

適応型通信サービスにおけるコンテキスト情報提供方式

中村 哲也[†] 松尾 真人[†] 板生 知子[†]

ユーザにとって最適なマルチメディア通信サービスを提供するために、複数のネットワーク上の情報源を介してユーザを取り巻く状況(コンテキスト)を把握しアプリケーションに提供することのできる、コンテキスト情報提供方式を提案する。複数のネットワーク上に存在するセンサーなどの情報源を扱い、必要な情報が取得不可能な場合にも代替コンテキスト情報を提供することにより、ユーザの動的な状況変化に応じた柔軟な情報取得を可能にするとともに、新たなコンテキストを活用したサービスの開発・提供を容易にする。本論文では、多種多様なコンテキストに関する情報を統一的に扱うために導入した情報モデルと、モデルに基づいたコンテキスト情報提供方式の概要について述べる。

Context handling architecture for adaptive networking services

TETSUYA NAKAMURA,[†] MASATO MATSUO[†] and TOMOKO ITAO[†]

This paper proposes a method to get information about context, that is the environment or situation of user, from various information resources via networks, and to provide it to the applications. If a required context information is not obtained initially, this method provides substitutive context information obtained from the alternative resource. So it makes it possible to handle context information flexibly under the condition that user's context change dynamically. It also makes it possible to design various context aware applications easily. This paper describes the common model which is abstracted from different type of contexts to handle context information uniformly, and the concept of the context handling architecture in which the model is based.

1. はじめに

ウェアラブルデバイスに代表される各種モバイルデバイスや分散処理の技術革新を背景に、ユーザを取り巻く状況(コンテキスト)に適応した数多くのマルチメディア通信サービス(以降、コンテキストウェアサービス)が開発されつつある。ここで“コンテキスト”(Context)とは、ユーザの位置や興味、ユーザ履歴、周囲のリソースの可用性、リソースの負荷状況などで、これを利用することで、サービスをユーザにとってより最適なものにする事が可能になる^{1)~3)}。

しかし現状のコンテキストウェアサービスにおいては、対象とするコンテキストが限定されており、サービスをよりユーザにとって便利にするために有効なコンテキストに関する情報が存在していても、それが反映されないままユーザはサービスを楽しんでいる。また情報を取得する情報源はサービスに特化したものであり、情報源が故障などするとサービスが提供不可能になる。

このような問題に対し、ネットワーク自身がコンテキストに関する様々な情報を取得し、管理することで、ユーザ自身がコンテキストを意識しなくとも、よりユーザの状況に適したサービスをネットワークの側から提供することが可能となる⁴⁾。また故障などによる情報源の変化や、ユーザの動的な状況変化にも対応できるようになる。

これによって将来、複数のネットワークに接続されたセンサーなどのあらゆる情報源を結合し、ネットワークの中でそれらの情報を取捨選択しながらサービスに活用することで、ユーザがどこにいても、どんな状況でも、ユーザはその場その時に最適なサービスを楽しむことができるようになる。また、ネットワークが汎用的に用意した情報を活用することにより、多様なサービス提供が容易になる。

そこで本稿では、ネットワークの中の様々な情報源から、コンテキストに関する情報を取得・活用するコンテキスト情報提供方式を提案し、コンテキストウェアサービスを容易に提供できる環境の実現を目指す。

[†] NTT 未来ネットワーク研究所
NTT Network Innovation Laboratories

2. 背景

2.1 コンテキストウェアサービス

コンテキストウェアサービスについて、展示会などにおけるガイドサービス⁵⁾を例にとって具体的に説明する。これは、位置情報に適應してサービスを提供するものである。各ユーザは展示会の入り口で位置情報をセンサーで把握させるためのバッジと携帯端末を受け取る。ユーザがある絵画の前まで行くと、赤外線センサーがユーザがその絵画の前にいることを検知しシステムに通知する。するとガイドサービスはユーザの興味や理解に応じた解説を、そのユーザが保持する携帯端末に流す。またある特定の絵が見たいと携帯端末に要求すると、バッジにより検知したユーザの現在地からその絵のある場所までの道順を携帯端末に表示してくれる。この例の中で出てきたユーザの位置や興味・知識が、ガイドサービスで用いられるコンテキストである。

2.2 現状の問題点

コンテキストに関する情報を収集、活用するための機構(コンテキストハンドリング機構)に関する研究^{2),3),5),6)}は、現状では上述したガイドサービスなど、限定された場所、限定されたコンテキストに関する情報のみを対象としているため、次の問題点がある。

問題点 コンテキストに関する情報の取得方法や活用方法が、静的に一意に決まっいて、アプリケーション(サービスを表現するソフトウェアを、アプリケーションと呼ぶ)がコンテキストを直接制御する必要がある。これにより、

1. 状況に応じた柔軟な対応が出来ない。例えば、ユーザが身につけるバッジや、センサーなどの情報源が故障すると、サービス提供が不可能になる。またその情報源は、複数のシステムにまたがった分散したネットワーク上に存在するものではなく、ある1つのシステムに閉じたものである。
2. あるサービスにおいて参照されるコンテキストに関する情報は、その機構固有の形式で蓄積されるので、その情報を用いた新たなアプリケーションの開発が困難である。例えば上記のガイドサービスにおけるコンテキストの制御システムを、他の同種のガイドサービスへ流用することは困難である。

このように現状では、これらコンテキストに関する情報を取得し活用する技術が、個々のアプリケーションの中で個別に実現されている。いわば対象とするサービスに閉じた機構となっている。

3. コンテキストハンドリング機構の概要

3.1 目的

先述の問題点に対し、ネットワーク自身がコンテキストに関する様々な情報を取得し、管理することで、故障などによる情報源の変化やユーザの動的な状況変化にも対応できるコンテキスト情報提供方式を提案する。

このために、複数のネットワークに接続されたセンサーなどのあらゆる情報源を結合し、ネットワークの中でそれらの情報を取捨選択しながらサービスに活用するコンテキストハンドリング機構の構築を目的とする。

3.1.1 例題サービス

本稿がめざすネットワークワイドのサービスとして、今後のサービス充実が期待される、遠隔医療・介護を取り上げた。以下にそのサービスシナリオを示す。

患者が病院に出かける際、診察予約サービスにアクセスする。患者はバスに乗り病院に向かう。診察予約サービスの要求に従い、システムは家庭内で取得した生体情報や、バスの中での患者のリアルタイムな生体情報を、ネットワークを介してセキュリティ上安全に取得・提供する。また患者の現在位置を把握し、病院への到着を知らせる。診察予約サービスは、患者の到着を待って、直ちにしかるべき診療科に案内する。

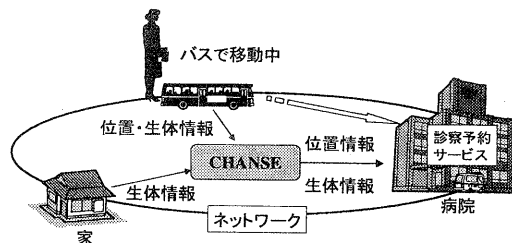


図1 遠隔医療・介護サービス例

例として挙げたサービスは、既存のシステムでは実現が困難である。即ち情報源からのデータの取得が不可能な時に、他の代替情報源からのデータ取得はなされないため、サービス継続が困難になる。

さらに位置と生体情報という、複数の種類のコンテキストをアプリケーション非依存に扱うことにより、上で挙げた遠隔医療のサービス例で用いるのと同じ種類のコンテキストを必要とする他のサービスからの利用が可能になる。例えば、生体情報を用いて、ダイエットに必要な減量プログラムを提示したり、位置情報から適切な

フィットネスクラブを紹介してくれるようなサービスなどが考えられる。

3.2 機構実現のための考え方

コンテキストに関する情報を活用しようとする際、取得できる情報は状況に応じて変化したり、他の情報で代替できるケースも十分あり得る。例えば、“ユーザ位置”というコンテキストに対し、ガイドサービスのようにユーザがバッチを持っていれば、そこから“部屋”を特定したり、GPSを使えば“緯度経度”を取得するなど、状況に応じた情報の取得方法の選択が重要となる。また取得した情報の解釈自体も、アプリケーションによっては、単に“地区”がわかればよいレベルから、“建物名”、“部屋番号”まで特定する必要がある場合も考えられ、状況に応じて解釈の仕方を使い分けることが必要となる。

そこで 2.2 節の問題点の解決に向けて次のように考える。

- (1) 1つのコンテキストに対し、代替可能なコンテキストに関する情報、またその情報源を複数用意しておく。そしてアプリケーションや状況に応じて情報の取得方法と解釈方法を組み合わせる。(問題点 1. に対応)
- (2) アプリケーションに対し、汎用的なコンテキストに関する情報の参照・利用手段を規定することで、情報の取得方法をアプリケーションから隠蔽する。(問題点 2. に対応)

本稿では、アプリケーションが活用できる、意味的な情報を“コンテキスト情報(Context Information; CI)”と呼ぶ。これに対しセンサーなどによるモニタリングから得られる生データや、データベース参照により得られるデータなど、CIの元となる情報を“コンテキストデータ(Context Data; CD)”と呼ぶ。CIとCDの導入により、上記(1)、(2)は次のようにモデル化できる。

- (1) 1つのCIに対し代替可能な複数のCDを用意しておき、状況に応じて適切なCDを選択し、CIにマッピングする。
- (2) アプリケーションに対しCDを隠蔽し、CIを参照するための利用手段のみを提供する。

これにより以下の効果が期待できる。

● サービス環境実現の支援

取得するCDを状況に応じて変更できる。すなわち特定のCDがなくても代替のCDを用いることで、必要なCIを生成し、サービスの実行を可能にする。これにより、ユーザにとっては、いつでもどこでも、自分の要求やコンテキストにマッチしたサービスを楽しむことが可能になる。

● アプリケーション開発の支援

アプリケーションは、ネットワーク内部の構造や情報源の種類などを意識せず、規定されたAPI(Application Program Interface)の関数を呼び出すための引数のみを指定すればいいので、構築・開発が容易になる。

3.3 CHANSE の提案

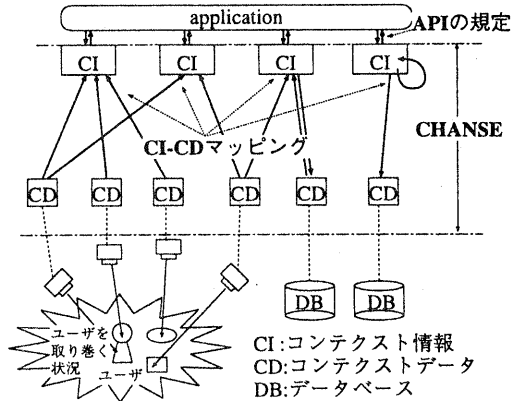


図2 コンテキストハンドリング機構 CHANSE

3.2 節で述べた解決案に基づくコンテキストハンドリング機構、“CHANSE (Context HANDling Service Environment)”を提案する(図2)。また CHANSE による処理の流れを図3で示す。(1)アプリケーションか

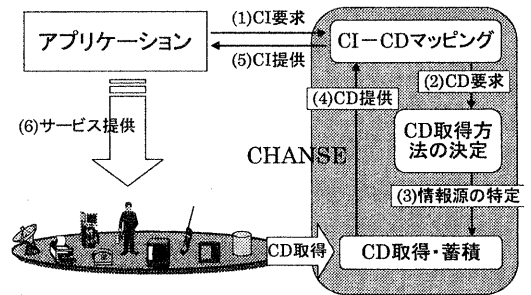


図3 処理の流れ

らのCIの要求に対し、(2)CI-CDマッピング機構で必要なCDを決定する。この際、1つのCIに対し複数の対応するCD、CDからのCI生成方法を用意しておき、どのCDを取得すべきかを決定する機能を有する。(3)CDが決定すると、それを取得するために必要な情

報源を特定する。ここで CD を取得するための情報源として、センサーなどのユーザをモニタリングするもの他に、履歴情報などを蓄積しているデータベースや、ネットワークに結合したリソース（ハードウェア・ソフトウェア etc.）を扱う。(4) 情報源から CD が取得できれば CI-CD マッピング機構で CI に解釈し、(5) アプリケーションに提供する。もしいずれの CD も取得することが不可能な場合は、CI の履歴情報や関連 CI などから CI を生成する。(6) アプリケーションは CHANSE から提供される CI を活用してサービスをユーザに提供することになる。

3.4 CHANSE の課題

CHANSE の実装に向けて以下が重要となる。

- CI-CD マッピングの処理方式の明確化
 - CD 取得方法の明確化
 - CI 参照の API の規定
- これらの課題に対し、以下で検討する。

4. CHANSE の基本概念

4.1 CI のモデル化

アプリケーション非依存なコンテキスト情報提供方式を実現しようとする時、重要となるのはアプリケーションが要求する CI の情報構造の規定である。そのため何らかの属性で CI を抽象化して扱う必要がある。本稿では、例題サービスで必要となるコンテキストとして位置情報と生体情報を抽出した。この 2 種類のコンテキストについて、汎用化を考慮しながら、アプリケーションが必要とする CI の情報の特性を考察し、情報モデルを作成した。以下が本モデルで抽象化した CI の属性である。

カテゴリ 一般的にコンテキストを表す情報は多種多様であるため、その概念体系はコンテキストの種類毎に規定することが好ましい。それゆえ CI をコンテキストの種類毎に分類した。各々を“カテゴリ”と呼ぶ。例えば“位置”や“生体情報”は固有のカテゴリにあたる。

CI 表記 CI のカテゴリの概念体系の中で定義される情報の表現方法。同じ“位置”でも、“緯度経度”、“住所”、“建物名”で表す場合があり、アプリケーションはこれらの表記を意識する必要がある。

精度 CI 表記に対する情報の精度を表す。CI 表記として“住所”を指定しても、“都市”の精度から“番地”の精度の要求まで様々であり、必要とする精度はアプリケーションに依存する。

リアルタイム性 アプリケーションに提供する CI の実時間性である。CI がいつのデータなのかによって

その性質は変わってくる。たとえ同じ CI 表記、精度の CI であっても、その CI がリアルタイムなデータであるか、スケジュールデータベースから抽出した過去のデータなのかにより、データの信頼性は異なる。

以上挙げた 4 つの属性は、アプリケーションが CI を CHANSE に要求する際に指定する。

4.2 CD のモデル化

アプリケーションの CI の要求に対し、CHANSE が CI を生成し得る情報源（以下、CD リソース）を探索・特定できるよう、CI に対応した情報構造を CD が持つ必要がある。それゆえ CD を以下の属性で抽象化した。この属性の組を“CD ユニット (CDU)”と呼ぶ。CD リソースは CDU を用いて、自身が提供するデータに関する情報を CHANSE に登録する。

カテゴリ CD リソースが提供する CD が、どの CI のカテゴリに属するかを明示する。これにより CHANSE は、CDU 探索範囲を絞ることができる。

CD 表記 CD のデータ表現形式を“CD 表記”と呼ぶ。

これは CD リソースから CD を取得するためのプロトコルに相当する。例えば CI 表記“住所”を算出できる CD の表記としては、“電話番号”や“緯度経度”がある。CI への変換手順は CD 表記毎に異なる。

精度 データの精度が、要求されている CI を生成し得るかどうか判断できる必要がある。例えば精度“座席位置”の要求に対し、“建物”の精度までしか表現できない CD は使えない。

リアルタイム性 CI のリアルタイム性の要求に対し、CD がその条件を満たしているかどうかの判定をするために、必要である。

プロパティ 使用料金をはじめとした CD リソースに付随する情報である。他に、CD リソースがデータを更新する頻度を表すサンプリング間隔などがある。表 1 に CDU の属性を示す。CHANSE は CD リソースを CDU 毎に認識することになる。例えば PHS の位置認識システムと GPS の位置認識システムは、リアルタイム性については同じ属性値を持つが、CD 表記、精度については異なる属性値を持つ。

4.3 CI-CDU マッピング処理方式

3.1.1 節であげた例題サービスについて、CI-CDU マッピングの例を示す。CI のカテゴリとして、“位置”、“生体情報”の 2 つを規定した。

4.3.1 位置

位置の表記方法は大きく“緯度経度”、“住所”、“ラ

表1 CDUの属性

属性	属性値例
カテゴリ	位置, 生体情報, ...
CD表記	緯度経度, 電話番号, ...
精度	room, floor, < 10 m, ...
リアルタイム性	realtime, non-realtime
プロパティ	料金 = サンプリング間隔 = ...

ランドマーク”の3つに分類でき、これらを各々CI表記とした。

一方、精度については“ランドマーク”の場合，“建物名”の精度から，“部屋番号”，“座席番号”までの精度が考えられる。CI表記と精度の関係を図4で示す。

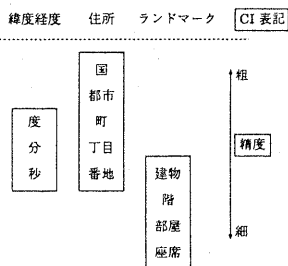


図4 位置の表記と精度

一方、位置を検出するに当たっては次のようなCDリソースが考えられる。

超音波センサ位置, 赤外線ビーコン位置, GPSによる緯度経度, PHSによるアンテナ位置, スケジュールDBからのスケジュール情報

これらの各々に、対応するCDUを規定した。CDUの例として、GPSに対応するCDUと、そのCDUが提供するCDを表2に示す。CI表記・精度とCDU表記のマッピング例を図5に示す。CDU表記からCI表記の各精度に伸びている矢印は、そのCDUがどの精度で各CIを生成できるかを表している。

表2 GPSのCDUとCDの例

C D U	カテゴリ	位置
	CD表記	緯度経度
	精度	秒
	リアルタイム性	realtime
U	プロパティ	料金 = 3円/hour サンプリング間隔 = 5 sec.
	CD	E.135,48'1", N.35,45'29"

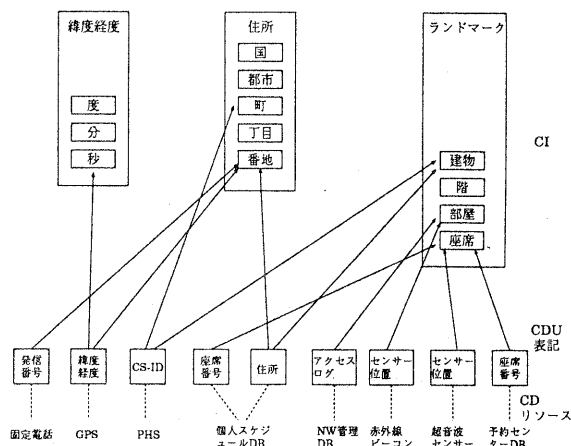


図5 位置のCI-CDUマッピング例

4.3.2 生体情報

生体情報の場合、その情報の種類は、位置と違い多数存在する。例としては、“体温”，“呼吸音”，“血圧”などが挙げられる。また“体温”，“心電図”，“加速度”の3種類のCDから判定する“心筋梗塞の前駆現象”情報のような、いくつかのCDを組み合わせる判定するCIも存在する。

生体情報におけるCIの精度は，“体温”など数値で表現される場合は、CDリソースの数値の許容誤差として表すことにする。

一方、生体情報のカテゴリに属するCDリソースとして、以下の例が挙げられる。

体温, 血圧, 心電図, ... (以上, ウェアラブルデバイス (W.D.) より), 姿勢, 歩行圧力, 呼吸音, ... (以上, 室内 センサ), 履歴 (カルテ)

表3に、CDUの一例としてウェアラブル体温計によるCDUとCDを示す。また、図6にCI表記毎のCIとCDUのマッピング例を示す。

表3 ウェアラブル体温計のCDUとCDの例

C D U	カテゴリ	生体情報
	CD表記	体温
	精度	1.00×10^{-2} °C
	リアルタイム性	realtime
U	プロパティ	サンプリング間隔 = 30 min.
	CD	36.84 °C

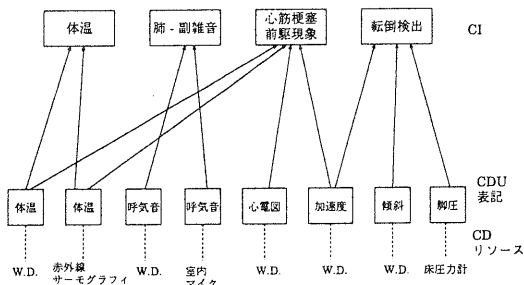


図6 生体情報のCI-CDUマッピング例

4.4 CD取得方法

4.4.1 アプリケーションのCI要求

アプリケーションはCHANSEに対して、4.1節で示したCIの各属性毎に、要求する属性値を設定する。表4に、その例を示す。表中のリアルタイム性に指定されている“***”は任意の値を許容することを示す(ワイルドカード)。“realtime”は、CIのリアルタイム性を要求する場合に指定する。

またこれらの属性値に加えて、アプリケーションがCIを要求する際にCIのサンプリング期間を要求することもある。これはCHANSEがCIを提供する際の、CIの継続提供時間である。CHANSEへの要求時点のみのCI要求もあれば、一定時間(スパン)にわたる連続データとしてCIを要求する場合もある。例えば、ある時刻におけるユーザの位置を要求するアプリケーションもあれば、心電図のように、1,2日間の連続データとしてCIを要求する場合もある。

CHANSEは、CI要求のAPIとして、表4で示される属性値を引数に持つCI要求の関数を提供する。戻り値がCIとなる。このAPIについては4.5節で再度述べる。

表4 CI要求例

属性	属性値例
カテゴリ	位置, 生体情報, など
CI表記	ランドマーク, 緯度経度, ...
精度	room, floor, ...
リアルタイム性	***, realtime, ...
サンプリング期間	point, 2 min., ...

4.4.2 マッピング処理手順

例題サービスの“位置”を例に、アプリケーションのCI要求からの処理フローを示す。例題サービスが指定するCI要求の例を表5で表す。

CHANSEはCIを生成し得るCDUを探索するための条件を予め用意している。これを“CDU条件テ

表5 例題サービスのCI要求の例

カテゴリ	位置
CI表記	ランドマーク
精度	建物
リアルタイム性	realtime
サンプリング期間	point (病院到着時)

ル”と呼ぶ。CDU条件テーブルはCI表記と精度の組み合わせ毎に用意されている。CI表記=“ランドマーク”, 精度=“建物”に対応するCDU条件テーブルを表6に示す。ここで表中のCDU条件No.は、条件が複数ある場合の識別番号である。

表6 CI表記=“ランドマーク”, 精度=“建物”のCDU条件テーブル

CDU条件No.	条件		
	CD表記	精度	リアルタイム性
1	緯度経度	秒	realtime
2	住所	番地	realtime
3	***	建物	realtime
4	***	< 20 m	realtime

またCDU条件テーブル毎に、各条件の関係を示すロジックを保持する。これを“CDU条件ロジック”と呼ぶ。CDU条件ロジックは、各条件の関係(AND条件, OR条件)を規定する。複数のCDを組み合わせて1つのCIを生成する場合、CDU条件テーブルの各条件はAND条件になる。

一方、CDリソースの故障などの原因によりCDが取得できない場合にも、状況に応じて適切な代替CDリソースを選択し、アプリケーションが要求するCIを生成するために、1つのCDU条件テーブルに代替可能な複数のCDU条件を用意する。これらの各CDU条件はOR条件であり、表6の場合、4つの条件はその内のいずれかを満たすCDUを探索することにより、要求されるCIの生成が可能となる。

例として先に生体情報のCIとして挙げた“心筋梗塞の前駆現象”について見てみると、このCIは“加速度”, “心電図”, “体温”の3種類のCDを組み合わせて生成されるので、3種類のCDU条件のCDU条件ロジックはAND条件である。その中の個々のCD(“体温”など)に対し複数のCDU条件が存在するなら、それらの各条件のロジックはOR条件である。

CI要求からCD取得までの処理を図7に示す。アプリケーションからCI要求を受けるとCDU条件テーブルを参照し、条件を満たすCDUを探索する。次にCDを取得すべきCDUを選択し、対応するCDリソースか

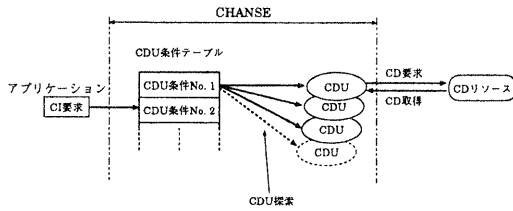


図7 マッピング処理手順

ら CD を取得する。取得した CD を CI に変換し、API における CI 要求の戻り値として、アプリケーションに提供する。

このとき、以下の考慮すべき点がある。

- (1) CHANSE は、代替可能な複数の CDU 条件を OR 条件として保持しており、すべてが同じ品質の CI を生成できるとは限らない。すなわち、リアルタイム性や CDU の精度、などの差異である。それゆえ、CI 要求により適した CDU を探索するための方式を明確にする必要がある。例えばアプリケーションが CI の精度を重視するのか、コストを重視するかで、CDU の探索方法は変わってくる。
- (2) CDU 探索で条件にマッチする CDU は、1 つとは限らない。また CDU を選択しても要求する CD が必ずしも取得できるとは限らない (センサーが故障している、など)。それゆえ複数ある CDU の候補から効率よく CDU を選択し、CD を取得する手段について考察する必要がある。

これらの各々について実現方式を検討した。具体的な評価については今後の課題とする。

4.4.3 CDU 探索

CDU 条件テーブルで用意されている複数の CDU 条件の利用方法に関する考察である。CDU を探索するには、いかに CDU 条件テーブルを活用するかが重要となる。以下の 2 つの方式が考えられる。

案 1. 予め CDU 条件テーブルの中の CDU 条件に優先順位をつける方式。この順位づけにより、優先順位の高い CDU 条件から CDU を探索する。例えば CDU 条件の“精度”、“リアルタイム性”に着目し、より正確な情報を提供し得る CDU 条件の順位付けを規定する。これにより情報の品質を保証することが可能になる。

案 2. CDU 条件テーブルに記述された CDU 条件を満たす、すべての CDU を一度に候補として探索する方式。CDU 条件テーブルでは扱わない属性 (コストやサンプリング間隔など) を重視する場合

などに有効である。すべての CDU 条件を満たす CDU を候補として挙げることで、コストなどの CDU のプロパティを参照して、実際に CD を取得する CDU を選択することが可能となる。

4.4.4 CDU 選択・CD 取得

CDU を探索すると、上の案 1,2 いずれの探索方式においても、複数の CDU の候補が挙がることもある。すると効率良く CD を取得するために、複数ある CDU の候補から CDU を特定し CD を取得する必要がある。その CD 取得手段は以下の 2 つが考えられる。

- 案 a. 優先順位の高い CDU から CD を取得する方式。CD 取得に失敗したら次の CDU から CD 取得を試みる。例えば、コストを重視した順位づけなどである。
- 案 b. 候補として挙げられた CDU の各々から、取得可能な CD を一度にすべて取得する方式。取得した CD に対して CI 生成に用いる CD を決定する。この方式は、例えば最も早く取得した CD を用いる、などの即答性が要求される際にも有効である。

4.5 CI 参照の API

4.4.1 節で示したように、CHANSE は CI 要求の API として表 4 で示される属性を引数に持つ CI 要求の関数を提供する。戻り値が CI となる。

また 4.4.3, 4.4.4 節で検討したように、CDU を探索し、CDU を選択・CD を取得するには、4 通り (1-a, 1-b, 2-a, 2-b) の組み合わせがある。検討した各方式は、情報の品質やコスト、即答性などのうち、どの観点を重視するかで使い分けることが可能になる。表 7 に、4 通りの方式の各々で重視される属性と、それらの優先度の関係を示す。

アプリケーションが CI を要求する際に設定する属性値と、重視する条件を指定するための 4 通りの API を規定した。

表 7 アプリケーションの要求条件

処理方式	要求条件	
	重視する観点	優先度の例
1-a	品質、コストなどの CDU の属性 情報の品質を重視し、CDU 候補も順位づけ	品質 > コスト
1-b	品質、(即答性) 品質重視で取得可能なもの	品質 > 即答性
2-a	コストなどの CDU の属性 品質は妥協してコストなどで順位づけ	コスト > 品質
2-b	なし、(即答性) とにかく CD を取得できればよい	即答性 > 品質、 コスト

5. おわりに

本稿では、コンテキストハンドリング機構 CHANSE において、汎用的にコンテキスト情報を扱うための抽象化モデルを考案し、CI と CD のマッピング処理方式を明らかにした。またアプリケーションの CI 要求の API について、CI 要求と優先順位の条件指定の API を規定した。

今後は本方式の詳細化に向けて、CDU 探索・CD 取得方式の比較・評価を行なう。また CDU 条件テーブルの記述、CI への変換規則の明確化を図り、実装によりフィージビリティの検証を行なう。

謝 辞

本検討を進めるにあたり御指導、御討論いただいた NTT 情報流通プラットフォーム研究所 高橋克巳氏、同サイバーソリューション研究所 平岩明氏、同未来ねつと研究所 奥川徹氏、久保田稔氏に感謝いたします。

参 考 文 献

- 1) 空一弘. ユーザ状況情報活用機構の実現と通信支援サービスへの応用. 情処研報. Vol.97, No.MBL-4, pp.33-38. Feb. 1998.
- 2) Daniel Salber, Anind K. Dey and Gregory D. Abowd. The Context Toolkit: Aiding the Development of Context-Enabled Applications. CHI'99. pp. 434-441. Pittsburgh, PA, May. 15-20, 1999.
- 3) Daniel Salber and Gregory D. Abowd. The Design and Use of a Generic Context Server. PUI'98, Perceptual User Interfaces Workshop. pp. 63-66. San Francisco, CA, Nov. 5-6, 1998.
- 4) 板生 知子, 松尾真人. 適応型ネットワーキングサービス環境 DANSE. 信学論 B-I. vol.J82-B, no.5, pp.730-739, May. 1999.
- 5) 角 康之, 江谷 為之, シドニー フェルス, ニコラ シモネ, 小林 薫, 間瀬 健二. C-MAP: A Context-Aware な展示ガイドシステムの試作. 情報処理学会論文誌. Vol.39, No.10. Oct. 1998.
- 6) Richard Hull, Philip Neaves, James Bedford-Roberts. Towards Situated Computing. HP Labs Technical Reports. HPL-97-66.
<http://www.hpl.hp.com/techreports/97/HPL-97-66>.
- 7) 中村 哲也, 松尾 真人, 板生 知子. 適応型通信サービスのためのコンテキストハンドリング機構の提案. 情処全大 2S-02, vol.3, Sep. 1999.