自律協調型サービス指向アーキテクチャの提案

M2006MM002 藤山 麻衣

指導教員 青山 幹雄

1. はじめに

ホームネットワークシステム(HNS : Home Network System)では多様なサービスを協調的に提供する必要があり，サービス提供におけるユーザ，空間，機器といったエンティティの相互作用は複雑化している．本稿は相互作用の相違に着目し，サービス提供を対話性と自律性で分離する．それぞれに異なる制御アーキテクチャを適用し，これらを統合して自律協調型サービス指向アーキテクチャ(SOA : Service-Oriented Architecture)を提案する．

2. HNS サービス提供における問題点

2.1. HNS への要求

家庭では様々な人がサービスを利用する．ユーザや周囲の状況に合ったサービスを少ない操作で提供することがユーザビリティの点から重要である．自動的にサービスを提供する方法が研究されているが，ユーザの望まないサービスはかえって利便性を損なう．よって，システムの自律協調的なサービス提供には家庭環境やユーザの状況を獲得し，システムの動作に反映する仕組みが求められる．

2.2. HNS サービス提供の特性

エンティティの相互作用はその多様さから複雑化している(図 1)． 図 1 に示す通り，家庭にはテレビのチャンネル変更のようにユーザが直接結果を受けるサービスと，エアコンの温度調整のように部屋の環境を介して結果を受けるサービスが存在する．環境に作用するサービスは，同じ空間内に複数存在することが想定され，状況によって起動されるサービスは変化する．このようにプロバイダが状況により変化する場合，従来の SOA のようなリクエスタの直接指定による呼出しでは対応できない．

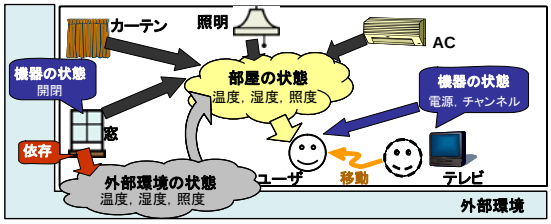

図 1 HNS の相互作用

さらに，サービスや室内の環境に対する外部環境の影響や複数サービスの競合，副作用の可能性も考えられる．

3. 解決のアプローチ

3.1. 前提とする HNS モデル

本稿では HNS を家庭内の機器，ユーザ，空間といったエンティティの状態を中心にモデル化する(図 2)．これにより相互作用の複雑さを軽減できる．本稿ではエンティティの状態をプロパティと呼び，以下の通り 3 種類定義する．

(1) サービスプロパティ：動的に変化するサービスの特性や状態である．各々の機器は自身のサービスが直接制御可能なサービスプロパティを持つ．

(2) 環境プロパティ：ユーザの周囲を取り巻く環境の状態である．複数の機器や外部環境からの影響を受ける．

(3) ユーザプロパティ：ユーザの特性や状態，システムに対する要求を表す．

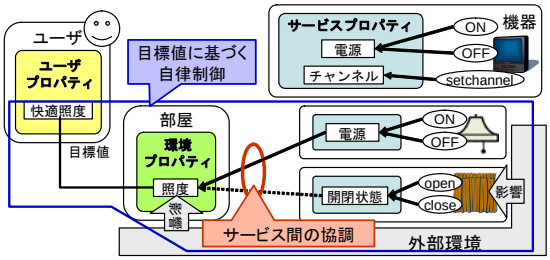

図 2 状態中心の HNS モデル

この HNS モデルでは，環境プロパティを操作するサービスは複数考えられ，これらを協調動作させることでユーザの要求に合致する結果を出す必要がある．また，ユーザプロパティを目標とし，プロパティを目標値に保つために自律制御を行う必要がある．そこで，本稿は自律協調的なサービス提供を実現するためのアーキテクチャを提案する．

3.2. 自律協調型 SOA の提案

3.2.1. 自律協調の定義とその要件

本稿における自律協調の定義は以下の通りである．

(1) 自律：各エンティティが，環境の状態が目標値に収束するように自身を制御する．

(2) 協調：複数エンティティ間で調整を行いながら全体として機能を実現する．

自律性の要件として，各機器によるエンティティ状態の認識，状態に基づくサービス起動の判断，サービス起動が，協調性の要件として，機器間の相互運用性と疎結合な協調モデルが挙げられる．

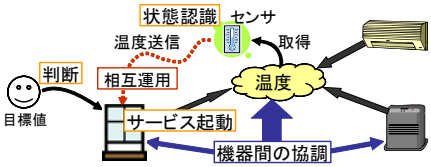

図 3　自律と協調

3.2.2. 自律協調型 SOA の実現

SOA で自律協調を実現するには，アンビエントインテリジェンスの特性[1]を利用する．SOA の特性には，プラットフォーム独立性，標準プロトコルによる相互接続性があり，エンティティ間の相互運用が容易になる．従来はリクエスタがプロバイダを直接指定してサービス起動を行ってきたが，パブリッシュ/サブスクライブ方式のような非同期イベント処理により，プロバイダが状況に応じて変化する HNS においてもエンティティ間の協調が可能である．本稿で利用するアンビエントインテリジェンスの特性は，エンベデッド，コンテキストアウェア，アンティシペイトリである．これによりサービスが環境に組込まれ，状況を認識し，認識された情報に基づきサービス提供を行うことが可能になる．アンビエントインテリジェンス環境下では，事象の変化を捉える必要があり，本稿では各種センサの利用を前提とする．

4. 自律協調型サービス提供モデルの提案

4.1. 相互作用の分析

本稿の HNS モデルにおいて，サービスプロパティはサービスに直接操作され，環境プロパティはサービスプロパティの影響を受ける．ユーザプロパティは自律制御の目標値やサービス決定に利用する．各プロパティの特性とサービス提供における扱い方は異なる．本稿ではサービス提供を性質により分離し，各サービス提供に適切な制御モデルを提案する．本稿では 2 つの観点から分離を行う．

4.1.1. プロパティ制御の観点による分離

サービス提供はユーザの物理的干渉が起点となるものとセンサによるプロパティ検出を起点としたシステムによる自律的なものに分けられる．

(1) 対話型:ユーザとサービスのインタラクティブな相互作用が起こる(図 4)．ユーザのリクエストに対して動的にプロパティを変更する．

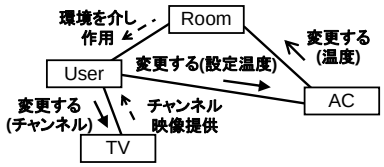

図 4　対話型サービス提供

(2) 自律型:操作されるプロパティの性質は静的で，値を一定に保持しようと環境が自律的にサービス提供を行う(図 5)．

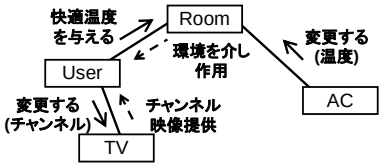

図 5　自律型サービス提供

4.1.2. ユーザとプロバイダの関係による分離

相互作用におけるエンティティ間の関係に着目する．

(1) 直接:プロバイダとユーザは直接相互作用する(図 6)．制御対象がサービスプロパティの場合に起こる．

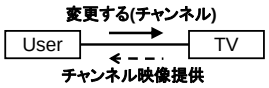

図 6　直接サービス提供

(2) 間接:プロバイダは空間を介してユーザに作用する(図 7)．制御対象が環境プロパティの際に起こる．

図 7　間接サービス提供

4.1.3. サービス具体例

上に述べた 2 つの性質により，サービス提供を 4 つに分類することができる．サービスの具体例を表 1 に示す．

表 1　サービス提供の分類:サービス具体例

		プロパティ制御	
		対話型	自律型
ユーザとプロバイダの関係	直接	リモコンによるテレビチャンネル操作，テレビの電源 ON	ユーザ移動時のテレビチャンネルの値引継ぎ
	間接	リモコンからのエアコンの設定温度変更，ユーザの手による窓の開閉	ユーザプロパティに基づく温度，湿度，照明などの自動調節

4.2. 相互作用モデルの決定

4.1 節で分離したサービス提供の関係と分離された各サービス提供の相互作用モデルを表 2 に示す．

表 2　サービス提供の分類と相互作用モデル

		プロパティ制御	
		対話型	自律型
ユーザとプロバイダの関係	直接	直接要求/応答　(1)	直接イベント駆動 (3)
	間接	間接要求/応答　(2)	間接イベント駆動 (4)

4.2.1. 対話型サービス提供(表 2-(1), 2-(2))

対話型サービスでは，相互作用の起点はユーザである．ユーザの操作に対しては迅速な対応が求められるためサービス提供は同期的に処理されるべきと考える．さらに，ユーザはサービスプロバイダを指定してサービス起動を行い，その結果を待つ．これらのことから要求/応答モデルを適用

可能と考える.

4.2.2. 自律型サービス提供(表 2-(3), (4))

HNS ではセンサにより環境の情報を同期的, ないし非同期的に検出する. 自律型サービス提供は, 検出情報に関連あるサービスを, 特定の状況下で起動するものである. 情報の発生とそれに対応する処理を行うという点からイベント駆動モデルが適切と考える. イベント駆動モデルにより, 同時多発的に発生するデータを生起時刻順に順次処理を行うことで処理効率も向上すると考えられる.

4.3. 各サービス提供の特徴

(1) サービス競合(表 2-(2), (4)):特定のプロパティに作用するサービスが複数存在することを指す. 状況やユーザ要求に基づくサービス選択が必要になる.

(2) 副作用(表 2-(2), (4)):特定の環境プロパティの値を変更するために起動したサービスにより, 異なる環境プロパティの値が変更されることを指す. 例えば, 室温を下げるために窓を開けた際に湿度が変化し, ユーザの快適湿度の条件を満たさなくなる場合が挙げられる.

(3) 外部環境の影響(表 2-(2), (4)):外部環境の変化により室内環境も変化してしまうこと, また環境に作用するサービスの結果が外部環境により変化することを指す.

5. 自律協調型 SOA の提案

5.1. 異なる相互作用モデルの統合

本稿ではユーザ操作が起点となる対話型サービス提供に対し要求/応答モデル, センサによるプロパティ検出が起点となる自律型サービス提供に対しイベント駆動モデルを適用し, これらを統合したアーキテクチャを提案する. 対話型ではプロパティが動的に変化することを想定している. これに対し, 自律型ではプロパティは静的であり, 外部環境や内部環境の変化が起こる状況下で, 値を一定に保つことが求められる. このように, 対話型と自律型では制御の機構が異なっており, 制御機構の相違を感じさせることなく, ユーザに対しサービスを提供する必要がある. 本稿は, 性質の異なるサービス提供を統一的に制御するために自律協調型 SOA を提案する.

5.2. 自律協調型 SOA

5.2.1. 構成要素

システムの構成要素とその関係を図 8 に示す.

User:HNS の利用者
Event:各種プロパティの周期的, 非周期的な検出情報
Sensor:Event を検出すると Notification を送信
Notification:イベントの発生とその状態を示すデータ
Service:関連する Notification を受信すると起動される
Subscription:
各サービスが対応するイベントの種類や周囲の状態を規定
Event Notification Service:
Subscription を元に Notification を関連するサービスに分配
Request Service:
リモコンなどユーザ操作を受けてサービスを起動

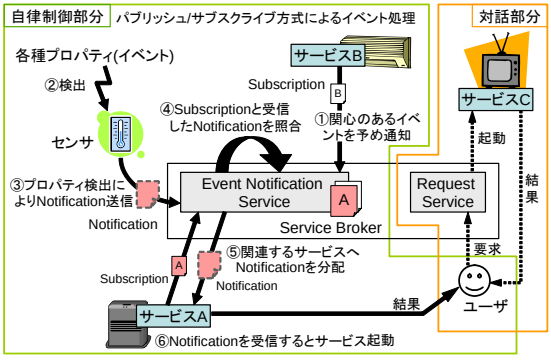

図 8　アーキテクチャ構成図

5.2.2. 対話型処理

リモコン操作などのように User が直接 Service を指定してリクエストを送信する場合は, Request Service がリクエストを受けてサービスを起動する.

5.2.3. 自律型処理

各種プロパティの周期的, ないし非周期的な検出をイベントと捉え, パブリッシュ/サブスクライブ方式でイベント処理を行う. Service は Notification Service(以下 NS)に対して予め Subscription を送信しておく. Sensor が Event を検出する Notification が NS に対し送信され, Subscription と照合される. マッチする Subscription を持つ Service に対してのみ Notification は送信され, Service が起動される.

5.3. 統合アーキテクチャにおける課題と対応策

(1) 課題

1) サービス競合:特定の Notification が複数サービスの Subscription と合致した場合に発生する.
2) 副作用:サービス実行の結果, 目的とは異なるプロパティも変化させることを指す.
3) 副作用の連鎖:副作用により変更されたプロパティを元の値に保持するためサービスが起動され, 異なる副作用を発生させること. Event 発生, Notification 送信, Service 起動がループする可能性がある.
4) 時間遅れ系サービスに対する Notification 競合:先に要求されたプロパティが目標値に達しないまま異なるプロパティの制御に移行する可能性がある. 室温と湿度の検出のように異なる Notification が同じサービスに送信された場合に発生する.
5) 外部環境への依存:サービス結果が外部環境の状態により変化する. 例として窓やカーテンが挙げられる.
6) 対話型処理と自律型処理の競合:2 つの制御機構の存在により, Notification とユーザによるリクエストが同時に受信される可能性がある.

(2) 対応策

1) 外部環境への依存:Subscription により外部環境への依存性を関連付け，NS 内で外部と室内環境の差異からサービスの作用方向を決定し，Notification と照合する.

2) 対話型処理と自律型処理の競合:リクエストはユーザの直近の要求であるため，リクエストを優先する.

5.4. 自律協調型 SOA の振舞い

ユースケースは対話型と自律型の 2 通りある(図 9).

(1) 対話型処理:ユーザが物理的に操作を行う場合

1) User が Service を指定し Request Service に要求

2) Request Service が Service を起動

3) Service が User に作用

(2) 自律型処理:センサにより情報が検出された場合. Sensor による情報検出を Event とみなし，パブリッシュ/サブスクライブモデルに基づき処理を行う.

1) システムに加入したサービスは予め関心のあるイベントを Subscription として NS に通知する

2) 環境からの Event を Sensor が検出する

3) Sensor は Event 発生と関連するエンティティの状態を Notification として NS へ通知する

4) NS は Subscription と Notification を照合し，関連するサービスのみに Notification を仲介する

5) Notification を受け取った Service が起動される

6) Service が User に作用する

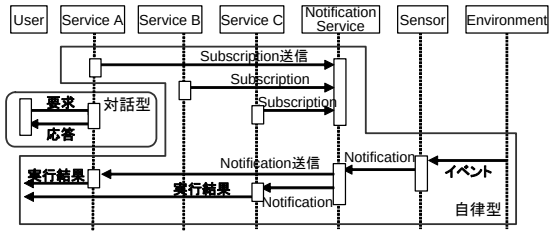

図 9　構成要素間の相互作用

6. 考察

本稿は 2 つの性質に着目してサービス提供を分離した. HNS では制御するプロパティの性質とその相互作用，副作用や競合，外部環境からの作用の有無などが異なるサービス提供が混在する. これらを分離することで，有効な相互作用モデルを選択できる. 自律処理に対するイベント駆動モデル適用により，センサにより同時多発的に発生するプロパティ検出を順次的に処理可能になり効率向上が図れる.

対話型と自律型の処理が同時発生する場合が考えられるが，ユーザ操作は直近のユーザ要求を示すものであり，対話処理を優先して行う必要がある. また，対話処理で変更されたプロパティを自律処理の新たな目標値とすることで，ユーザに適応したサービス提供が実現可能である.

ユーザプロパティを利用することにより，ユーザに適応したサービス提供が実現可能である. これにより，家庭環境の快適性，利便性の向上が期待できる.

また，本稿で提案するサービス提供モデルはオフィスなどの類似環境下においても適用可能であると期待する.

7. 関連研究

文献[3, 4]は機能とその協調関係に着目したサービス提供を提案する. これらは予め決められた実行順序でサービスを起動するため，状況に応じた臨機応変なサービス提供は不可能である. 本稿ではエンティティの状態を中心にサービス提供をモデル化することで，対象とするプロパティを変更可能なサービスを状況に応じて起動可能である.

文献[2]はプロパティ中心にサービス提供をモデル化している点は同様であるが，本稿で示したユーザプロパティのようなユーザ要求をサービス提供に利用するための仕組みは提案されていない.

8. 今後の課題

副作用やサービス競合などの解消方法の検討が必要である. また，イベント処理に利用する Notification や Subscription の詳細を定義する必要がある. 今後 HNS のサービスの多様化に伴い，分離の観点を再検討する必要がある.

9. まとめ

HNS におけるエンティティ間の相互作用の複雑化を解消するために，本稿ではサービス提供を 2 つの特性によって分離し，それぞれに対して有効な相互作用モデルを適用し，システム全体としてはユーザ操作による対話型のサービス提供も，システムの自律的制御によるサービス提供も統合して扱う自律協調型 SOA を提案した.

参考文献

[1] E. Aarts, et al., Algorithm in Ambient Intelligence, Ambient Intelligence, Springer-Verlag, 2005, pp. 349-373.

[2] 青山 幹雄，ほか，アンビエントサービス環境上の連続的サービス提供とその評価，情報処理学会 SES 2007 論文集, Aug. 2007, pp. 109-117.

[3] 藤山 麻衣，ほか，SOA に基づく情報家電サービス連携の自動生成，情報処理学会第 152 回ソフトウェア工学研究会, Vol. 2006-SE-152, May 2006, pp. 49-56.

[4] 井垣 宏 ほか，サービス指向アーキテクチャを用いたネットワーク家電連携サービスの開発，情報処理学会論文誌, Vol. 46, No. 2, Feb. 2005, pp. 314-325.