ウィンターワークショップ2010 in 倉敷　　～サービス指向～

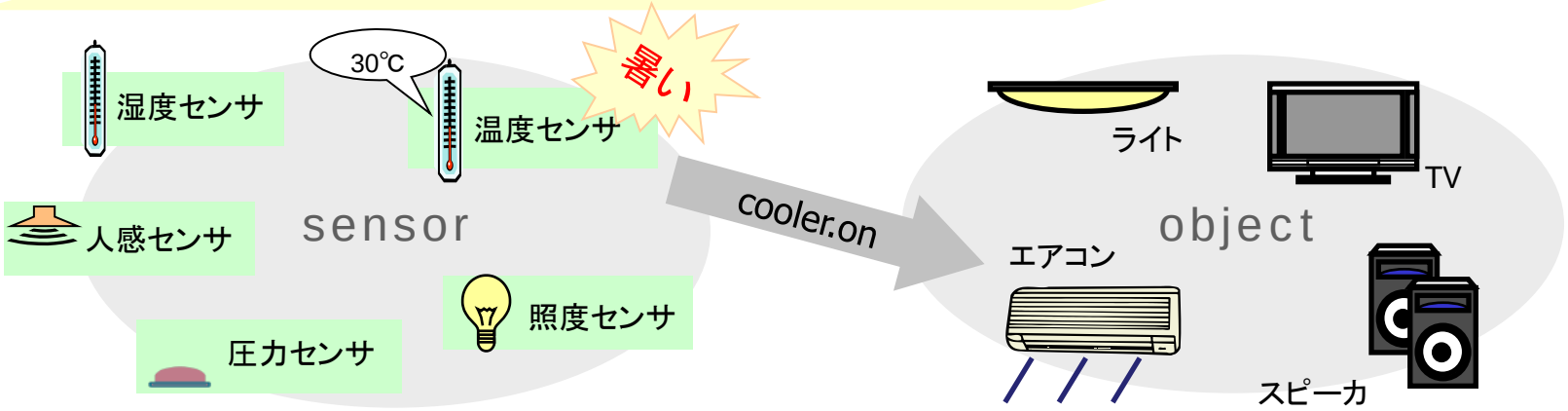
SMuP:センササービスのマッシュアップを実現するサービス指向基盤

神戸大学大学院　工学研究科
坂本寛幸、井垣宏、中村匡秀

コンテキストアウェアアプリケーション

コンテキストとは

コンピュータやネットワーク上の情報，または，
現実の世界から発生する情報から捉えられる「**状況**」情報



コンテキストを活用したアプリケーション　＝　コンテキストアウェアアプリケーション

既存のコンテキストアプリケーションの問題点

センサとアプリケーションが
密に結合！

- アプリケーションが複雑に
- 柔軟なカスタマイズができない

センサへのポーリングによる
通信量の増加

- 利用するセンサ数が増えれば増えるほど
  通信量は増大してしまう.

* 坂本寛幸, 井垣宏, 中村匡秀, “コンテキストアウェアアプリケーションの開発を容易化するセンササービス基盤,” 電子情報通信学会技術研究報告, vol.108, no.458, pp.381-386, March 2009.

[先行研究] センササービス基盤*

Single Sensor Service

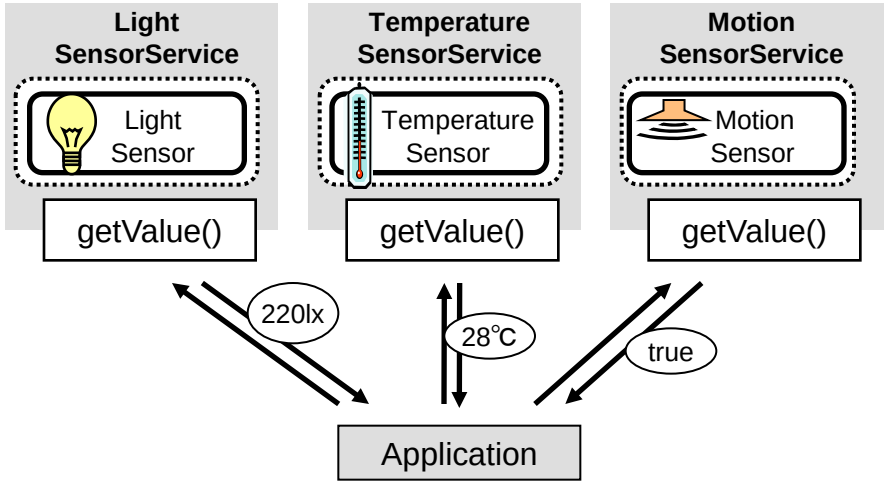
- **getValue**
- **getResisterdContextValue**
- **register**
- **subscribe**

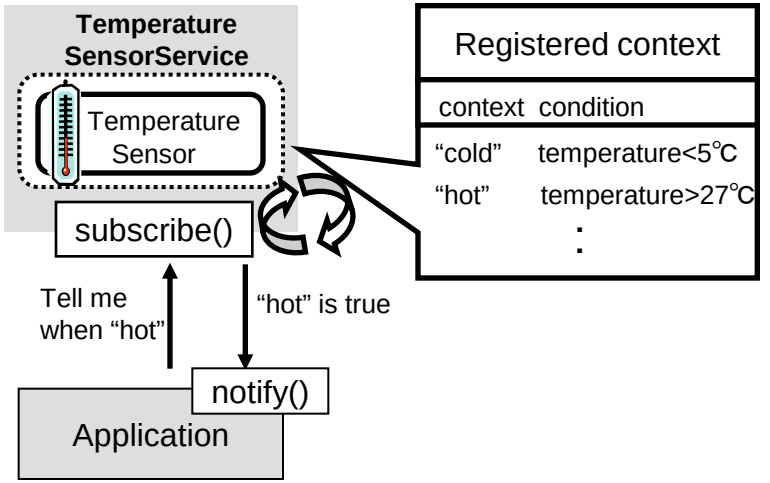
- 温度センサ，照度センサなどのセンサデバイスをWebサービスとして公開.（SOAP,RESTで利用可能）
- サービス化されたセンサ（**センササービス**）は標準的なインタフェース（API）を公開.
  ⇒ アプリケーション-センサ間の疎結合化を実現
- コンテキスト推定のための条件判定をセンサ側に委譲することで、Publish-subscribe型のメッセージ交換パターンを実現.
  ⇒ センサ-アプリケーション間の通信量を削減

[getValueインタフェース]
センサのプロパティ値を取得可能

[registerインタフェース]
コンテキスト推定のための条件式（**コンテキスト条件**）を登録

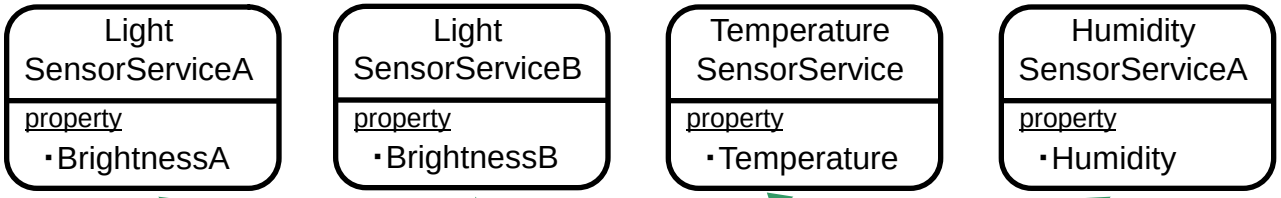
[subscribeインタフェース]
コンテキスト条件が満たされた時にアプリケーションに通知する.





複雑なコンテキスト推定のための課題

センササービスだけでは，自身のプロパティを利用した単純なコンテキストの推定しかできない



○ 「暑い」 : Temperature > 27℃

× 「明るい」: 部屋の平均の照度 (BrightnessA+BrightnessB)/2 > 200lx

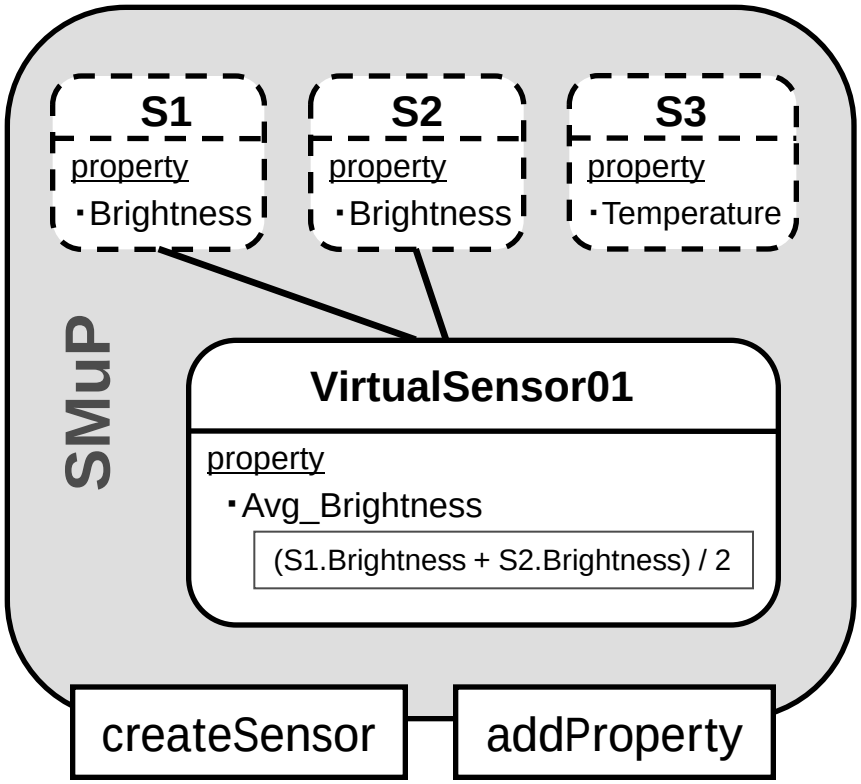
× 「蒸し暑い」 : Temperature > 27℃ && Humidity>80%

開発者がアプリケーション内で複雑なロジックを実装する必要がある

複数センサを利用したコンテキストの推定ができるプラットフォーム
SMuP (Sensor Mashup Platform) を提案.

SMuP(Sensor Mashup Platform) (1/2)

既存のセンササービスをマッシュアップすることで仮想センササービスを作成



Step1 仮想センササービスを作成する.

createSensorインタフェースを利用

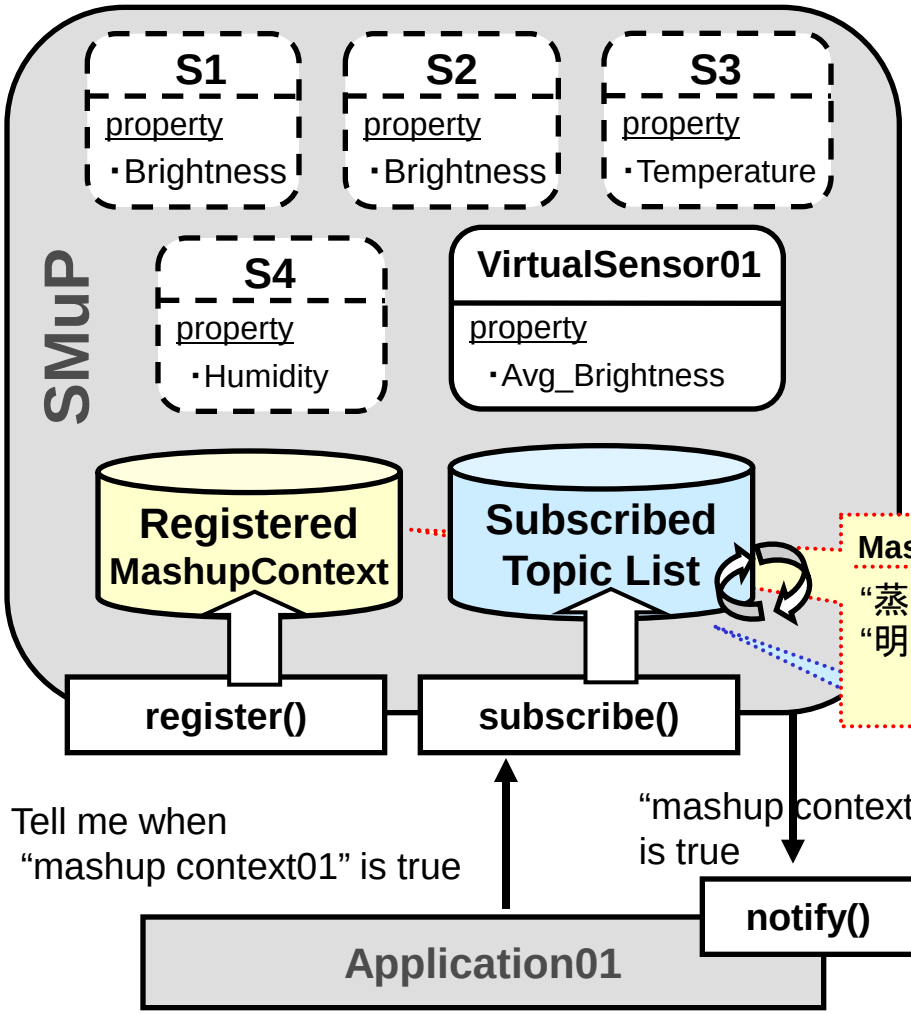
Step2 登録されているセンササービスの
持つプロパティをマッシュアップし,
仮想センサのプロパティを作成する.

平均の照度（Avg_Brightness）:
(S1.Brightness + S2.Brightness) / 2

Step3 マッシュアッププロパティをStep1 で
作成した仮想センササービスに
登録する.

addPropertyインタフェースを利用

SMuP(Sensor Mashup Platform) (2/2)



コンテキストの登録

複数のセンサが持つプロパティを横断的に扱うために，SMuP 自身が *register* および *subscribe*インタフェースを公開

[register]
マッシュアップコンテキスト
条件を登録

[subscribe]
マッシュアップコンテキスト
条件が満たされた時に
アプリケーションに通知する.

MashupContextName	MashupContextCondition
“蒸し暑い”	S3.Temperature>28 && S4.Humidity >70
“明るい”	VirtualSensor01.Avg_Brightness < 200
:	:

topic	subscriber
mashup context01	Application01

センササービスとコンテキストの品質特性

精度

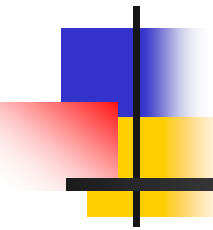
個別のセンサの精度として考えられる分解能や範囲の組み合わせによる仮想センサやコンテキストの精度

即時性

利用するセンサの数や階層が深くなるにつれ，現在の値とセンサによって取得された値のタイミングにずれが発生する可能性があるため，コンテキストやセンサに応じた即時性に関する判断が必要となる．

効率性

サービスとしての資源効率性や時間効率性，すなわちターンアラウンドタイム，スループットやCPU 使用率，ネットワーク負荷といった効率に関するメトリクスはマッシュアップにおける重要な判断基準となる可能性がある．



ありがとうございました。

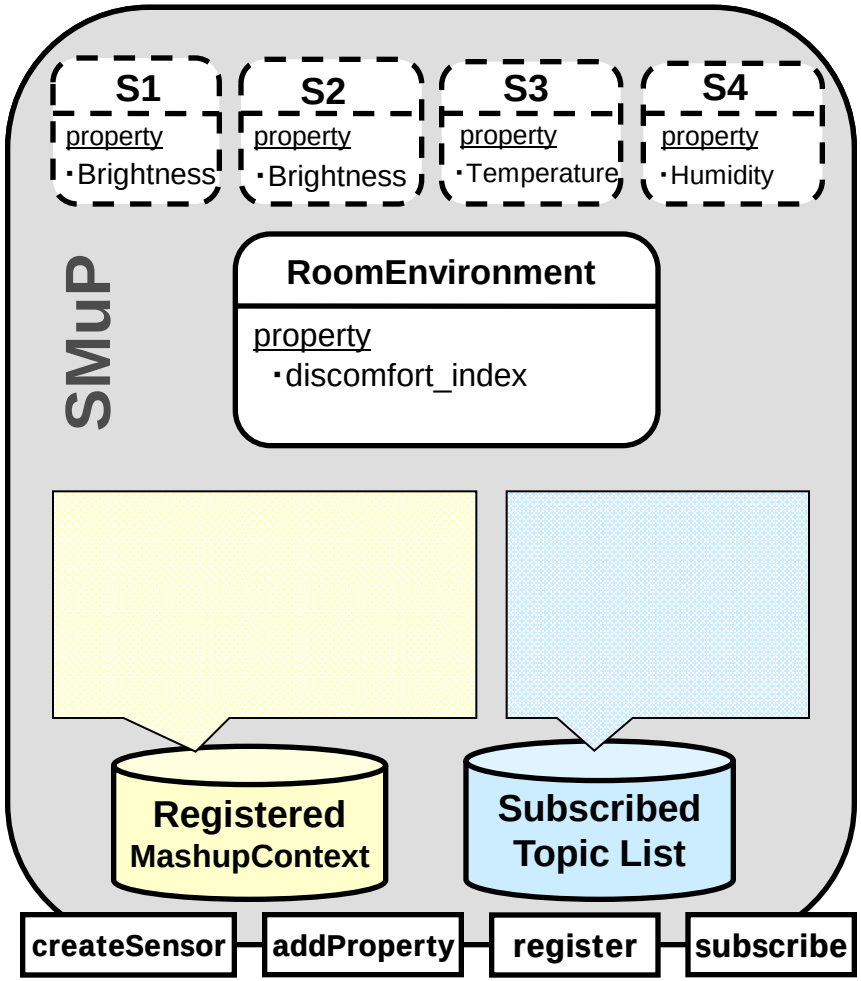
SMuPを利用したアプリケーション開発

- センサに関するロジックをSMuPに任せる
- アプリケーション内ではコンテキストが推定された際の振る舞いに関するロジックのみを実装すればよい

SMuPを利用することでコンテキストアウェアアプリケーションの開発が容易に

- ケーススタディ
  - 不快指数を利用した冷房サービス
  - エコ照明サービス

ケーススタディ
不快指数を利用した冷房サービス(1/3)



「不快指数」・・・
温度と湿度から計算される蒸し暑さの指標

不快指数(T:温度　H:湿度)

$$0.81T + 0.01H(0.99T - 14.3) + 46.3$$

不快指数が
75～80: 不快　⇒冷房(28度)
＞80:とても不快　⇒冷房(26度)

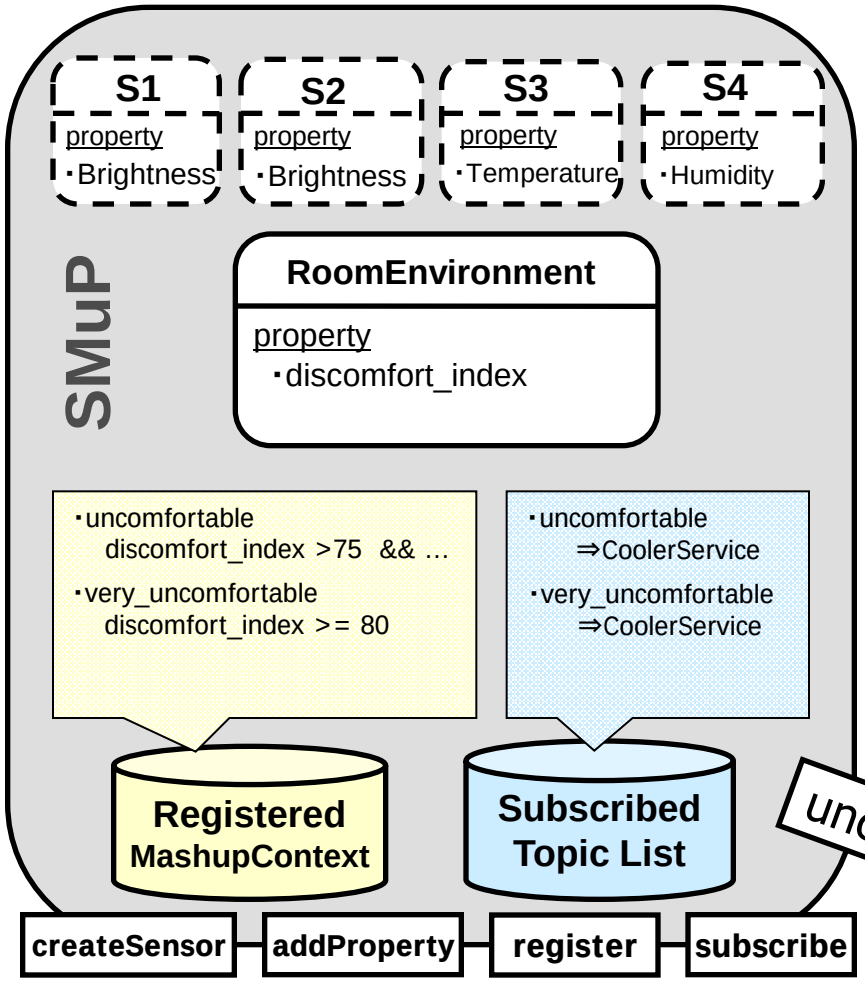
RoomEnvironmentSensorの作成

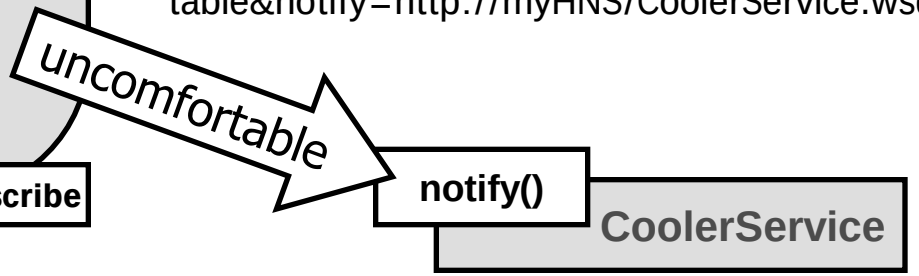
http://myHNS/SMuP/createSensor?name=RoomEnvironment

マッシュアッププロパティ(discomfort index)の登録

http://myHNS/SMuP/addProperty?sensor=RoomEnvironment&property=discomfort_index&expression=0.81*s3.Temperature+0.01*s4.Humidity*(0.99*s3.Temperature-14.3)+46.3

ケーススタディ
不快指数を利用した冷房サービス(2/3)





マッシュアップコンテキスト:「不快」の登録

http://myHNS/SMuP/register?context=uncomfortable&condition=RoomEnvironment.discomfort_index>75&&RoomEnvironment.discomfort_index<80

マッシュアップコンテキスト:「とても不快」の登録

http://myHNS/SMuP/regster?context=very_uncomfortable&condition=RoomEnvironment.discomfort_index>=80

マッシュアップコンテキスト:「不快」の購読

http://myHNS/SMuP/subscribe?context=uncomfortable&notify=http://myHNS/CoolerService.wsdl

マッシュアップコンテキスト:「とても不快」の購読

http://myHNS/SMuP/subscribe?context=very_uncomfortable&notify=http://myHNS/CoolerService.wsdl

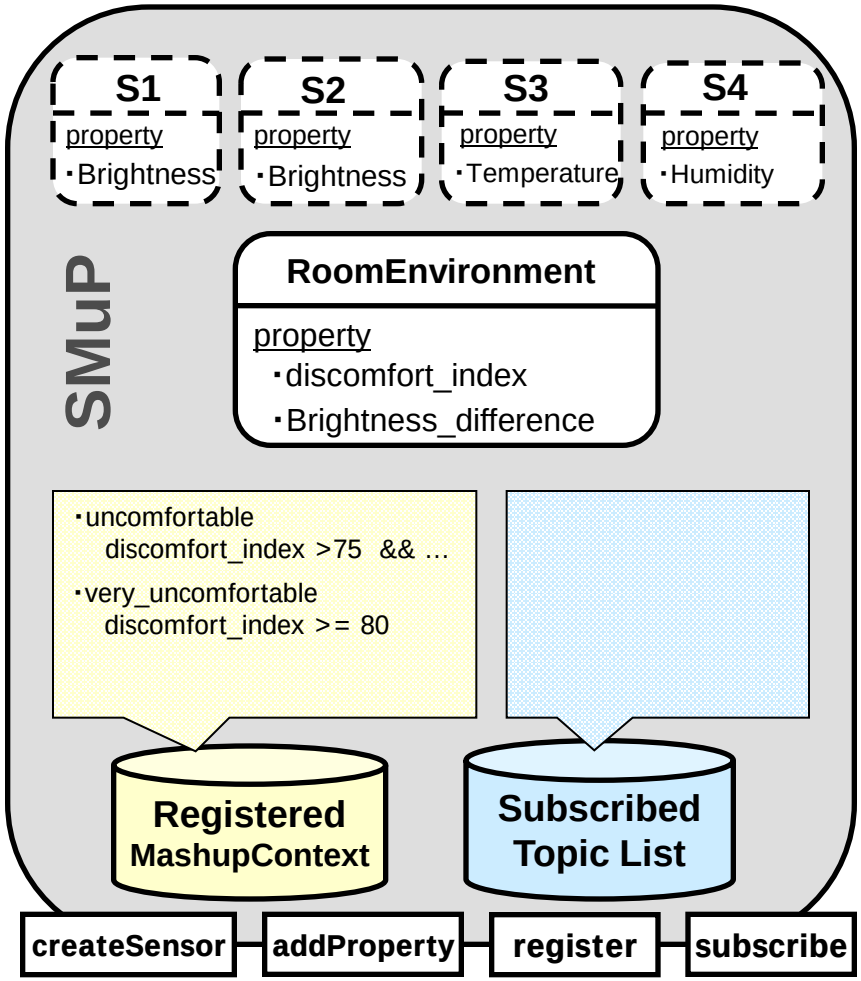
ケーススタディ
不快指数を利用した冷房サービス(3/3)

CoolerService のプログラム例

notifyメソッド内に各コンテキストが推定された際の振る舞いを記述

```
public class CoolerService{
    public boolean notify(String context){
        if(context.equals(“uncomfortable”)){
            cooler.on(preset=28)
        }else if(context.equals(“very_uncomfortable”)){
            cooler.on(preset=26)
        }
    }
}
```

ケーススタディ
エコ照明サービス(1/2)



部屋の内外の照度差（S1:室外　S2:室内）
S1.Brightness – S2.Brightness

部屋の内外の照度差が
>=100lx :部屋の外が十分明るい　⇒カーテンを開ける
<100lx :部屋の外が暗い　⇒　照明をつける

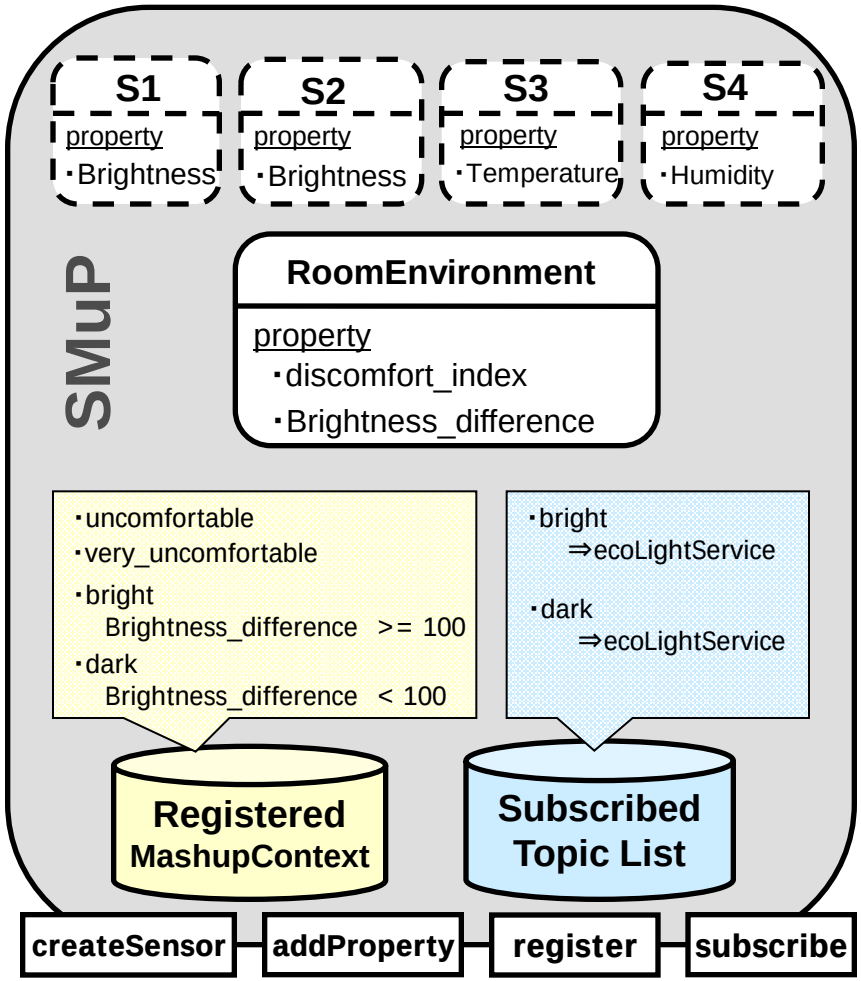
新しい仮想センサの作成

RoomEnvironment センサを利用

マッシュアッププロパティ(Brightness_difference)の登録

http://myHNS/SMuP/addProperty?sensor=RoomEnvironment&property=Brightness_difference&expression=S1.Brightness-S2.Brightness

ケーススタディ
エコ照明サービス(2/2)



マッシュアップコンテキスト:「室外が十分明るい」の登録

http://myHNS/SMuP/register?context=bright&condition=RoomEnvironment.Brightness_difference>=100

マッシュアップコンテキスト:「室外が暗い」の登録

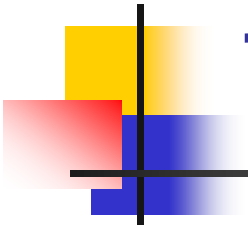
http://myHNS/SMuP/register?context=dark&condition=RoomEnvironment.Brightness_difference<100

マッシュアップコンテキスト:「室外が十分明るい」の購読

http://myHNS/SMuP/subscribe?context=bright&notify=http://myHNS/ecoLightService.wsdl

マッシュアップコンテキスト:「室外が暗い」の購読

http://myHNS/SMuP/subscribe?context=dark&notify=http://myHNS/ecoLightService.wsdl



まとめ

- 複数センサを利用したコンテキストの推定ができるプラットフォームSMuPを提案.
  - 仮想センササービスの作成
  - マッシュアップコンテキストの登録
- ケーススタディによりSMuPの有効性を確認した.
  - センサに関するロジックはすべてSMuPに任せる
  - アプリケーション開発者はコンテキスト推定結果にもとづく振る舞いに関するロジックのみ実装.
- 今後の課題
  - SMuPの実装
  - SMuP利用のためのユーザインタフェース