

階層的共有モデルにおける資料の個別化

村上 隆治[†] 横田 裕介[†] 上林 彌彦^{*†}

[†] 京都大学大学院情報学研究科 〒606-8501 京都府京都市左京区吉田本町

E-mail: †{takaharu,yy}@db.soc.i.kyoto-u.ac.jp

あらまし 共有資料に対し独自の個別化を行うことは協調作業における有効な資料活用法の一つである。我々は階層的に定義した資料共有モデルを用いてこれに対応する XML ベースの協調作業支援システムの構築を行っている。XML を用いてデータを動的に管理し資料への個別化の内容も利用者の役割の関連に応じて共有できる。利用者の役割に応じて提供されるビューの生成には多段 XSLT を用いたメカニズムを用いている。このため資料に対する個別化の内容を保持する XSL ファイルの編集が重要となる。本稿では利用者が資料に対して行う個別化操作について本モデルにおけるデータ構造とシステム構成について述べる。

キーワード XML , パーソナライゼーション , 協調型 DB , Web とインターネット , ユーザインタフェース

Personalization for a Hierarchical Document Sharing Model

Takaharu MURAKAMI[†], Yusuke YOKOTA[†], and Yahiko KAMBAYASHI^{*†}

[†] Graduate School of Informatics, Kyoto University Yoshidahonmachi, Sakyo-ku, Kyoto-shi, Kyoto,
606-8501 Japan

E-mail: †{takaharu,yy}@db.soc.i.kyoto-u.ac.jp

Abstract In cooperative work, personalization of shared documents is very much important. We have developed an XML based cooperative support system using a hierarchical document sharing model. Contents of personalized documents in this system are sharable according to the relation of users' roles. The mechanism which uses the multi stage XSLT is the method for generation of the view offered according to a user's role. For this reason, edit of the XSL file holding the contents of personalization is important. This paper describes a data structure and a system configuration for personalization of shared documents in this model.

Key words XML , Personalization , Collaborative DB , Web , Internet , User Interface

1. はじめに

コンピュータが広く社会に行き渡り、様々な業務の中で利用されている。特にここ 10 年の普及はめざましく、社会に深く根付き日常の業務に欠かせないものとなったといえる。またコンピュータネットワークの普及に伴い業務形態も複雑化してきている。この影響は電子メールをはじめとする通信手段の一般化だけにはとどまらない。一般家庭にまで普及したブロードバンドにより在宅勤務などの勤務形態も生まれ、作業への携わり方が多様化してきているといつてよいだろう。このような社会構造の変化により個々の業務に携わる人が、空間的あるいは時間的に分散するといったことも考える必要がある。しかし業務を行うにあたっては従来と同じように参加する人の間でコミュニケーションを図り、業務の方向性や利用するデータといった

面で情報の共有を図る必要がある。

このような社会情勢のなか、多くの人が関係する協調作業を支援することの重要性はより高まっているといえる。共同で作業を行うにはある程度の共通の知識を持つことは必要である。協調作業の支援を行うために必要な機能の中で、作業に用いられるデータの共有は最も重要なものである。共同で作業を行う人でもそれぞれの作業者は異なった役割や立場を持つ。作業を分業して行っている場合における作業者間の作業内容の違いや、指導者とその他の一般作業者といった立場の違いがそれにあたる。このような違いによりそれぞれの作業者の役割や立場に応じて、利用するデータに作業者たちが独自に資料にコメントを書き加えたい場合や、参照するデータを選択し作業に必要なデータを取り除くなどのパーソナライズ操作を行いたいといった状況が十分に考えられる。また特定の役割にある作業者のみが参照できるデータを取り扱うような状況を考えることもできる。協調作業の支援をこうした観点から考えると、共有

* 2004 年 2 月 6 日逝去

資料に対して作業に携わる作業員それぞれの役割や立場に沿ったビューを与えることは、業務の効率化だけではなくセキュリティ的な面からも重要な意義を持つといえる。

我々のモデルは作業に関わる人の間で共有されるデータを階層的に定義した作業空間を用いて管理するものである。このモデルで行っている作業を行う人それぞれの役割に応じた資料管理はRBAC(Role Based Access Control)の考え方に基づいたものである。それぞれの役割に基づいて作業空間を定め、それぞれの作業空間に応じて扱う資料を定め、またそれぞれの資料に対して作業員それぞれのアクセス権限を定める。共有資料に対するパーソナライズもこの作業空間を単位として行われる。これにより作業空間にアクセスする作業員それぞれの行う作業に応じたビューを提供することができる。またこのモデルは作業中起こる動的な作業空間の構成変化や資料へのパーソナライズを管理する。

本モデルでは作業員それぞれの役割や立場の間の関連性を階層的に定義した作業空間を用いて表現している。作業空間を木構造に構成し作業空間の関連性をこの構成によって表現している。それぞれの作業空間ごとに行われた共有資料へのパーソナライズの内容は作業空間の階層に沿って管理され、作業員たちの間で共有される。これにより作業を行う人々の間の役割や立場の関連性に応じた段階的な資料の共有を行うことができる。この段階的な資料共有を実現するために、我々はこれまでにXMLに基づいた作業空間の定義を行ってきた。またXMLで記述された共有資料に対して、XSLTを用いた資料に対するビューの生成及びその管理を行うためのモデルやメカニズムの実現を図ってきた[3]。

本稿ではこの階層的な作業空間を用いた共有資料の管理モデルにおけるそれぞれの資料に対するパーソナライズについて述べる。資料に対するパーソナライズ操作に関しては、資料の編集を行う際には利用者が編集を行うためのインタフェースを前もって用意するものが多い。ここではスキーマを利用することでインタフェースの自動作成を行い作業員の負担を軽減し、またパーソナライズ操作と対応する記述の管理を容易にすることを図る。これにより本モデルにおいて資料に対するパーソナライズ操作とその記述を対応付けて管理できる。XMLを用いることにより利用中に新たに追加されたデータに対してもスキーマを元にした動的な処理が可能となる。そこで作業員が資料に対して行うパーソナライズについてその処理に関わるXSL文書中のデータの記述とそのXSL文書を作成する手順について述べる。

本稿は以下のような構成になっている。次節で我々のシステムで導入した資料共有モデルについて述べる。まず本モデルにおける作業空間の考え方について触れ、その階層的構造を用いて作業員の役割に沿ったビューを生成するメカニズムについて述べる。それに続いて作業員が資料に対して行うパーソナライズ操作について触れ、これらのパーソナライズ操作に関わるXSL文書中のデータの記述とXSL文書の作成に関する手順について述べる。最後に今後の課題について触れ、本稿をまとめる。

2. 文書の共有モデル

我々は作業を行う人の役割の違いや関連に応じた資料の共有を行うためのモデルとして階層的な作業空間を用いた資料共有モデルをこれまでに提案し開発してきた。本節ではこのモデルについての説明を行い、またそれらの作業空間を階層化することによって可能になる資料の段階的な共有について説明する。

2.1 背景と要求

複数の人が共有する情報に対するパーソナライズは、ウェブページのリンクのカスタマイズといった表現上の変換や、デバイスに応じたトランスコーディングなどの構造的な変換に分類できる[5][7]。

我々が対象としているパーソナライズは協調作業の中で作業員の役割に応じて内容を変更することである。資料の変更が動的に反映できることや元文書と別にパーソナライズの内容を管理できることは利点となる。こうした点を考慮し、XML文書の資料パーソナライズのための技術として、XMLで書かれた文書を構造的に扱うことができ、またその文書のデータの検索技術としても大きな利用価値のあるXSLTの利用を選択する[9]。

さらに我々は資料の共有を支援する汎用的なモデルという観点から、XMLのデータとしての意義に焦点を絞り、パーソナライズ操作としてデータ内容の追加や削除、変更を対象とする。これによりパーソナライズによっても構造の変更が抑えられ、多段階にわたる変換を行うことができる。

このような段階的なビューの違いを実現する方法としては共有するデータを一つのXML文書に記述し、そのデータのノードごとにアクセスできる人を設定するものが考えられる。我々のモデルではパーソナライズの内容を記述する文書は元のデータとは別に管理する。これにより子孫以外の作業空間への影響を考慮することなく、自由に作業空間独立なパーソナライズを行うことができる。

我々のモデルにおいてはXSL文書の構造やその編集は重要である。作業員が行ったパーソナライズ操作からXSL文書の記述への変換をシステムにより支援することを考えると、資料に対して行った操作と対応する記述の関連のとりやすい書式とすることは、非常に大きな利点となる[4]。ここで用いているデータの扱いについては次節で述べる。

次にXML文書の編集に関するインタフェースについて考える。データの構造によらずデータ入力画面を自動的に構成する例として表形式のものがある[1]。これはデータごとにインタフェースを準備する必要がない利点があるが、データの表示がフラットになるためデータの構造が分かりにくいという欠点もある。一方利用者に分かりやすい編集という観点からは[2]などがある。こちらの形式はXML文書を表示する際に見えるままの形で編集できるため編集が容易であるが、データに対して編集のためのスタイルを事前に用意する必要がある。

本稿ではユーザインタフェースを自動的に作成することでアプリケーションごとにユーザインタフェースを構築する手順の低減を図る。扱うデータのスキーマを利用してスキーマに沿ったツリー構造を表現したインタフェースを自動生成することで、

データの構造を目に見える形で表現しやすく、また新しいデータに対してもすぐに対応できる。

我々は協調作業に参加する人の分散化などを考慮し、多くの人が容易に利用できるように利便性を考え、作業者がウェブブラウザを通して利用するサーバサイド型のシステムとしての構築を考えている。

2.2 作業空間

協調作業では個々に異なる権限を持つ人が集まり作業を行う。同じ一つの資料を例にとってもそのデータをどのように利用するかは異なることがありうる。資料に対するパーソナライズを行う機能はこうした状況への対処法として有効なもののひとつである。資料へのパーソナライズ操作とはデータの追加や削除、変更を指す。ここではそれぞれの作業者に合わせたパーソナライズが施された資料データをビューとする。我々のモデルでは作業空間の構成は作業中に参照されるこれらのビューに基づいたものであり、このモデルにおいては作業者の役割や作業内容の違いといったものはこうした作業空間を通して得られるビューの違いとして表現される。

そうした違いを表現するために用いる作業空間は作業者のアクセス権限を規定したものと定義することができる。作業空間ごとにその作業空間にアクセス可能な作業者名を保持しておく。またその作業空間のなかで参照するつまりその作業空間独自のビューを得ることのできる資料名を管理する。これらのデータを作業空間ごとに持つことによりそれぞれの作業空間独自のビューに対するアクセス管理を行うことができる。

作業空間ではその他に資料に対して行われるパーソナライズの内容及びそのパーソナライズを行う作業者名を管理する。作業者はそれぞれ行う作業の内容に応じて共有されている資料に対してメモを追加することや、不要な情報を除去するなどして資料のビューを変更することで資料をより利用しやすくするといった要求が考えられる。こうした要求に対応するために資料に対するパーソナライズを管理している。また先程述べたように作業空間はそこで得ることのできるビューに対応する形で規定されたものである。そのため、資料に対するパーソナライズを行う権限を持つ作業者は、その作業空間にアクセスする権限とは別に設定する。

作業者が共有資料にアクセスする場合には必ずいずれかの作業空間を通じてアクセスする。これにより作業者はそれぞれの持つ役割に応じた権限に従い、必要なビューを得ることができる。このようにして設定された作業空間を用いることで同じ作業空間を通して得られるビューは、どの作業者にも同じものとなる。これにより同じような作業を行っている人は同じ作業空間にアクセスするという状況を実現でき、資料の共有を行うことで共同して作業を行うことが可能となる。また作業空間単位でアクセス可能な作業者を設定していることによって、特定の間でのみ同じビューを共有するといったことを行うこともできるようになる。ひとつの資料は複数の作業空間から参照することができるため、異なる役割を持つ人たちのグループでも共通して利用することができる。

現在の社会では、協調作業の中でひとりの作業者のもつ役割

はひとつとは限らず、複数の異なる役割を持つこともある。こうした役割をそれぞれの作業者を基準として考えた場合にはネットワーク型の役割モデルを構築する必要があると考えられる。しかし複数の役割を同時に持つような人の場合でも、その人の持つそれぞれの役割をひとつひとつ個別に考えるとそれらの役割は特殊なものではなく、ここで述べたような作業空間モデルで対応することができると考えている。ひとつひとつの役割はなんらかのプロジェクトに属しているといったものである場合や、その人の持つ属性に応じたものである場合などを考えることができる。このような考え方のもとで複数の役割を持つ作業者を考えた場合には、その作業者の行う作業はそのいずれかの役割に関係するものであり、その作業者の持つ権限や作業中に利用する資料もその作業内容に応じて決まると考えることができる。こうしたそれぞれの役割に対応する形で作業空間を構成しておくことでその作業者の持つ役割はそれぞれここで述べたような木構造の作業空間モデルで表現することができる。複数の役割を持つ人はそれぞれの役割に応じて複数の作業空間へのアクセス権限を持っていると考えることで対応することができる。複数の役割を持つ人はそれぞれの作業に応じてその作業を行う際の役割に応じた作業空間にアクセスし必要な権限や資料のビューを得ることで作業を行うことができる。

ここで説明した他にそれぞれの作業空間は作業空間の親子関係に関するデータを持つ。これは我々のモデルにおける階層的な作業空間構成に関するものであり、次節で説明する。また作業空間のこれらのデータを変更できる作業者名を管理する。

こうして定義した作業空間のデータは作業の進行に沿って起こる動的な変化に対応することが求められる。本モデルでは作業空間の状態という概念を考え対応する。ここで状態とは同じ作業空間において規定されたデータが異なる状況を指す。一時的にアクセスを遮断するような事態がそれにあたる。状態の規定や変更は資料へのパーソナライズと同じ手法によって実現できるようにデータ構造やそのメカニズムを考えている。

2.3 パーソナライズとビューの共有

パーソナライズの重要性の一方で協調作業では共通の認識を持って作業することが必要である。作業仲間間で共有されるべきデータの量は、それぞれの作業者の集合によって異なる。例えば一般の作業者とその指導者の間では扱う資料のビューには違いが生じる可能性は高いが同時に行う作業内容の強い関連性も無視できないと考えられる。こうした人たちのアクセスする作業空間においてはそこで行われるパーソナライズの内容にも関連性があるべきである。こうした状況を表現するため作業空間を階層的に構築する。本モデルでは木構造に作業空間を構築し、このような作業空間の構造を利用することによって段階的な資料へのパーソナライズと階層的な資料の共有を表現している。木構造に構成された作業空間には親子関係を考えることができる。前節で述べたように作業空間の構成に関するデータとしてそれぞれの作業空間の親となる作業空間の名前や子となる作業空間の名前がそれぞれの作業空間で管理され、以下に述べるようなビュー生成の過程等で利用される。

本モデルでは作業空間の階層構造に沿ってパーソナライズさ

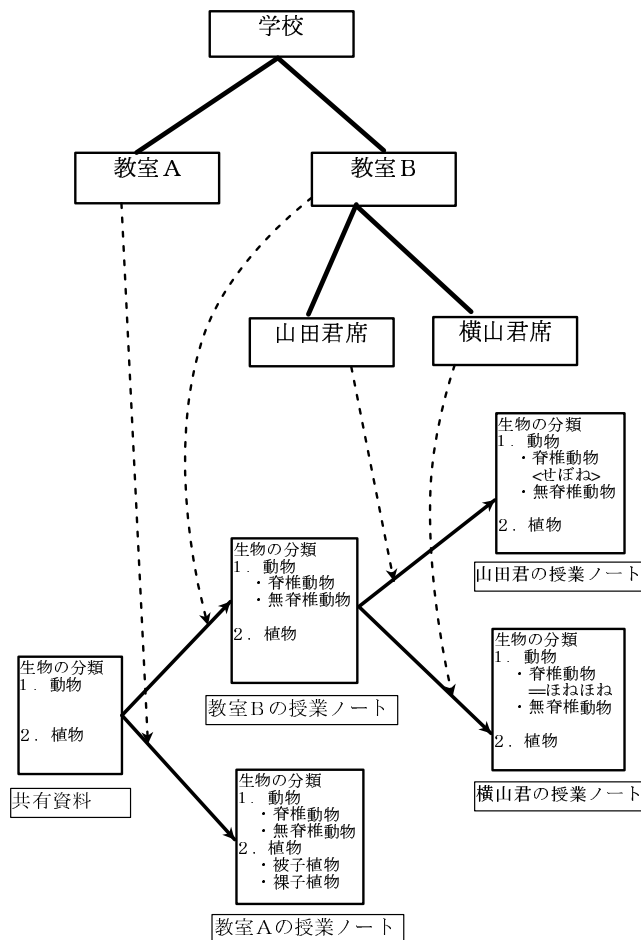


図 1 階層的作業空間とビュー

れたビューの共有を行うメカニズムを規定している [3]。ここではまず前提として共有されている資料は XML で記述された文書とする。この点については、近年多くの分野で XML に準拠したデータ記述言語の制定、利用が広まりつつある状況を考えれば妥当であるといつてよいだろう。それに対するパーソナライズは共有資料である XML 文書の変換として規定し、XSLT を用いてその実現を図っている。それぞれのパーソナライズの内容は XSL 文書の形で記述され、作業空間ごとに管理する。ビューは用意された作業空間の階層構造に沿って、上位の作業空間で施されたパーソナライズから順に共有資料に対して適用することによって生成する。ビューの生成の過程で処理は作業空間の構造に従い分岐する。それによりより近い関係にある作業空間の間ではより共有性の高いビューを得ることができ、作業者の役割の関連性を考慮した段階的なパーソナライズの共有を行うことを可能にしている。

図 1 はこうしたモデルを用いた資料共有の例を示している。この例はある学校における授業で資料の共有モデルを用いた状況を示している。全体で共有されている資料に対して教室 A、教室 B それぞれで教師がパーソナライズを行うことで授業を進めているが、それぞれの教室ごとに授業の進具合が異なっており、教室ごとにパーソナライズされたビューが異なっている状況について示したものである。作業空間の構成は授業資料の共有関係を元に構築したものであり、学校全体で共有されて

いる資料に対して、教室ごとに独自のパーソナライズを行うことができる。それぞれの教室にアクセスできる教師や生徒は教室ごとにパーソナライズされたビューを得ることになる。また山田君と横山君はともに教室 B に属する生徒であるが、教室 B の中にそれぞれ独自の作業空間としての自分たちの席を持っている。そこで教室 B で共有しているビューに対してパーソナライズを加えることで個々人のノートを作成している。上の図はこの状況にあわせて構成された作業空間の構成であり、下の図はこのように構成された作業空間構成におけるひとつの共有資料の例として、授業資料のビューの関連性について示したものであり、それぞれの作業空間で得られるビューが生成される過程を示したものである。破線はそれぞれの作業空間で行われたパーソナライズが影響する変換を示している。この図の状況において教室 B で授業がさらに進行し、教室 B で行われているパーソナライズの内容が変更された場合には教室 B の子作業空間としての山田君席、横山君席で得られるビューもその変更に対応する形で変化する。このようにしてパーソナライズの動的な変更に対応した階層的なパーソナライズ内容の共有の実現を図っている。

3. 資料のパーソナライズ

本節ではここまでで説明したモデルにおいて共有資料に対して行うパーソナライズについてそのデータの表現やシステム内部におけるそのデータの扱いについて述べる。

3.1 操作と構造

作業空間の木構造を利用した階層的、段階的な資料データの共有の XSLT を用いたモデル化についてここまで説明してきた。本節ではその実現にあたりそれぞれの作業空間において作業者が資料に対して行うパーソナライズの操作について述べる。

本システムにおいては XSLT による変換は、データ XML 文書を HTML 文書に変換するような XML 文書の構造を大きく変換することを目的とはしていない。XML の大きな利点のひとつであるデータと表現を分離して表現するという点を有効に考え、共有資料としての XML 文書をあくまでデータを記述したものととして扱い、そこに含まれるデータの追加や除去などの操作を行うものであると考えている。本システムでは木構造に構築した作業空間の構成に従って多段階に XSLT を適用して XML 文書を変換することでビューを生成している。そのためパーソナライズ操作の対象となる XML 文書の構造が大きく変更されることは望ましくない。XML 文書のデータ記述と表現を分離して考えることはこの観点から重要な意味を持つ。XML 文書のデータとしての部分のみを考えると、そのデータ構造は表現まで考える場合と比較して単純な設計が可能である。そのためデータの追加や除去といったパーソナライズ操作で変化する XML 文書の構造もそれほど複雑にならずに済むと考えている。

また資料に対して作業者が行うパーソナライズ操作自体をより簡単に行えるようにする配慮と行われたパーソナライズ操作の内容に応じた XSL 文書への記述の関連性が分かりやすい記述とすることが重要である。このことは操作に対応したユーザインタフェースの生成や、XSL 文書中の記述形式の生成を考え

```

<xsl:template match="*[id='aaa1']">
  <xsl:copy>
    <xsl:apply-templates="node()|@" />
    <xsl:element name="mobile">
      <xsl:attribute name="kind">
        phs
      </xsl:attribute>
      0990-????-2911
    </xsl:element>
    <xsl:element name="faxnum">
      0124-????-23942
    </xsl:element>
  </xsl:copy>
</xsl:template>

```

図 2 XSL 文書の記述の例

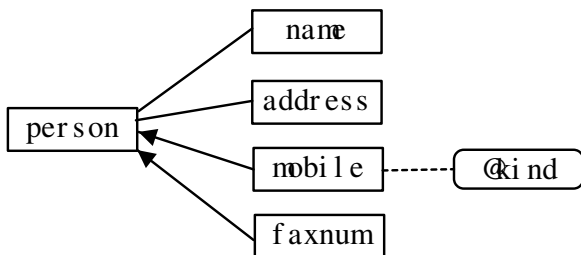


図 3 データ構造の例

るた場合には重要となる。

本システムではこのような点を考慮して XSL 文書の記述形式を以下のような基準で定めた。原則として全てのデータはそのままコピーされるものとする。またパーソナライズのための操作としては XML 文書中の要素の追加、要素の除去及び要素や属性の値の書き換えとする。これらはパーソナライズ対象の XML 文書の構造を大きく変えないために重要である。これらのパーソナライズ操作それぞれに対応する形で、XSL 文書の記述の生成を行う。

要素の追加操作における文書中のどの位置に対して行われるのかという操作位置の指定は、その操作が行われる位置の親ノードを基準として指定するものとする。これはデータの操作を簡潔にするだけでなく、その操作によって生成される要素の他の要素に対する依存性を低くするためのものである。要素の追加を同じ位置にある兄弟要素を基準として指定すると、その要素が除去された場合にパーソナライズで追加されるはずの要素が追加できないといった事態が生じる恐れがある。親要素を基準とした位置の指定はこうした事態を回避するためにも意味がある。これはデータに対する操作を簡潔に記述し、またそのパーソナライズ操作の管理に重要である。要素の除去操作や、値の書き換え操作などに対応する記述の場合には、その位置の指定はそれぞれの操作を行う対象となる要素や属性の位置を指定することで行え、またそれが最も他の要素への影響が少なく済むといえるだろう。

作成される XSL 文書中の記述の例として要素の追加を行うものの一例を図 2 に示す。まずこの記述によって表現されている操作を図 3 を用いて説明しておく。この操作の対象となって

```

<person id="aaa1">
  <name>yamada</name>
  <address>mi e-shi ma-soredoko</address>
  <mobile kind="phs">
    0990-????-2911
  </mobile>
  <faxnum>
    0124-????-23942
  </faxnum>
</person>

```

図 4 データの例

いる要素は属性 id によって指定されているが、ここでは person という要素に id という属性がついているような例であるとする。図 3 はこの要素にはすでに name 要素と address 要素が子要素として存在しており、この要素に対して新たに mobile 要素と faxnum 要素を子要素として追加するようなパーソナライズ操作のイメージを示したものである。図 2 を用いてデータの変換を行った結果の例としては図 4 のような XML の記述が考えられる。

それぞれのパーソナライズ操作に対応した記述として図 2 に示したような xsl:template 要素を生成する。このような xsl:template 要素は処理の対象となる文書中のノードを match 属性に指定する。ここで特に指定されていないノードに対してはそのノードをそのままコピーするための記述を別に用意する。これによって特定の部分のみに対しパーソナライズ処理を行うための記述を生成することができる。この例は match 属性で指定された要素に対して、ふたつの要素をその子要素として追加する操作を行うものである。xsl:copy 要素は操作対象となるノードそのものをそのままコピーする処理を示し、xsl:apply-templates 要素はそのノードの子孫である要素を処理するためのものである。このふたつの XSL の要素を用意することでパーソナライズ操作の対象となるノードをそのままの形で保持することができる。xsl:copy 要素の子要素として指定されているふたつの xsl:element 要素は、パーソナライズ操作によって対象ノードの子要素として追加される要素を生成するものである。この例では kind 要素と faxnum 要素のふたつがパーソナライズ操作の結果として追加されることになる。

要素を除去する操作や値の書き換え操作といった他のパーソナライズ操作に対してもこの例と同じように、xsl:template 要素を生成することによって対処することができる。ノードの除去操作に対応する記述は、除去される要素や属性の位置を match 要素に指定した xsl:template 要素を生成し、その中にそのノードのコピーを行う記述などが置かれていない形の記述となる。要素の中身を書き換える操作の場合には、その対象となる要素ノードそのものではなく、その値を持つテキストノード自体を match 属性に指定した xsl:template 要素を生成し、書き換える値を持つテキストノードをその子要素として持った記述により対応することができる。属性の値を書き換える場合にはその属性を持つ要素を match 属性に記述し、その要素の中身をコピーする前に対応する属性をコピーする記述を生成して、

```

<define name="mobile">
  <element>
    <name>mobile</name>
    <group>
      <attribute>
        <name>kind</name>
        <data type="token" />
      </attribute>
      <data type="token" />
    </group>
  </element>
</define>

```

図 5 スキーマ記述の例

その子要素として書き換える値を持つテキストノードを持つことで記述できる。

ここで述べたようにパーソナライズ操作に対応した形の XSL 文書中の記述は、そのパーソナライズ操作を行う対象となる位置の指定とその操作の種類の指定、そしてその操作によって生成されるノードの構成やその中に持つデータの値といった情報の組み合わせで表現することができ、(operation, place, node, data) のように表すことができる。また上で述べたようにして XSL 文書中の記述の生成を行うと、要素の除去や要素の値の書き換えといった操作はひとつの操作に対してひとつの xsl:template 要素が生成されるが、要素の追加操作に対応する記述はその操作の基準となる位置指定の記述に応じてまとめて記述することになる。今後の説明の都合上、その基準位置が place であるパーソナライズ操作に対応する記述を行っている XSL 文書中の xsl:template 要素を T(place) として表すことにする。

次節ではこうした xsl:template の記述の生成をパーソナライズを行う XML 文書のスキーマの記述を用いて行う場合についてシステムの行う処理の流れについて説明する。しかしこうした記述の生成は XSL 文書中の記述生成に用いるだけでなく、ユーザインタフェースの生成などにも用いることができる。こうしたインタフェースの生成により作業者はデータ入力用の画面を予め生成しておくといった手間を省くことができ、新たなデータを扱うような場面においてもその都度パーソナライズを行う対象となる XML 文書のスキーマの記述からのインタフェースの生成に柔軟に対処できる可能性を持っていると考えている。ここでは RELAX NG の単純化された文法によって記述された XML 文書のスキーマの記述を利用して HTML によるユーザインタフェースを生成した例を挙げておく。図 5 に示した記述は先の例で取り上げた図 4 の mobile 要素に対応するスキーマを RELAX NG の単純化された文法によって記述したものである。この記述を元にしてこの要素の階層的なノードの構成を表現した形でデータの入力を行う HTML の記述を table 表現を用いて生成した例を図 6 に示す。

3.2 処理の流れ

ここまでに本モデルで扱う共有資料に対するパーソナライズの考え方と、そのパーソナライズ操作の内容に対応する XSL 文書中における xsl:template 要素による記述について説明し



図 6 UI の例

た。そこで本節ではそうした XSL 文書への記述についてその記述内容をパーソナライズされる対象の共有資料のスキーマ記述から生成するためのシステムの動作手順について説明する。ここで用いるスキーマの記述については、前節の最後にデータ入力を行うユーザインタフェースの作成についての例を示した際に用いたものと同じく RELAX NG の単純化された文法を用いることにする。

ここではパーソナライズの操作に対応する形となる XSL 文書中の xsl:template 要素の記述をその文書のスキーマの記述を用いて生成する流れを説明する。前節の表記をもちいてパーソナライズの操作を (operation, place, node, data) という形で表したと仮定して以下の説明を行う。この場合には、operation は要素の追加などといった操作の種類を指定するものである。place はパーソナライズの操作に応じて指定された操作の基準となる位置を示したものである。node は要素の追加の際に用いられる情報であり、追加される要素の構造に関する情報を保持するものであり、ここで扱う例の場合には RELAX NG の単純化された文法によるスキーマの記述があたる。data は要素を追加される場合や、値を変更するような場合に用いられる情報であり、その要素中のデータ値を保持する部分である。システムの行う大まかな処理の概要としては、place 部分の表記に応じて xsl:template 要素を生成し operation 部分の表記に応じて node や data の値を用いて要素の生成などを行うものである。

要素の除去操作や要素の値を書き換えるような操作の場合には、(delete, *[id="bbb1"], -, -) のように表記でき、属性値として id を持ちその値が bbb1 である要素を除去する操作である。-の部分はこの操作では意味を成さないことを示している。この場合にはまず T(*[id="bbb1"]) の存在が XSL 文書中にすでに存在するかどうかを調べる。この記述があった場合には、XSL 文書中の xsl:template 要素の子孫要素を全て消去することにより *[id="bbb1"] に相当する要素を除去する記述に書き換えることができる。T(*[id="bbb1"]) がまだ存在していない場合にはまず対応する xsl:template 要素を生成する。要素の除去操作の場合にはその要素が既にパーソナライズ操作の内容に合った記述となっているので処理を終了してよい。

要素を追加する操作の場合には (add, *[id="aaa1"], faxnum, "0124-????-23942") のように表記できる。この操作の場合にはまず T(*[id="aaa1"]) を探す。もし見付からなかった場合

には新たに対応する `xsl:template` 要素を生成しその子要素として `xsl:copy` 要素をさらにその子要素として持つ `xsl:apply-templates` 要素を生成する。次に `faxnum` という記述より対応するスキーマの記述を探しその記述から要素を生成する XSL の命令の記述を生成する。この際のデータ値を持つ部分を生成する際に”0124-????-23942”の記述よりその値を持つノードを生成する。この要素を先程生成した `xsl:copy` 要素の子要素とすることにより対応する XSL 文書の記述を生成することができる。

一方 `T(*[id="aaa1"])` が見つかった場合にはその記述が要素の追加に対応した記述であるかどうかを確認する。他の操作に対応するものであり操作を上書きする場合にはこの記述を破棄し、`T(*[id="aaa1"])` が見付からなかった場合と同様に `xs:template` 要素の生成から処理を行うことで操作に対応する記述を生成することができる。要素を追加する操作は同一の親要素に対して複数の子要素の追加を行うことができるため、`T(*[id="aaa1"])` の記述が要素の記述に対応するものであった場合には、その記述にさらに要素を追加するための記述を付け加えることによって対処することができる。この場合には上に述べた場合と同じようにして `faxnum` 要素を追加するための XSL の命令の記述を生成し、`T(*[id="aaa1"])` に既に存在する `xsl:copy` 要素の子要素として追加することによって要素の追加操作に対応する記述を生成することができる。この例は先に用いた図 3 の例において `faxnum` 要素を追加する状況に対応するものである。ここに述べたような処理の結果として図 2 のような XSL の記述が生成されることになる。

4. ま と め

我々のモデルは協調作業における作業者の役割に応じた資料のパーソナライズを管理することができ、またそれによって得られる資料のビューを作業者相互の役割の関連に基づいて共有することを目的としたものである。現在では多くの分野や多くのアプリケーションで XML で記述されたデータへの対応が広まりつつあり、本モデルでは共有資料の一般性を考慮した本モデルのより汎用的な用途への応用を考え、作業において共有される資料を XML で記述された文書としている。またシステムで扱うこうした共有データ及び、そのビュー生成処理のメカニズムを XML 及び XSLT に基づいて規定している。

本稿では本モデルにおける資料のパーソナライズにおいてそのビュー生成に利用される XSL 文書に着目しその記述及びシステムにおける扱いについて述べた。本モデルにおいて作業者の役割の関連性に基づいた資料の共有を実現するためのメカニズムとして多段階の XSLT による変換を用いたものを規定している。このメカニズムに対応する形でパーソナライズの内容を記述する XSL 文書の記述形式について述べた。これは作業者が行うパーソナライズの操作にそれぞれ対応した形で記述することができる。それぞれの作業空間で行われるパーソナライズ操作は親となる作業空間で得られるビューに対する差分情報であるという点から考えても作業者の行うパーソナライズ操作を主体とした記述は妥当であると考えている。

さらにこうしたパーソナライズに対応する XSL 文書の記述をシステム側で生成する手順についての検討を行った。資料のスキーマ記述を利用することで機械的な XSL 文書の記述の生成を行うものである。これと同じ考え方による作業者がパーソナライズを行う際のユーザインタフェースの構成についても触れた。スキーマに基づいた一連の手順により操作の自動化が容易になり、アプリケーションごとデータごとに事前に作業者が行なわなければならない手間を低減できる。スキーマ記述に定義されていないようなデータ構造を扱うパーソナライズなどまだまだ課題はあるが、作業者の負担軽減のためのひとつの意義ある方向性と考えている。

また本稿ではあまり詳しく触れなかったが、本モデルでは作業空間ごとに管理している作業者名や共有される資料についてのデータといった情報そのものも XML 文書で規定している。作業空間のこれらのデータが動的に変化するような協調作業も考えられるが、ここで述べた共有資料に対するパーソナライズと同様の考え方をういて対応することを考え、XSLT を利用することにより動的な状況変更のある協調作業に柔軟な対応するメカニズムについても検討を行っている。

本モデルは XML で記述されたデータを扱い様々な協調作業に対応するための基盤となるフレームワークの構築を目指したものである。より様々なデータを扱い本モデル上で利用されるアプリケーションの開発や、作業を効率的に行うための作業空間の構成についての社会的な検討などを今後の課題もまた重要な課題といえるだろう。

文 献

- [1] 今村誠, 森口修, 鈴木克志, “XML 入力画面自動生成方式,” FIT(情報科学技術フォーラム)2002, 2002 .
- [2] 矢尾浩, 外山春彦, 白井智, “表示スタイル変更可能な Web トップ XML エディタ,” 情報処理学会研究報告デジタル・ドキュメント 36-3, Vol.116, pp.17-24, 2002
- [3] 村上隆治, 谷垣美沙子, 横田裕介, 上林彌彦, “グループや役割を考慮した XSLT に基づく共有文書のカスタマイズ機構,” 第 13 回データ工学ワークショップ (DEWS), 2002 ,
- [4] T. Koyanagi, K. Ono, and M. Hori, “Demonstrational interface for XSLT stylesheet generation,” Markup Languages: theory & practice, Vol.2, No.2, pp.133-152, 2001
- [5] M. Hori, G. Kondoh, K. Ono, S. Hirose, and S. Singhal, “Annotation-Based Web Content Transcoding,” the 9th International World Wide Web Conference (WWW9), pp.197-211, 2000
- [6] M. A. Schickler, M. S. Mazer, and C. Brooks, “Pan-Browser Support for Annotations and Other Meta-Information on the World Wide Web,” the 5th International World Wide Web Conference (WWW5), 1996
- [7] G. Rossi, D. Schwabe, and R. Guimaraes, “Designing Personalized Web Applications,” the 10th International World Wide Web Conference (WWW10), 2001
- [8] N. C. Romano, Jr., F. Chen, and J. F. Nunamaker, Jr., “Collaborative Project Management Software,” the 35th Annual Hawaii International Conference on System Sciences (HICSS35'02), 2002
- [9] S. Lechner, G. Preuner, and M. Schrefl, “Translating XQuery into XSLT,” ER 2001 Workshops, LNCS 2465, pp.239-252, 2002