得してからシナリオを取得するのに要した時間	341.85	2.66
シナリオ解析に要した時間	16.30	0.13
サービス解決エンジンがどのサービスかを判断するのに要した時間	1002.10	7.79
サービスが共有空間ネットワークにチェックインするのに要した時間	3594.60	27.95
連携先の PoolGW にシナリオを送るのに要した時間	3193.50	24.83
連携先でシナリオ解析に要した時間	19.40	0.15
連携先でサービス解決エンジンがどのサービスかを判断するのに要した時間	529.15	4.12
連携先のサービスが共有空間ネットワークにチェックインするのに要した時間	1116.90	8.69
VNC サービスの実行までに要した時間	3045.15	23.68
<hr/>		
本項のシナリオに書かれたすべてのサービスが連携するまでに要した時間 (遠隔地間での会議で VNC サービスを利用したプレゼンテーションを行う例)	12636.95	

ある。例えば、旅行を計画する際、飛行機やホテルなどの予約等をする。これらは独立に提供されたサービスであるため、利用者はそれぞれのサービスを受ける場合、それぞれのサイトを順に訪れてサービスを利用しなくてはならない。この研究では、それぞれの単一サービスとを仲介する役割を果たし、複数の単一サービスを組み合わせると一つの利用手順を生成し、利用者に提供する手法を提案している。この研究では単にインターネット上の Web サービスの合成する手法であり、我々の研究のようなさまざまな機器などのサービスがプライベートネットワーク空間に存在するような環境では利用することができない。

・グループウェア

遠隔地間で協調作業を支援するソフトウェアとして、グループウェアがある。グループウェアの持つ機能として、テレビ会議、メンバーの予定を共有するためのスケジューラ機能、アイデアや議事録等を共有する文書共有機能などがある。グループウェアには二種類に大別される。ひとつは、グループ間で共有するサーバを設置し、共有したいサービスやファイルを、Web ブラウザをクライアントとして使用できる Web ベース型がある。もう一方は、各ノード上で動作するソフトウェアが通信しあって情報を取得・更新する P2P 型がある。Web ベース型のものはクライアント側に特別なソフトウェアを導入せずとも、Web ブラウザさえあれば利用できることが便利である。しかし、グループウェアでは、アプリケーションとしての完成度が高いが、サービス機器が固定化されていて、柔軟にサービスを組み替えることが難しい。

・ネットワーク管理技術

複数のコンピュータシステムを管理するフレームワークとして、Func (Fedora Unified Network Controller)⁸⁾ や Hinemos⁹⁾ などがある。これらのアプリケーションは Linux 上で動作するシステム管理フレームワークで、リモートシステム上のタスクを実行し、結果を取得して処理することができる。また、API を提供しているため、リモートシステム上のタスクを柔軟に操作することが可能である。本論文で提案するシステムも複数のスマート環境上で、それぞれのネットワーク上に遍在するノードに接続性を与えるタスクを実行する。また、各サービス間の連携をさせるタスクを実行すると、結果を処理することから類似している。しかし、これらの技術ではさまざまなサーバ管理しているすべてのホストが固定されたものであり、非常に密に結合しており、堅牢なネットワーク管理に利用される。そのためスマート環境のような環境では複数のサービスが利用者に応じて提供したいために柔軟にネットワーク構成されていることから本環境では利用することができない。

8. まとめと今後

本稿では、分散スマート環境間連携のための要件を示し、利用する状況に応じたサービス連携を実現するための分散スマート環境間連携機構の提案を行った。本稿が提案する分散スマート環境間連携機構では、XML 形式のシナリオに利用したい状況を記述しておくだけで、自動的かつ即興的に実現したい環境を構築する。また、シナリオ記述時の利用者の負担を考慮し、抽象的な記述によるサービス指定を可能にした。

今後の課題としては、まず、シナリオの記述用のインタフェースの設計と実装が挙げられる。シナリオには、利用したい状況・環境を複雑に記述することも可能であり、利用者にとって煩雑な作業にならないように容易にシナリオを記述することが望まれる。また、シナリオ記述の際には、現在は誰が記述しても記述したサービスはすべて利用できる。しかし、記述する利用者によって記述できる (または記述したとしても実際に動作する) サービスの認証をする必要があると考える。また、本稿では抽象的なサービス記述でのサービス利用が可能であると述べたが、今回の実装では想定するサービスのバリエーションが乏しいため、今後分散スマート環境間でのサービス連携の様々な可能性を考慮し、設計・実装を改めて試みる必要がある。

謝辞 本研究の一部は、総務省「戦略的情報通信研究開発推進制度 (SCOPE)」の支援を受けて行われた。

参 考 文 献

- 1) Mort Bay Consulting: jetty6 - Jetty Web-Server, <http://www.mortbay.org/jetty-6/>.
 - 2) 中野悦史, 西尾信彦“ Peer Pool: DNS クエリによって構成されるオーバーレイネットワーク ”IC2006 (2006)
 - 3) <http://openvpn.net/>
 - 4) Fielding, R. T.: Architectural styles and the design of network-based software architectures, PhD Thesis, University of California, Irvine (2000).
 - 5) 岩崎陽平, 榎堀優, 藤原茂雄, 田中宏一, 西尾信彦, 河口信夫“ REST に基づく異種スマート環境間のセキュアな連携基盤 ” マルチメディア, 分散, 協調とモバイル (DICOMO2008) シンポジウム論文集 (2008) .
 - 6) Hadley, M. J.: Web Application Description Language (WADL), Sun Microsystems Laboratories Technical Report SMLI TR-2006-153 (2006).
 - 7) 中野篤, 木綿一博, 打橋知孝“ XML 形式のシナリオを用いたサービス合成手法の提案 ”IEICE technical report. Information networks Vol.100, No.536ii(20010206) pp. 29-36 IN2000-180
 - 8) <https://fedorahosted.org/func/>
 - 9) <http://www.hinemos.info/hinemos/>
-