

空間OS

2014年9月29日

先端IT活用推進コンソーシアム
ビジネスAR研究部会
日本総合システム株式会社 中川雅三

- 空間OSがめざすこと
- 特徴
- アーキテクチャ
- 実装
- ロードマップ

空間OSが目指すこと

- ある場所(空間)にあるデバイスと人を**総連携**させるしくみ。
 - AITC BizAR部会で開発中
- 現状:
- 要件の検討
 - 実装実験(ほんのすこしずつ)
 - まずは四畳半くらいの空間から考える
- 目標:
- 実証プロトタイプ
 - できれば本物へ

- 異なる分野・メーカーの連携
- 人と機械の連携
- 現在と未来の連携

- 現状のIT
 - 提供者ごとにサービスが縦割り
 - 例：スマートホーム、スマート家電
 - サービス事業者の乗り換えや、自由な組みあわせができない
 - 例：ヘルスケア
 - 行政や個人、異なる分野の企業が参加しにくそう
 - 例：新発明
 - 良い技術が普及するとは限らない
 - 自社製品にしか使えないデバイスが消えてゆく
- 総連携で
 - 持ち主が自分の持ち物を自由に活用できるようにする。
 - 新しいビジネスが生まれる機会をつくる。
 - よりいっそうのIT活用・省資源・省エネを実現してゆく。

- 現状のIT
 - サービス対象は「ユーザー本人」だけ
 - しかも「成人の健常者」「小学生」といった限られたセグメント対象。
 - 「例外」は切り捨て
 - 機械に処理できないことは避ける。
 - 汎用的なセキュリティのために利便性を犠牲に。
 - 総連携で
 - ITデバイスが人と連携したり、人の連携を助けたりする
 - たとえば高齢者の生活では
家族、ご近所、宅配業者、かかりつけ医師、行政サービス、ボランティア.....
- 提供企業がすべてを作るのではなく、現場の人が工夫できる。
→機械にできないことを人がサポートして、より安全・安心に

- 現状のIT
 - バージョンアップ、製造中止、事業撤退で
 - デバイスが使えなくなる
 - 情報が失われる
 - ノウハウが失われる
 - 今のITは、「一生」「代々」は使えないリスクを抱えている
- 総連携で
 - 情報資産の断絶を防ぐ
 - 情報を蓄積しながら活用する
 - ITデバイスの「寿命」を伸ばす

さきほどのプレゼン

空間OSがめざすこと

- 見知らぬ機械同士が連携する
 - ロボットとルンバや屋内センサ群 – メーカーや世代を超えて
- 人の助けをかりる
 - 離れて住む息子 – 機械が人間と連携する
- 秘密や安全を守る
 - 機械や人の連携の前提 – 陽にはみえないけど
- 一生使い続ける
 - デバイスやサービスを入れ替えながら使い続ける

総連携によるITのインフラ化

- 新産業分野の創出
- 地域社会の活性化
- 持続可能社会の実現

– こちらの記事もご参照ください

<http://aitc.jp/bizar/theTheoryOfARIndustry/100years.html>

空間OSの特徴

こういうものをつくりたい

- 既存デバイス・OSの上に構築する
- ローカルを主体にする
- 空間で認証する
- すべての空間を満たす
- 現場の事情にあわせる
- 永続性を持たせる

- 既存のサービスを動かしつつ、空間OSの機能をプラスアルファする。
 - 「ユビキタス専用OS開発」の発想は必要ない。
 - ゼロから作り直しは実現不可能だろう。
 - 空間OSはサービスを連携させる新しいレイヤ。
 - 空間OSはいろいろなデバイスへインストールする。
 - 「現在と未来の総連携」のためには、あらゆる未知のデバイスとの接続が必要だろう。

- 「クラウド」に対するローカルの優位性
 - 「持ち主」がものごとを制御できる（特にプライバシー）
 - クラウド特有のリスクを回避できる
 - ベストエフォートなサービスのパフォーマンス低下
 - 災害・紛争による広域ダウン（ここは無事なのに...）
 - ビジネスの優先（仕様変更、サービス撤退など）
 - リアルタイム性を実現しやすい
 - ネットワークのトラフィックを減らせる
 - IoTのビッグデータを現場で1次処理
- クラウドも必要
 - バックアップやビッグデータ分析などはクラウド

- 柔軟な認証で活用範囲を広げる
 - 空間にひもづいた認証で認可のハードルを低く
 - 「そこにいる人が持っている携帯」を信頼する。
 - 「その場所の持ち主が信頼している人」を信頼する。
 - 「この場所」を信頼して情報提供する。
 - 匿名で人をつなぐ
 - この建物に居る人へ避難指示を出す。
 - 管理者へ修理を依頼する。

- 対等なつながりで
 - 部屋の空間OSは他の部屋の空間OSとつながる
 - 会議にリモート参加している人の部屋と、会議室のデバイスを相互運用する。
 - サイバー空間内の空間OSとつながってもよい
 - 会社、同好会、離れて住む家族...
 - 「空間」は論理的なものでもかまわない。
- 階層的なつながりで
 - 部屋の空間OSは、建物の空間OSの構成員である
 - 建物の空間OSは、地域の空間OSの構成員である
 - 緊急地震速報を地域の空間OSから建物の空間OSへ通知する。
 - 許可された資源を公共利用に供出する

- 様々な空間固有の事情に対応する。
 - 個人の家、会議室 ...
 - 独り暮らし、3世代家族 ...
 - 日本、ケニア ...
- 未来になっても使える
 - 現在のデータを未来のマシンが使う。
 - 現在のマシンが未来にデータを使う。

すべてを見通すような設計は不可能

→ 柔軟なデータ構造で多様な現場に合わせてられるように

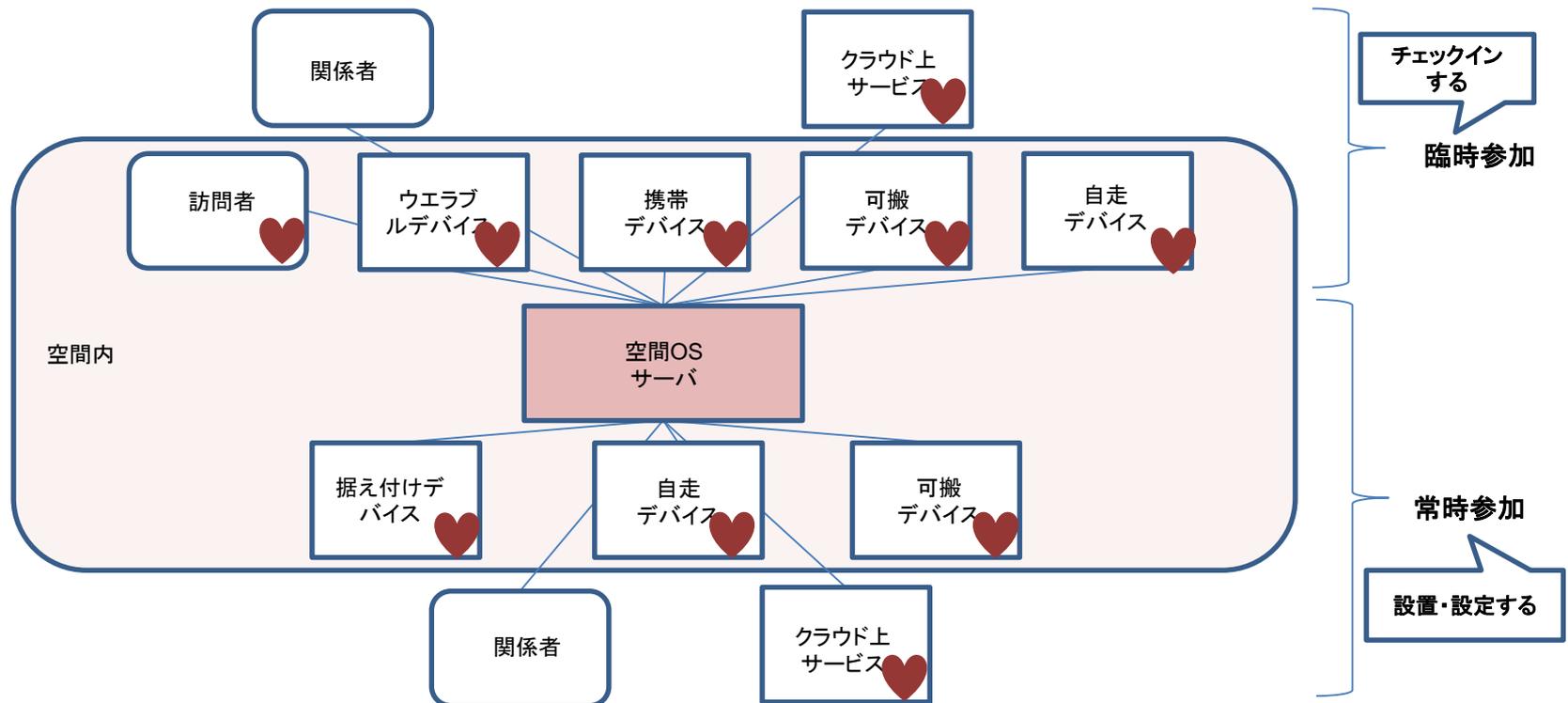
- 社会のインフラとして：
 - メーカー・業者・製品の寿命を超えて、サービスを提供し続ける。
 - 人の加齢・世代交代や世の中の変化へ適応しつづける。
 - 災害や紛争で広域サービスが停止しても、ローカルなサービスは動き続ける。
 - 地域や社会に貢献する。
 - モノやエネルギーの消費を最適化する。
 - ITによって得られる知見を有効活用する。
 - 人の一生を適切に記録する。
 - 一生の健康・病歴情報を医師が閲覧できようしておく。
 - 家族のアルバムを代々うけつぐ。
 - 人が一生を終えたら、指示された情報を消去する。
 - 「空間OS」の存在を前提に政策や制度をつくる。

空間OSのアーキテクチャ

デバイスと「人」で分散OS

空間OSのアーキテクチャ

- 「人」もシステムの構成要素
 - 人は周辺のデバイスを経由してシステムとつながる
- 一部の人とデバイスは離合集散(臨時参加)



♥ = 空間OSエージェント

- デバイス上のアプリケーションとして実装
 - デバイスの資源を空間OSへ提供する。
 - 空間OSの資源をアプリケーションへ提供する。
 - ホストOSに合わせて実装する。
 - いわば「OS Abstraction Layer」 的な存在

- 空間OS構成デバイスをつなぐハブの役割
 - 認証
 - サービス仲介・中継
 - コンテキストストア
 - 人間API

- その場所にあることを起点として、相互に認証しあったデバイスの認証チェーンを構成
 - 空間OSはデバイスがその場にあることを認証
 - 方法は様々：NFC、WiFi、画像認識、音声認識....
 - その場から離れたことの検出も必要
 - 検出できない場合は、認証の鮮度を判定
 - 画像認識などで人も認証
 - 簡易な手段としては「デバイスの持ち主」として人を認証
 - デバイスと空間OSは相互に認証
 - 参加デバイスにとっても、空間OSがニセモノでないことが重要
 - ここにはデジタル証明書を使える。
 - 複数の認証情報を組み合わせて精度を高める
 - 認証チェーンの要素：アナログ認証情報の束
 - アナログ：センサー情報に基づく認証は確率的
 - デジタル証明書：1個の真偽値

- 構成デバイス間のサービスを仲介
 - デバイスが持つI/O装置のサービス
 - 「携帯電話から部屋にあるプロジェクターを利用」
 - デバイス上のアプリケーションのサービス
 - 「会議室にいる人の携帯電話へスケジュールを登録」
 - クラウド上のサービス
 - 「部屋の管理者へメール送信」

空間OSのアーキテクチャ

- 空間OSアプリケーションへ環境情報(コンテキスト)を提供する。
 - 事実
 - 認証情報、センサー測定値、人々の情報..
 - 記録
 - 過去の事実の記録
 - 処理の途中経過
 - デバイス上に分散した空間OSアプリケーションが連携するためのメッセージ群
 - 「黒板モデル」に似ている。
 - 空間OSアプリケーションがエージェントで、コンテキストが黒板。

- 構成要素「人」へのAPIを標準化
いわゆる「UI」を実装するための部品
 - － 人のコンテキスト
 - 信頼度(システムにとっての)、ここでの社会的ルールetc...
 - － 人への連絡(機械→人)
 - テキストメッセージ伝達
 - 視覚、聴覚、触覚、...、第6感
 - － 人にまつわるセンシング(人→機械)
 - 位置、発話、身振り...

- 既存のアプリケーション+ α でスタート
 - 「+ α 」=空間OS経由でアプリケーション連携
 - 「会話ロボットがルンバへ掃除を指示する」
 - 「エアコンが部屋のブラインドを閉める」
 - 「デバイスが自分の修理依頼を出す」
- 将来は空間OSならではのアプリケーションへ進化
 - 行動履歴から足の老化を検出して運動を促す
 - 「あれ取ってくれ」「ご自分で...」
 - 引っ越し先の空間OSへ設定をひきつぐ
 - 「習慣・ルール: エアコンはブラインドを閉めてよい」
 - 「家族・知人の顔・身体特徴」

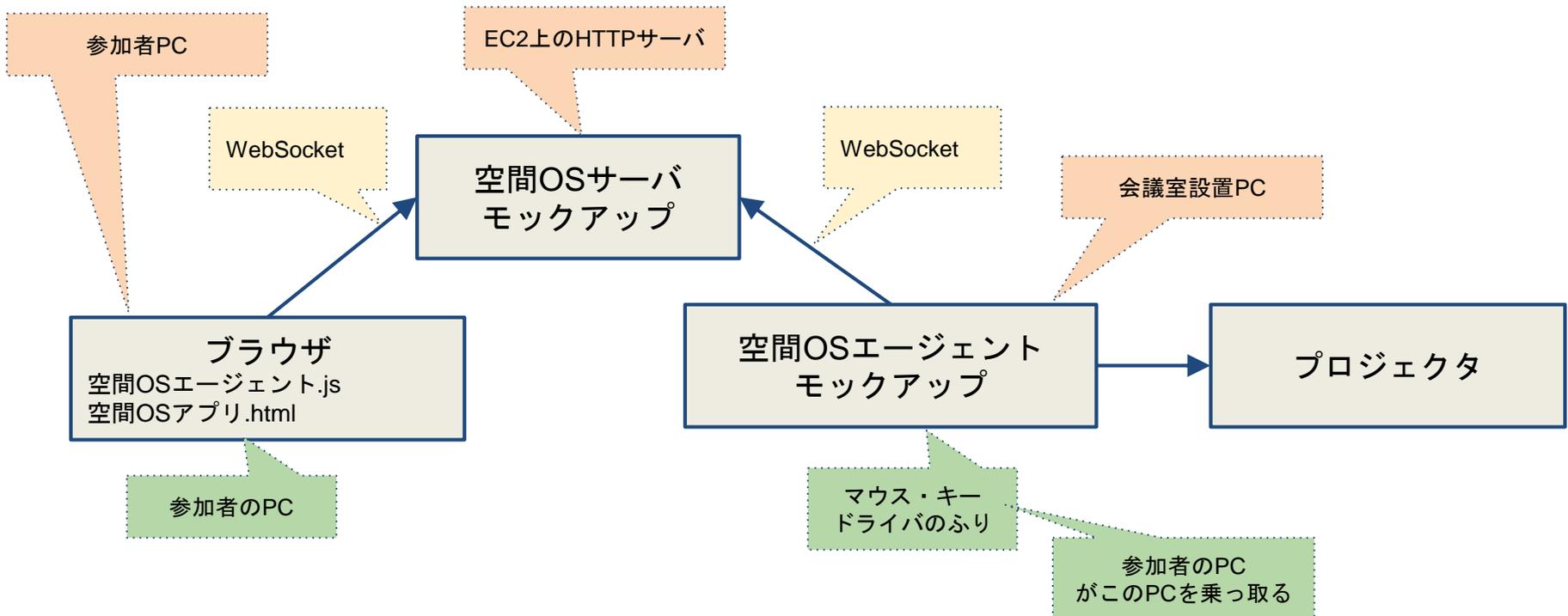
空間OSの実装

- 基礎部分は既存の技術で
 - やろうと思えば作れる。
- 固有の技術開発も必要
 - コンテキストストア
 - セキュリティ
 - 生態系的な分散処理

- 空間OSサーバ
 - 最初は専用サーバとして実装
 - Raspberry Piのようなハードウェア
 - はじめはクラウドもあり
 - 「ローカルを主体に」には反するが、ロードマップの初期には手軽さも重要。
 - ゆくゆくは空間OSエージェントの一種とする可能性
 - 「サーバ専用デバイス」を不要に。

- 最初は素朴な実装でもいい
 - 空間認証
 - 来たことを手入力
 - 壁に貼ってあるQRコードをスキャン
 - NFCやbluetooth low energyで
 -
- 空間OSサーバとデバイス間の通信はHTTP
 - TCP/IPの通らない通信路では、同等のステートレスな通信方式で。
 - アプリケーションと空間OSエージェント間の通信は、intentなどの各OS固有の機能を使う。
 - 異なるデバイスのアプリケーション間の通信は、空間OSが中継。
- 1個の機能に複数の実装方式があってもよい
 - 例：他デバイスの出力装置との通信
 - 空間OSがアプリケーションデータをデバイスへ中継
 - 空間OSがアクセスポイント情報と認証トークンを渡して、アプリケーションがデバイスへ直接アクセス

- 実装実験: プロジェクトにつながったPCを共有
 - プロジェクト接続PCのキーボード・マウスを他のPCからリモート操作
 - 参加者PCのブラウザで空間OSサーバの画面を操作する



- 実装実験：プロジェクトにつながったPCを共有
 - 空間OSサーバをクラウドに置く実装は、リアルタイム性への要求がそれほど厳しくなければOKかも。
 - HTTPはRESTな使い方以外もOK。
 - WebSocketつかえる
 - アプリ記述用に重要になりそうなJavaScriptと親和する。
 - やってみて気づくこともある。
 - この実験ではLinuxのキーボードデバイスを置き換えた。英数字は送れても、漢字を送るのは難しそうだ。

- 最下層はRDFストア+ α で実装
 - RDFで任意のデータを表現
 - SPARQLで問い合わせ
 - + α
 - データごとの権限チェックによるアクセス制御
 - 情報鮮度管理による、「現在の事実」と「記録」の区別
 - 信頼度の扱い
- RDFストアへのラッパーで実装できそう
- 課題は語彙・オントロジーの整備
- メーカー主導では作れないだろう → コミュニティAPI

空間OSの実装

RDFデータの例

```

@prefix :      <http://bizar.aitc.jp/ns/spaceos/0.1/> .
@prefix ha:    <http://bizar.aitc.jp/ns/spaceos/0.1/#人間API> .

# My first RDF
_:r01
  :場所      :ここ      ;
  :在室      _:p01      ;
  :在室      _:p02      ;
  .

_:c01
  :主人      :p01      ;
  :電話番号  "012345678" ;
  :何        :携帯電話 ;
  ha:テキストメッセージ "https://192.168.0.2/51354/ha/message/send" ;
  ha:緊急連絡 "https://192.168.0.2/51354/ha/message/emergency" ;
  .

_:c02
  :主人      _:p02      ;
  :何        :PC        ;
  ha:テキストメッセージ "https://192.168.0.2/51221/ha/message/send" ;
  ha:緊急連絡 "https://192.168.0.2/51221/ha/message/emergency" ;
  .

_:p01
  :最終認識  "2014-07-11T10:45:23.5+09:00" ;
  :人物      _:np01      ;
  .

_:p02
  :最終認識  "2014-07-11T10:55:20.2+09:00" ;
  .

_:np01
  :通称      "中川雅三" ;
  .
  
```

この部屋には2人いる

1人は携帯電話を持っている

1人はPCを持っている

1人は身元がわかっている

1人は誰かわからないが、10時45分にはここにいた。

身元がわかっている人は中川さんだ

主語を無名ノードにするとアクセス制御しやすいかも

空間OSの実装

SPARQLクエリの例

```

PREFIX : <http://bizar.aitc.jp/ns/spaceos/0.1/>
PREFIX ha: <http://bizar.aitc.jp/ns/spaceos/0.1/人間API>
SELECT ?url
WHERE {
  ?room :場所 :ここ .
  ?room :在室 ?p .
  ?c :主人 ?p .
  ?c ha:テキストメッセージ ?url .
}
LIMIT 10

```

クエリー
 ここにいる人の所持しているデバイスへメッセージを送るためのアクセスポイントは？

結果
 URLは空間OSが提供する人間APIアクセスポイント
 ↓
 連絡先を知らずに連絡がとれる

```

-----
| url |
=====
| "https://192.168.0.2/51354/ha/message/send" |
| "https://192.168.0.2/51221/ha/message/send" |
-----

```

- 「鮮度情報つきアナログ値の束」の認証チェーンとは...
 - どう線引きするか
 - 工学の問題ではない？
- 障害やマルウェアもロバスト化する？
 - 「潜伏期間数十年」「もぐらたたき」を防げるか
 - 空間OSに「電源スイッチ」や「リセットスイッチ」は実装できるか
 - 「電源スイッチ＝単一障害点」として空間OSサーバをエージェント化せずに残すとか
- 数十年分のゴミ情報をためないためには？

- 新しいプログラミング技術を作り出す必要がある。
 - 大きく異なるタイプのデバイス・OSを多数連携させる。
 - グリッドやクラスタと違う。
 - ホストOSの制約で実装できない機能があることもある。
 - 「HTTPメッセージとIntentとExtensibilityをどうマップするか」とか。
 - 全く異なる種類のプロセスを多数並行して実行する。
 - 「プロセス」とは何かを明確に定義できないかもしれない。
 - デバイスが勝手に抜けたり再参加したりする。
 - 「どこかへ行っていた人の携帯でプロセスが動き続ける」など
 - 一瞬で終わるプロセスから数十年にわたるプロセスまでである。
 - 故障したデバイスを入れ替えても実行がつづく。
 - プロセスの「終わり」って何？
 - 現実の時空間を通してプロセスが影響しあう。
 - 「ロボットがそこに立ってたら、ルンバが掃除できない。」
 - 「応答が遅いと別の意味になってしまう..」
 - 「このユーザは気難しい人だ」

空間OSのロードマップ

まだロードマップは描けないが...

- たぶんなかなか儲からない: 空間OSは実現していない
 - 価値はある。
- 機は熟している
 - デバイスの能力が十分高くなった。
 - 個人や小チームによるボトムアップなものづくりが盛り上がってきた。
- 開発者としてAITCは適しているのでは？
 - 実現する技術がある
 - 水平連携の組織
 - 非営利

実装家募集中；

最新情報は
こちらをご参照ください

 <http://aitc.jp>

 <https://www.facebook.com/aitc.jp>



ハルミン

AITC非公式イメージキャラクター