

サービス主導型コンテキストウェアネスにおけるコンテキストの再利用

山田大輔、中尾太郎、中村竜也
株式会社NTT データ 技術開発本部
{yamadad, nakaotr, nakamurattc}@nttdata.co.jp

近年、電子タグやセンシング技術を基盤とするユビキタスコンピューティング環境が整備されるのに従い、ユーザやモノの置かれた状況や環境をコンテキストとして捉え、情報サービスや情報を提供するコンテキストウェアネスが注目されている。筆者らは、情報からボトムアップ方式でコンテキストを構成する従来のアプローチではなく、サービスの提供者がコンテキストを主観的に捉え、サービス主導でトップダウン的にコンテキストをモデリングする手法を提案してきた。本稿では、サービス主導型コンテキストウェアネスにおけるコンテキストの再利用を実現する方法について述べる。

Context Reuse in the Service Oriented Context Awareness

Daisuke Yamada, Taro Nakao, Tatsuya Nakamura
NTT DATA CORPORATION, Research and Development Headquarters
{yamadad, nakaotr, nakamurattc}@nttdata.co.jp

Differing from conventional approaches of context awareness which compose contexts with limited basal vocabulary in a bottom-up manner, we have proposed a service-oriented context awareness that supports heterogeneous data sources and allows service providers to flexibly compose contexts taking their services into account. In this paper, we introduce our approach to realize reuse of context in the service-oriented context awareness and describe its implementation.

1 はじめに

近年、電子タグ、センサネットワークに代表されるセンシング技術の高度化と、情報管理技術の進展により、ユーザやモノの置かれた環境や状況をコンテキストとしてとらえ、能動的な情報サービス提供の制御のために利用するコンテキストウェアネスに対する注目が高まりつつある[1]。図 1 にコンテキストウェアネスの概念図を示す。

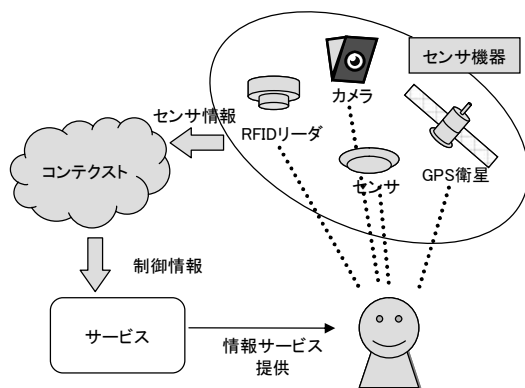


図 1 コンテキストウェアネス概念図

コンテキストウェアネスのゴールの一つに、実世界情報の

監視を代行することで、ユビキタスコンピューティング時代のサービス開発コストを下げることもある。この実現によって、センサネットワークを運営する者や、CRM 情報を保持する者、SCM を運営し、商品の物流情報をもつ者などの情報保持者と、サービスロジックを作成するサービス提供者が、互いに独立して存在しながらも、コンテキストウェアネスを構成できるようになる。すなわち、情報の生成論理と情報の利用論理を分離しながらも、状況に応じたサービスを提供できるようになる。情報の生成論理と情報の利用論理の分離のためには、実世界から取得される各種情報を抽象化し、コンテキストとしてモデリングすることが欠かせない。

これまでに、Schilit[2]やDey[3]らは、情報からコンテキストをモデリングする際の画一的な枠組みを策定しようとした。しかし、モデリングしようとするコンテキストが複雑化すれば、計算機にコンテキストを理解させることは困難となる[4][5]。

こうした、コンテキストを統一されたやり方でボトムアップ方式によってモデリングしようとする従来のアプローチとは異なり、筆者らは、サービス提供者がそのサービスの設計時に、そのサービスにおいてコンテキストとして利用する情報を選定し、コンテキスト判断ロジックも提供することで、サービス提供者が必要に応じてコンテキストを次々にモデリングしていくサービ

ス主導型のコンテキストウェアネスを提案してきた[6][7]。

サービス主導型のコンテキストウェアネスでは、柔軟なコンテキストモデリングが可能になるだけでなく、計算機による自律的な環境理解に頼る必要がないため、サービス提供者の思惑に沿った複雑なコンテキストのモデリングが可能となる。

サービスが次々に作成されれば、それに伴ってさまざまなコンテキストがサービス提供者によって構成されることになる。こうして構成されたコンテキストは、それぞれが、収集した情報やユーザプロフィールをもとに、サービス提供者の視点から環境を理解したものであるといえる。こうしたコンテキストを、あらためて抽象度の上がった環境情報として利用し、別のコンテキストを構成できれば、既存の環境理解を活用した、より抽象度の高いコンテキストモデリングが可能となる。

本稿では、サービス主導型コンテキストウェアネスにおいてコンテキストの再利用を可能にするコンテキストの構成方法と、その実現方法について述べる。

以下、2章でサービス主導型コンテキストウェアネスについて述べ、3章でそれを実現するために筆者らが開発したフレームワーク YACAN(Yet Another Context AwareNess)について述べる。4章ではサービス主導型コンテキストウェアネスにおいてコンテキストの再利用を可能にするコンテキストの構成方法について説明する。最後に5章で、提案手法に関する今後の課題に触れるとともに本稿をまとめる。

2 サービス主導型コンテキストウェアネス

実世界から得られる情報の多様化に伴い、利用されるコンテキストは次第に複雑化してきた。そこで、実世界あるいは仮想世界のあらゆる事象や状況を、一般的な方法で計算機可読なコンテキストとして表現する試みがこれまでにいくつかなされてきた[8][9]。

Schilitらは、コンテキストを computing context, user context, physical context の3つのカテゴリに分類することでコンテキストの記述を一般化しようとした[2]。しかし、Chenらによって time context や context history という概念の不足が指摘されるなど、このアプローチは不完全に終わった[4]。Deyらは、primary contexts (location, entity, activity, time)をインデックスとして、より複雑な contextual information をボトムアップ的に構成するアプローチを提案した[3]。しかし、Schilitの手法と同様に、定義した4つのインデックスの範疇に収まらないコンテキストは依然多数存在する。

筆者らは、「あるコンテキストを利用しようとする者は、どのような情報をどう処理すれば目的のコンテキストを得られるかを理解している」という考察に基づき、サービス提供者がトップダウン的に目的のコンテキストをモデリングするサービス主導型のコンテキストモデリングを提案してきた[6][7]。サービス主

導型コンテキストウェアネスでは、サービス提供者が利用するコンテキストをモデリングする。

例えば、会議室に集まった人たちが、議論をしているのか、誰かのプレゼンテーションを聞いているのかを、計算機が自律的に情報収集して判断を下し、汎用的なコンテキストとして準備しておくことは難しい。しかし、その会議室にコンテキストウェアなサービスを提供しようとする者が、サービス提供のきっかけとなるコンテキストを、「椅子の圧力センサが複数反応していて、部屋の数箇所に設置されている音声センサから得られる音源方向の変化頻度がある一定程度以上を継続している状態を議論している状態とする」などとして定義することは容易である。

このようなサービス主導型のコンテキストモデリングを実現するためには、コンテキストモデリングの方法を確立すること、そして実世界の環境や状況の変化に応じて、モデリングされたコンテキストに紐付けられているサービスを適切に活性化させる枠組みが必要となる。

3 YACAN

YACANは、筆者らが開発しているサービス主導型コンテキストウェアネスを実現するフレームワークである。図2にYACANが想定しているコンテキストウェアサービスの参照モデルを示す。参照モデルは物理層、コンテンツ層、コンテキスト層、サービス層の4層から成る。

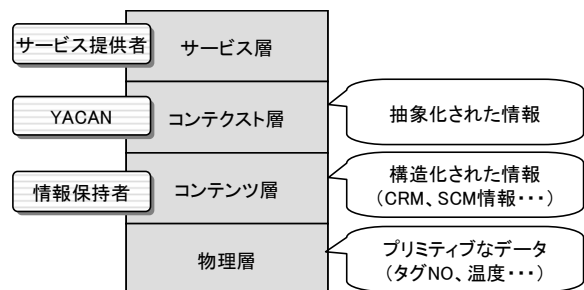


図2 コンテキストウェアサービスの参照モデル

物理層は、センサネットワークにおいてセンサから取得できるタグ情報や温度などプリミティブなデータを扱う層である。コンテンツ層は、物理層から得られるプリミティブなデータを構造化し、計算機可読な形にした情報を扱う層である。以降、この構造をもつ計算機可読な情報をコンテンツと呼ぶ。コンテンツの構造はスキーマとして公開されているものとする。センサネットワークやCRM、SCMなどのYACAN外部のアクタが提供する情報をコンテンツとして想定している。

コンテキスト層では、コンテンツの変化と、それによって活性化されるサービスの紐付けを行う。YACANは、コンテンツからコンテキストを構成し、サービス提供者に代わって実世界

または仮想世界の事象や状況の変化を監視することでコンテキスト層を担い、環境や状況に応じたサービストリガリングを提供する。

サービス層では、コンテキスト層からのトリガリングに応じて、サービスロジックを実行し、ユーザにサービスを提供する。この層のアクタとして、サービス提供者を YACAN 外部に想定している。

YACAN の構成を図 3 に示す。YACAN は、Content Provider、Content Broker、Context Manager の 3 つのコンポーネントから成る。

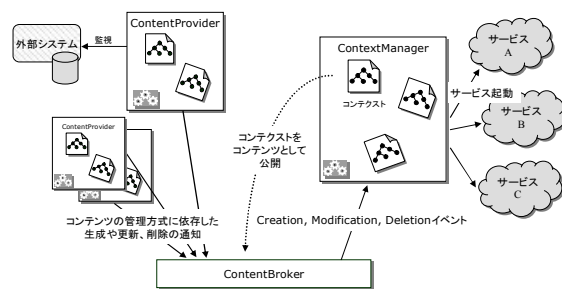


図 3 YACAN 構成図

Content Provider: Content Provider は、コンテンツを保持するシステムにおいて、YACAN に対して自律的にコンテンツの変化イベント(生成・更新・消滅)を通知する機能を持ち、状況や環境データなどのコンテンツを提供するシステムである。イベント通知機能は Content Provider がもつ場合と、YACAN がアダプタとして提供する場合がある。後者の場合、コンテンツを保持するシステムに対して、YACAN が提供するアダプタが監視を行い、アダプタが必要に応じて変化イベントを YACAN 本体に通知することになる。Content Provider は、データベースや web ページなど、任意の形式で状況や環境データを管理しており、任意のデータは URI で参照できるものとする。コンテンツの変化イベントは Content Broker に任意の方法で通知される。

Content Broker: Content Broker は、Content Provider の多様性を吸収し、Context Manager に統一された方式でコンテンツの変化イベントを通知する。Content Broker によってイベント伝播の経路制御が行われる。

Context Manager: Context Manager は Content Broker からコンテンツの変化イベントを受け取り、コンテキストに従ってサービスの起動を行う。Context Manager は、サービス提供者が記述したコンテキストのスキーマを格納するコンテキストデータベースをもつ。Context Manager はイベントの到着時に、コンテキストデータベースから適用

可能なコンテキストスキーマを取得する。そのコンテキストスキーマに、実際のコンテンツを代入することで、コンテキストをインスタンス化する。Context Manager は、インスタンス化したコンテキストを管理し、Content Broker からの通知イベントに従い、コンテキストに対応付けられたサービスアプリケーションを起動する。

YACAN では、異なる適用領域で定義されたコンテキストの共用性、発展性、相互運用性などの要件を満たすために、ウェブオントロジー言語 OWL (Web Ontology Language) [10]によってコンテキストを記述している。

4 コンテキストの再利用

サービス主導型コンテキストアウェアネスにおいて、サービス提供者によってモデリングされたコンテキスト自身も、事象や状況を抽象化して表したコンテンツの一種であるといえる。コンテンツを抽象化したコンテキストをあらためてコンテンツとして提供できれば、コンテキスト設計時に利用可能なコンテンツが増えるとともに、抽象度の高いコンテキストのモデリングが可能となる。

例えば、あらたに「テレビ会議を行っている状態」をコンテキストとしてモデリングしたい場合、本来ならば、「椅子の圧力センサと部屋に設置されている音声センサが共に反応していて、さらにテレビ会議の回線が利用されている状態を、テレビ会議を行っている状態とする」といった定義が必要になる。ここで、椅子の圧力センサや部屋の音声センサから得られる情報から抽象化した「議論している状態」を定義したコンテキストがあれば、「議論をしている状態で、テレビ会議の回線が利用されている状態を、テレビ会議を行っている状態とする」といったように、各種センサの状態を直接扱わずにコンテキストを定義できる。事象や状況の抽象度を高めた既存のコンテキストを再利用することで、複雑な事象や状況のコンテキスト表現が容易になる。

4.1 コンテキスト記述

コンテキストの再利用を可能にするためには、コンテキストに、コンテキストアウェアなサービスを提供するためにプラットフォームに対して行う定義の記述に加えて、将来的な再利用を可能にするための内容の記述が必要となる。そこで、次の項目をコンテキストとして記述するようにコンテキスト記述を拡張する。

Trigger Description: サービスがサービス提供の契機とする状況の記述。サービス提供の契機とする状況は、監視対象のコンテンツと、それらの監視条件の指定によって定義する。コンテンツの指定と、コンテンツの値を参

照するために、コンテキストが利用するコンテンツの構造は全て既知でなければならない。また、全てのコンテンツは URI により参照できなければならない。

図 4に、前述の「議論している状態」を表すコンテキスト Discussion Context をモデリングした例を示す。Discussion Context においては、椅子に内蔵された圧力センサの情報と、部屋に設置された音声センサの情報を、コンテンツ Pressure Sensor と Sound Sensor として Trigger Description に定義している。

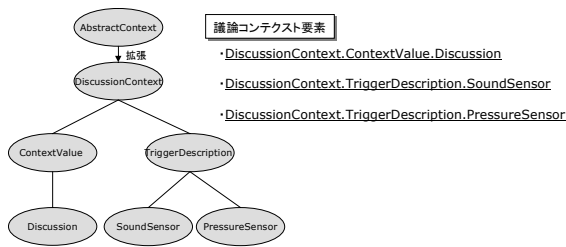


図 4 Discussion Context

Context Value: コンテキストに紐づいたサービスによって生成されるコンテンツの構造を記述する。このコンテンツに対する URI を用いた参照を可能にすることで、このコンテンツを別のコンテキストの構成要素として利用できるようになる。

何を Context Value とするかはサービスの設計者が任意で決定する。Context Value の値は、サービスロジックが作成する。

図 4の Discussion Context では、単純に“議論中”であるかどうかを示す Boolean 値をコンテンツとして公開している。

全てのコンテキストは、コンテキストに共通する基本的な振る舞いが定義されたアブストラクトな唯一のコンテキストを拡張するかたちでモデリングされる。

また、任意のコンテキストは拡張可能であり、同じ振る舞いを継承するような場合、異なるサービスに適用するよう拡張できる。

4.2 コンテキストのインスタンス化

モデル化されたコンテキストは構造しかもたない、いわばコンテキストのスキーマである。図 5に椅子の情報を構造化した Chair Content を示す。Chair Content は椅子の製造者や埋め込まれたセンサ情報をもつ。

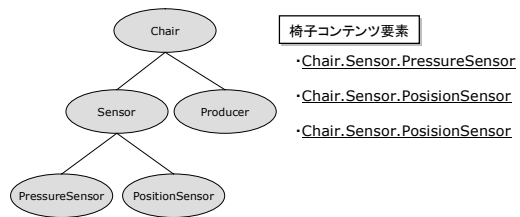


図 5 Chair Content

コンテキストのスキーマの属性に、コンテキストが実際に監視する対象となるコンテンツの実データを代入することで、コンテキストをインスタンス化する。YACAN はインスタンス化されたコンテキストに基づいて動作する。

YACAN では、監視対象とするコンテンツへの URI を設定することでコンテキストをインスタンス化し、YACAN による監視を開始するものとしている。YACAN は、コンテキストの属性で指し示されたコンテンツが生成・更新・削除された場合、それを検知し、コンテキストをトリガする。コンテキストのトリガは、そのコンテキストに対応するサービスの起動のトリガとなる。

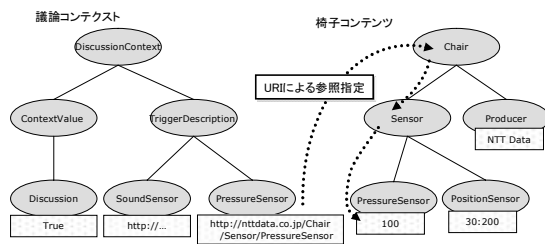


図 6 コンテキストのインスタンス化

図 6に Chair Content によってインスタンス化された Discussion Context を示す。この場合、Discussion Context の Pressure Sensor として、Chair Content の URI が代入されており、Chair Content の Pressure Sensor の値が Discussion Context における Pressure Sensor 要素の監視対象となっている。

コンテキストはインスタンス化されることではじめて、参照可能な Context Value をもつことになり、コンテンツとしての再利用が可能となる。

図 6の場合、インスタンス化した Discussion Context の Discussion の値を URI 参照することで、Sound Sensor や Pressure Sensor といった椅子コンテンツの詳細を知らなくても、その部屋で議論が行われているかどうかを監視する新たなコンテキストをモデリングできるようになる。

ただ、Context Value の再利用に際しては、基本的にそのコンテキストがどのようなコンテンツから構成されているのか、あるいは議論コンテキストがどのようなロジックで Discussion の値を算出しているかという情報が隠蔽されてしまう。Context Value が自分自身を説明できるような記述の枠組みがあれば、

コンテキストの積極的な再利用が進むだろう。

4.3 実装

YACAN において、コンテキストの再利用を可能にするコンテキスト記述の拡張を行った。Context Manager が拡張コンテキストを解釈できるようにするとともに、生成される Context Value を再利用可能なコンテキストとして保持・提供するための Context Repository を新たに用意した。

Context Repository は、Context Value をそのスキーマとともに格納し、URI による参照を受け付ける。さらに、Context Value の変化イベントを YACAN に対して通知する機能をもつことで Content Provider として振る舞う。

図 7、図 8 に Context Repository を利用したコンテキストの再利用の例を示す。図 7 は TV 会議が行われているかどうかを示す TV Conference Context である。Trigger Description とし Discussion を監視することが定義されている。

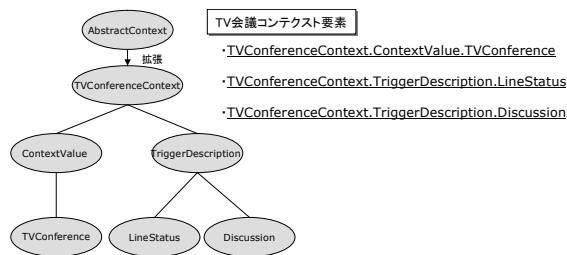


図 7 TV Conference Context

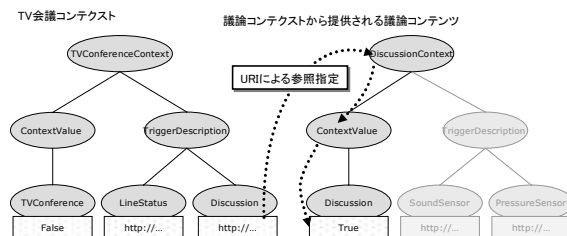


図 8 議論コンテキストの再利用

この Discussion 属性に、図 8 に示すように Discussion Context の Context Value である Discussion の値を代入してインスタンス化することで、TV Conference Context の設計者は、Discussion Context の導出過程の詳細にとらわれずに、その部屋で TV 会議が行われているかどうかというコンテキストをモデリングできている。

TV Conference Context は、Line Status と Discussion の二つの属性の変化によって発火する。Discussion Context によって Discussion をインスタンス化しているこの場合、Discussion Context が監視しているセンサ情報の変化が Discussion Context を発火させ、それに紐づいたサービスが Discussion Context の Context Value である Discussion の値を変化させ、

それに対応して TV Conference Context が発火するというシーケンスで TV Conference Context に紐づいているサービスが起動する。

5 まとめ

本論文では、サービス主導型のコンテキストウェアネスにおけるコンテキストの再利用について論じた。さらに、サービス主導型のコンテキストウェアフレームワークである YACAN においてコンテキストの再利用を実現する方法を提案し、その実装を示した。

サービス主導型コンテキストウェアネスでは、サービス提供者によってモデリングされたコンテキストは、サービス提供者による環境や状況の理解ロジックを内包している。

コンテキストをプリミティブな情報から一段階抽象度が高まったコンテンツとして扱い、再利用することでコンテキストの語彙の幅を広げることが可能である。そうしてサービス提供者による環境や状況の解釈を蓄積することで、あらゆる状況をコンテキストとして定義することができるようになって考えている。

現在の実装では、コンテキストの再利用を機能的に実現した。しかし、新しいコンテキストを設計する際に、Context Repository から適用可能なコンテキストを探してくるコンテキストディスカバリーや、再利用しようとしているコンテキストの詳細情報の参照などによって、コンテキストの再利用を支援する枠組みが不十分である。今後、OWL の機能を用いることで、オントロジーを活用したコンテキストディスカバリーやコンテキストの自己記述を実現することを検討している。

謝辞

本研究を進める上で貴重な御助言を頂きました ATR 知能ロボティクス研究所の小暮潔博士、桑原和宏博士、NTT コミュニケーション科学基礎研究所の岡留剛博士、平松薫博士ならびに、ご助力頂きました NTT データ先端技術株式会社の落田征士氏、株式会社アスケイドの代表取締役池田寛氏に謝意を表します。

参考文献

- [1] M. Weiser, "The computer of the 21st century," Scientific American, September 1991, pp. 94-100
- [2] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," Proc. of IEEE Workshop on Mobile Computing Systems and Applications, IEEE Computer Society Press, Dec. 1994, pp. 85-90.
- [3] A. K. Dey, "Understanding and Using Context" Personal and Ubiquitous Computing, Vol. 5, No. 1, 2001, pp. 4-7.
- [4] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile

Computing Research,” Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, November 2000.
<ftp://ftp.cs.dartmouth.edu/TR/TR2000-381.ps.Z>

[5] S. Hamad, “The Symbol Grounding Problem”, Physica D, Vol. 42, 1990, pp. 335-346.

[6] 中村竜也, 山田大輔, 中尾太郎, “コンテキストウェアコンピューティングとコンテキストの定式化”, 人工知能学会 第7回セマンティックとオントロジー研究会, SIG-SWO-A402-03, July 2004.

<http://www.jaist.ac.jp/ks/labs/kbs-lab/sig-swo/papers/SIG-SWO-A402/SIG-SWO-A402-03.pdf>

[7] 山田大輔, 中村竜也, 中山一美, “コンテキストの構造化に関する考察とコンテキストウェアフレームワーク”, 情報処理学会研究報告, 2004-UBI-5, pp. 63-70, Jun, 2004.

[8] R. Want et al. “The Active Badge Location System” ACM Trans. Information Systems, Vol. 10, No. 1, pp. 91-102, 1992.

[9] R. Want et al. “The PARCTab Ubiquitous Computing Experiment”, Technical Report CSL-95-1, Xerox Palo Alto Research Center, 1995.

[10] Web Ontology Language

<http://www.w3.org/2001/sw/>