

セマンティックグループウェア: RDF を用いた KnowWho の実現

井形 伸之 小櫻 文彦 片山 佳則 津田 宏
{igata, kozac, katayama.yoshin, htsuda}@jp.fujitsu.com

(株)富士通研究所 IT メディア研究所 言語処理研究部
〒211-8588 神奈川県川崎市中原区上小田中 4-1-1

概要: 本稿では、RDF ベースのグループウェア WorkWare++を知識管理の一機能である KnowWho に適用した事例を紹介する。WorkWare++は、人・文書・スケジュールといった日常業務に現れるオブジェクトを、各オブジェクト間の意味的な関係と共に RDF で表現・管理するグループウェアであり、異種アプリケーション情報を統合する働きを持つ。また、WorkWare++によって統合されたデータを利用し、「誰がどんな仕事をしているか」「誰と誰と一緒に仕事をしているか」などを検索する KnowWho 機能を実現している。本稿では、これらの機能を概観すると共に、WorkWare++で用いている Semantic Web 技術について述べる。

1 はじめに

Gartner Research 社の調査では、「業務に必要な情報の 50~75%は人から直接得る」「社内情報の 80%以上は個人 PC 内にあり、その人が辞めたら失われる」といった調査報告[Garter02]がある。この調査報告の主旨は「企業の IT 投資は短期的にはプラスだが、長期的にはマイナス」であり、タイトルにも”Paradox”と銘打たれているが、「個人の生産性向上だけでなく、コラボレーションやイノベーションへの投資の重要性」を指摘している点で参考になる。

企業知識資産の 1 つとして「人(従業員)」がある。特にナレッジワーカーは企業にとって最も重要な「知識」を有している存在であり、これらの資産を管理・活用することが、今後の知識管理(ナレッジマネジメント)でも重要となる。つまり、これまでの知識管理システムでは「文書の共有と検索(文書検索や KnowHow 検索)」が中心であったが、これからの知識管理システムでは「人(の持つ知識)の共有と検索(KnowWho 検索)」が中心であると言える。

従業員の持つ知識を重要な資産と考える企業では、これまでも様々な KnowWho 検索への試みがなされてきた。しかしながら、それらの多くは、従業員の属性情報(資格や業務歴など)を自己申告でデータベースに登録させるものであり、以下の 2 つの問題があった。

1. 人手による登録制のため、内容の充実・鮮度を保つことが難しい(規模が小さい、更新の手間が掛かる)。

2. 所属や資格など単純な属性情報の検索・参照しかできない。また、固定的な形式では表現しにくい人の背景情報(人脈など)が蓄積されない。

これらの問題に対し、我々は、従業員が日常業務で使用するアプリケーションの情報から、従業員の背景情報(スキルや人脈など)を半自動で収集するアプローチを選択し、その枠組として、セマンティックグループウェア:WorkWare++を開発した。以下では、WorkWare++内で用いている Semantic Web 技術[SW]と、WorkWare++によって収集した情報を KnowWho 検索に利用した事例である「ヒューマンナレッジ・ナビゲータ」について述べる。

2 WorkWare++とヒューマンナレッジ・ナビゲータ

2.1 セマンティックグループウェア: WorkWare++

企業内に存在する様々な知識は、一人の従業員によって蓄積されているものもあれば、複数の従業員(組織)によって構築されることもある。また、従業員の持つ知識(スキル)は日々変化するものであり、同時に従業員同士のつながり(人脈)も変化する。このような企業知識の性質にいかに対応できるかが、KnowWho システムのポイントとなる。また、単に、所属や資格といった属性情報だけを扱うのではなく、従業員の様々な側面(作成文書や組織・他の従業員との関係など)も管理することが重要だと考えている。

そこで、我々は、従業員が日常業務で生産する報告書や日報といった業務文書やスケジュール情報など(以下、活動ログと呼ぶ)から、従業員のスキルや人脈といった様々な側面を収集・管理するセマンティックグループウェア: WorkWare++を開発した[Igata03-1, Kozakura03, Katayama03, Igata03-2]. WorkWare++では、活動ログのなかに現れるオブジェクトを OWL で定義し、オブジェクト間の意味的なリンクを RDF によって記述・管理している。

図 1 に、WorkWare++のシステムイメージ図を示す。大きく、ベースアプリ層、メタデータ管理層、マルチビュー層の 3 層から成る。ベースアプリ層は、従業員データベース、情報共有システム(文書管理サーバ)、スケジューラ(スケジュール管理ソフト)などの既存アプリケーションであり、KnowWho 情報の生成のための新たなモジュールを有していない点に注目されたい。つまり、実際の情報は、既存アプリケーションが持ち、WorkWare++では、既存アプリケーションが持つ情報のメタデータのみを持っている。このような構成とすることで、ユーザ(従業員)は、日常業務で使用する既存アプリケーションを変更する必要がなく、KnowWho 検索のための情報の入力・更新のコストを意識する必要がない。

メタデータ管理層では、RDF (Resource Description Framework) [RDF]や OWL (Ontology Web Language) [OWL]といった Semantic Web 技術を用いて、活動ログに現れるオブジェクトの記述・管理を行なっている。ここでは、各既存アプリケーション情報のメタデータを個別に管理するのではなく、一元的な記述形式 (OWL によるボキャブラリ定義)を用いると共に、オブジェクト間の関係を記述・管理している。例えば、従業員オブジェクトと文書オブジェクトの関係として、「文書の作成者(dc:creator に相当)」関係を記述し、従業員データベースと文書データベース間のオブジェクトの関係を管理している。つまり、メタデータ管理層では、異種アプリケーション情報を統合する基盤技術として、RDF や OWL を利用している。これは、異種アプリケーションをシームレスに連携する EAI(Enterprise Application Integration)ツールの一例と見ることができる。

最後にマルチビュー層では、メタデータ管理層によって統合されたメタデータを視覚化し、各既存アプリケーション情報への一元的なアクセス手段を提供する。その 1 つが、次節で紹介する KnowWho 検索機能である。

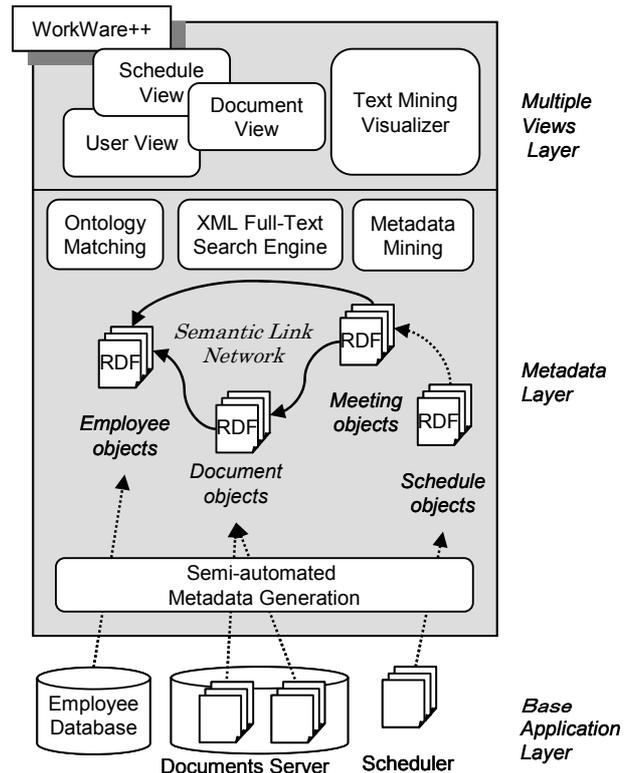


図 1 システムイメージ図

2.2 KnowWho:ヒューマンナレッジ・ナビゲータ

本節では、(株)富士通研究所の従業員約 1,000 名に対して、WorkWare++を適用した事例である「ヒューマンナレッジ・ナビゲータ」[Tsuda03]について述べる。具体的には、従業員データベースである LDAP サーバ、技術報告書などの業務文書を情報共有・承認管理を行なう社内事務処理支援システム、Web 上でスケジュール情報の共有・管理を行なうスケジューラソフトの 3 種のアプリケーション情報を統合し、研究所従業員の KnowWho 検索を実現した。尚、システムの構築に当って、RDF/XML を検索する XML 検索エンジン[Igata00]、オブジェクト間の関係を視覚化する Visual Text Mining 技術[Watanabe02]を利用した。

ヒューマンナレッジ・ナビゲータの利用シーンとして、「営業が顧客先で新技術に関する提案書」を求められた状況を想定している。通常、上記のような要求に対し、営業は、一旦会社に戻り、新技術に詳しい人を何らかの方法で探し出し(例えば、詳しい人を知っていそうな人から紹介してもらい)、探し出した人から提案書の元となる情報を得ようとする。自分の知り合いからすぐに新技術に詳しい人を探し出せた場合はよいが、多くの場合、探し出すだけで数日の時間を要し、ビジネスチャンスを失う場合もある。

これに対し、ヒューマンナレッジ・ナビゲータでは、(1)検索したい技術の特定(技術マップ)、(2)スキルを持つグループの特定(人脈マップ)、(3)個人のスキルの比較(スキル履歴マップ)、の3段階の手順により、新技術に詳しい人の検索作業を支援している。図2～図4に、上記の各段階における出力例を示す。

図2は、技術マップの例である。これは、「個人認証」に関連するキーワードを、組織との関連を含めて表示した例である。技術マップの情報は、文書オブジェクトの中から「個人認証」をキーワードとして持つ文書オブジェクトを検索し、検索結果の文書オブジェクトに含まれる他のキーワードと、文書オブジェクトに関係付けられた従業員オブジェクトの所属組織とを関係付けすることにより作成している。つまり、個人が執筆した文書の情報を元に、個人の所有知識から所属部署の所有知識を構築し、視覚化した例と言える。また、文書オブジェクトや従業員オブジェクトを元に、キーワードや組織間の関連性を計算し、関連性が高いものを近くに配置して表示している。このため、「どの部署で、どんな技術を研究開発しているか」を一目で認識することができる。

図3は、人脈マップの例である。これは、ある技術分野(図3では「個人認証」と「静脈」)に詳しい従業員同士のつながりを表示した例である。人脈マップの情報は、文書オブジェクトを元に技術に詳しい従業員オブジェクトを絞り込み、ミーティングオブジェクトの出席者情報から絞り込んだ従業員同士のつながり(人脈)を計算している。つまり、「よく一緒に同じミーティングに出席している人は知合い度(関連性)が強い」を具現化し、視覚化した例と言える。これにより、「誰が誰とよく仕事をしているか」を一目で認識することができる。

図4は、ある従業員のスキル履歴マップの例である。これは、ある従業員オブジェクト(図4は筆者の中の1人に対応)に関係付けられた文書オブジェクトのキーワードを、文書オブジェクトが持つ日付(文書が作成された日付)により、時系列順に並べて表示した例である。これにより、「この従業員は、これまでどのような仕事をしてきたか」を一目で認識することができる。

尚、探し出した従業員に対して、メールやNetmeetingなどのコミュニケーション手段を用意することで、さらに検索利用者の作業効率を向上させる可能性がある。現在は、sipプロトコルを用いて、従業員のプレゼンス情報を人脈マップ上に表示し、マルチモーダルなコミュニ

ケーションを提供するインスタントメッセージ技術[Okuyama01]と連携している。



図2 「個人認証」に関連した技術マップ



図3 「個人認証」と「静脈」に関連した人脈マップ

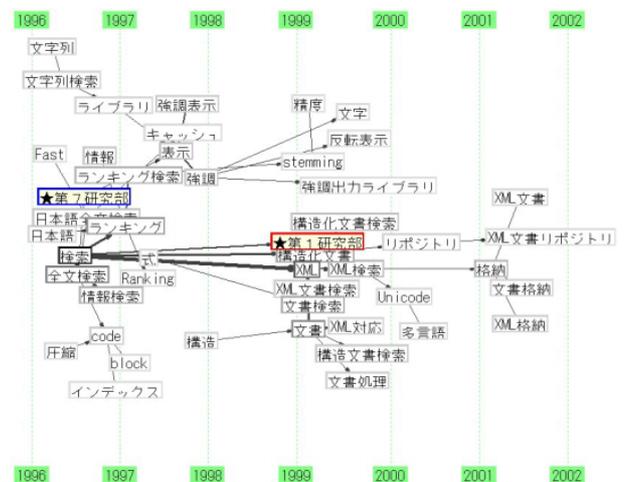


図4 ある従業員のスキル履歴マップ

3 WorkWare++における Semantic Web 技術

本節では、WorkWare++における RDF、OWL の利用例について述べる。筆者らは、前節で述べた WorkWare++と KnowWho は、メタデータ技術におけるシーズとニーズがマッチした事例であり、Semantic Web 技術の有効利用例の 1 つであると考えている。

3.1 シーズ(WorkWare++)とニーズ(KnowWho)の出会い

一般に、メタデータ技術では、「こんなメタデータがあれば、こんなに便利」(効果)、「だからメタデータを書きましょう」(モチベーション)といったスローガンが掲げられるが、「メタデータは誰が書くか」(コスト)の問題を必ず含んでいる。実際、コストに見合うほどの効果を得ることができないものは、モチベーションを維持できなく、「メタデータさえあれば出来るが、メタデータがない」といったデットロックに陥る。これを「メタデータデットロック」と呼んでいる。メタデータデットロックを打破するためには、コストの削減(自動生成や作成支援)、効果の拡大(より多くの利便性)の両面から、モチベーションを維持する必要がある。

WorkWare++では、既存アプリケーションの情報を入力として、メタデータを半自動生成する機能(コスト削減)を持つ。また、RDF や OWL により異種データを統合し、一元的なアクセス手段を提供する機能(効果拡大)を持つ。これは、従来の KnowWho の問題点であった(1)膨大な入力・更新コストが掛かる、(2)単純な属性情報しか検索できない、の 2 点を解決する手段を提供するものであり、WorkWare++の機能(シーズ)と KnowWho の問題点(ニーズ)がうまくマッチした事例であると言える。そして、WorkWare++の機能(シーズ)を支えるメタデータ技術として、RDF や OWL などの Semantic Web 技術がある。以下では、WorkWare++で用いている Semantic Web 技術の詳細について述べる。

3.2 WorkWare++における OWL の利用例

OWL Use Cases [OWL-UseCases]では、第 3 の利用例として、企業内 Web サイトの管理が挙げられている。これは、巨大企業において、異なる部署間の用語の違いを吸収する目的で OWL を利用する例である。WorkWare++における OWL の利用例も、これに類似したものであり、企業内で使用される異なるアプリケーションの用語(スキーマ)の違いを吸収する目的で OWL を利用

している。

企業規模が大きくなればなるほど、一様なアプリケーションを使用することは難しく、例えば、スケジュール管理ソフトを例にした場合でも、部署によって様々なソフトが乱立していることの方が通常であろう。しかしながら、そこで扱われている情報(オブジェクト)の性質には共通部分も多い。

そこで、WorkWare++では、各既存アプリケーションで扱われる情報(オブジェクト)の共通部分を OWL で定義(以下、共通 OWL 定義)すると共に、個々のアプリケーションのスキーマを OWL で定義(以下、個別 OWL 定義)し、共通 OWL 定義と個別 OWL 定義間のマッピングを記述することで、異種アプリケーションの用語(スキーマ)の違いを吸収している。

図 5 に、共通 OWL 定義、個別 OWL 定義、OWL マッピング定義の利用イメージと定義例を示す。ここで、”&kw;”や”kw:”は共通 OWL 定義の名前空間、”&local;”や”local:”は、個別 OWL 定義の名前空間を表している。また、”&v;”や”v:”は vCard(次節参照)の名前空間を表している。

共通 OWL 定義の詳細は次節で述べるが、共通 OWL 定義は 1 つであり、メタデータ管理層(図 1 参照)における RDF のボキャブラリ定義に相当する。

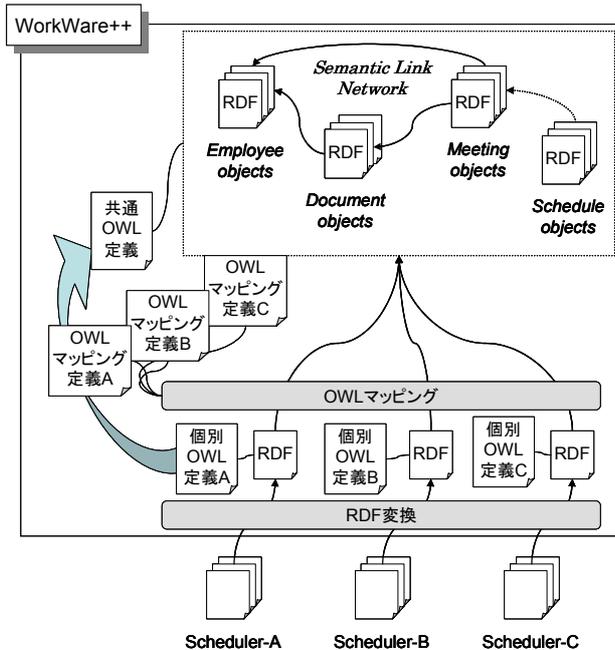
個別 OWL 定義は、個々のアプリケーションの情報を RDF に変換した時の OWL 定義に相当する。RDF 変換の容易性はアプリケーションのデータフォーマットによって異なるが、RDB などの単純なテーブル形式であれば、問題は少ない。また、個別 OWL 定義は、個々のアプリケーションのスキーマを壊す必要はなく、各スキーマを OWL で再定義したものでよい。

OWL マッピング定義は、共通 OWL 定義と個別 OWL 定義をマッピングする定義であり、両方で共通するクラスやプロパティ、およびインスタンスの対応関係を定義している。例えば、図 5 の例では、owl:equivalentClass を用いて、共通 OWL 定義の kw:Person と個別 OWL 定義の local:User が同じ(ここでの「同じ」とは、「定義が同じ」という意味ではなく、「インスタンス集合が同じ」という意味)であること、owl:equivalentProperty を用いて、共通 OWL 定義の v:FN(formatted Name)と個別 OWL 定義の local:uname(user name)が同じであること、また owl:sameAs によって、kw:igata@jp.fujitsu.com と local:igata が同一のインスタンスであることを定義している。

このように、WorkWare++では、共通 OWL 定

義、個別 OWL 定義、OWL マッピング定義の 3 種の OWL 定義を使用して、異種アプリケーションの用語(スキーマ)の違いを吸収している。

現在、これらの定義は手作業で記述しているが、最も記述が困難な OWL マッピング定義に関しては、カテゴリマッチング技術[Hoshiai03]などの利用を検討している。



・共通OWL定義の例

```
<owl:Class rdf:about="&kw;Person"/>
<owl:Class rdf:about="&kw;Employee"/>
<rdfs:subClassOf rdf:resource="&kw;Person"/>
</owl:Class>
<owl:DatatypeProperty rdf:about="&v;FN">
  <rdfs:domain rdf:resource="&kw;Person"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
...
```

・個別OWL定義の例

```
<owl:Class rdf:ID="&local;User"/>
<owl:DatatypeProperty rdf:ID="&local;usname">
  <rdfs:domain rdf:resource="&local;User"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
...
```

・OWLマッピング定義の例

```
[クラスのマッピング]
<owl:Class rdf:about="&local;User">
  <owl:equivalentClass rdf:resource="&kw;Person"/>
</owl:Class>
...
[プロパティのマッピング]
<owl:DatatypeProperty rdf:about="&local;usname">
  <owl:equivalentProperty rdf:resource="&v;FN"/>
</owl:DatatypeProperty>
...
[インスタンスのマッピング]
<local:User rdf:about="&local;igata">
  <owl:sameAs rdf:resource="&kw;igata@jp.fujitsu.com"/>
</local:User>
<local:User rdf:about="&local;htsuda">
  <owl:sameAs rdf:resource="&kw;htsuda@jp.fujitsu.com"/>
</local:User>
...
```

図 5 WorkWare++におけるOWLの利用例

3.3 WorkWare++における共通 OWL 定義

共通 OWL 定義は、異種アプリケーションの情報を統合するものであり、また一元的なアクセス手段を提供するためのものであるため、出来るだけ汎用的であることが望ましい。しかしながら、過度の汎用化は、定義自体の議論もさることながら、実装面での複雑化を招く原因ともなる。よって、より汎用的な定義は、さらに上位のオントロジに任せることとし、今回は KnowWho ドメインにおいて出現するボキャブラリ(従業員、組織、文書、ミーティング、...)を定義することとした。また、KnowWho ドメインにおけるボキャブラリを設計するに当たり、新たなボキャブラリを一から作成するのではなく、既に開発済みのボキャブラリを流用する方針を取った。

Web 上では、開発済みのオントロジをリストアップしたオントロジブローカと呼ばれるサイトが存在する。例えば、DAML Ontology Library [DAML-OntoLib]には、2004 年 3 月時点で、282 個のオントロジ(67,987 個のクラスと 11,149 個のプロパティの定義)がリストアップされ、Person クラスだけでも 52 個の定義がある。この中から、KnowWho ドメインに適用可能なボキャブラリの流用も検討したが、今回は、異種アプリケーションとの import/export 性を重視し、電子情報交換フォーマットとして標準化されている vCard [RFC2426], iCalendar [RFC2445], Dublin Core [DC]のボキャブラリを流用することとした。

vCard は名刺情報、iCalendar はイベント・スケジュール情報の電子交換フォーマットであり、これらを RDF で表現する試みは、共に W3C にて行なわれている [RDF-vCard, RDF-iCal]。Dublin Core は、書誌情報を記述するための 15 の要素を世界規模で同意した定義であり、これを RDF で表現する試みは DCMI(Dublin Core Metadata Initiative)にて行なわれている [RDF-DC]。

WorkWare++の共通 OWL 定義は、上記の RDF 表現の試みの成果をベースとし、上記で不足しているボキャブラリを追加する形とした。

図 6 に、共通 OWL 定義の概略図を示す。橙色のノードがクラス、緑色のノードがプロパティ、青色のノードが XML データ型を表している。また、実線は rdfs:domain または rdfs:range、破線は rdfs:subClassOf 関係を表している。ここで、"&v;"や"&v;"は[RDF-vCard]の名前空間、"&i;"や"&i;"は[RDF-iCal]の名前空間、"&dc;"や"&dc;"は[RDF-DC]の名前空間を表している。また、"&kw;"や"&kw;"は WorkWare++の共通 OWL 定義の名前空間を表している。

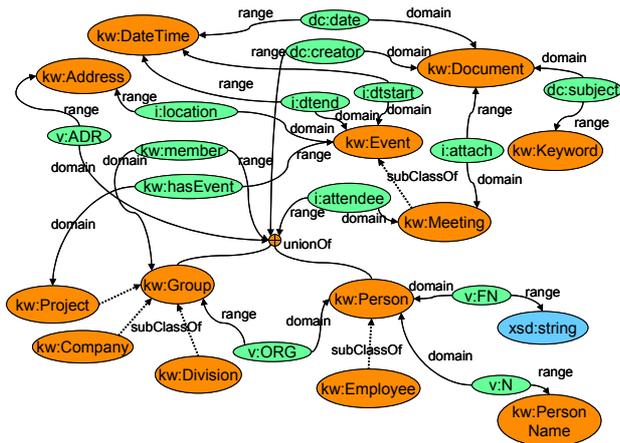


図 6 KnowWho ドメイン OWL 定義の概略図

KnowWho ドメインにおけるボキャブラリの基本クラスとして、kw:Person, kw:Group, kw:Event, kw:Document を定義している。また、それぞれのサブクラスとして、kw:Employee, kw:Project, kw:Meetingなどを定義している。ほとんどのプロパティは、既存仕様のボキャブラリを流用している。特徴としては、「kw:Projectはkw:Eventを持ったkw:Groupのサブクラス」「kw:Documentのdc:subjectのrdfs:rangeはkw:Keywordクラス」などが挙げられる。

尚、付録には、共通 OWL 定義を N3[N3]形式で記述したもの(抜粋)を添付した。

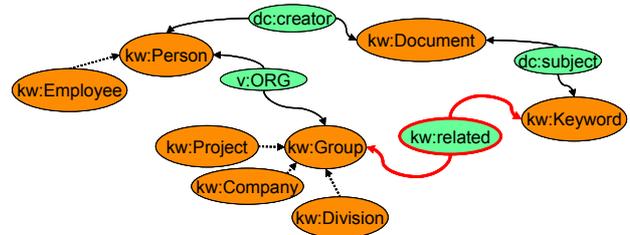
3.4 基本プロパティを元にした拡張プロパティの追加

図 2~4 に示したヒューマンナレッジ・ナビゲータの出力例では、共通 OWL 定義の kw:Group と kw:Keyword の関連性(図 2:技術マップ), kw:Person と kw:Person の関連性(図 3:人脈マップ), kw:Person と kw:Keyword の関連性(図 4:スキル履歴マップ)を計算し、関連性の度合いに応じてノードの配置やエッジの太さを調整して視覚化している。これは、図 6 に示した基本プロパティを元に、関連性という新たな拡張プロパティを追加していることに相当する。

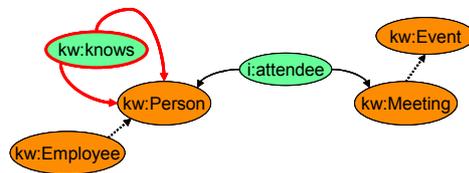
例えば、図 2 の技術マップでは、kw:Group と kw:Keyword の関連性を表示しているが、これは、kw:Keyword から、dc:subject ⇒ (kw:Document) ⇒ dc:creator ⇒ (kw:Person) ⇒ v:ORG を経由し、kw:Group と kw:Keyword 間に kw:related という拡張プロパティを追加した操作に相当する(図 7(1)参照)。また、図 3 の人脈マップでは、kw:Person 同士の関連性を表示しているが、これは、kw:Person から i:attach ⇒ (kw:Meeting) ⇒ i:attach を経由し、kw:Person 同士に kw:knows と

いう拡張プロパティを追加した操作に相当する(図 7(2)参照)。

上記の処理は、正しく Semantic Web レイヤにおける Rule 層による推論を行なっていることに相当し、WorkWare++における Ontology 層より上位の Semantic Web 技術の利用例とである。



(1) kw:Groupとkw:Keyword間のkw:relatedの追加



(2) kw:Person間のkw:knowsの追加
図 7 拡張プロパティの追加

4 関連研究

前節で示した OWL 定義は、他の既存仕様と対立するものではなく、共存できるものであると考えている。vCard, iCalendar に対応した各種スケジュール管理ソフトとは import/export 可能であり、例えば、RDF による iCalendar を実装したスケジューラである Semaview 社の eventSherpa [Sherpa]とは、相互に情報交換可能である。

また、知り合い(kw:knows に相当)を記述する RDF フォーマットとして、blog やソーシャルネットワークングの分野で広まりつつある FOAF (Friend Of A Friend)[FOAF]がある。WorkWare++で扱われるメタデータは、FOAF に変換できるだけの記述内容を持っているため、FOAF への export が可能である。

その他、人の学歴や職歴を記述する XML フォーマットとして、HR-XML(Human Resource XML) [HR-XML]がある。これは、主に人材派遣業における人材情報の管理・情報交換を目的にした仕様であるが、HR-XML との情報交換可能性を検討することによって、企業を越えた人材情報交換が実現できる可能性もある。ただし、HR-XML が XML ベースであることから、URI 付与問題などを含めて、RDF モデルへのマッピング可能性を検討する必要がある。

5 おわりに

本稿では、Semantic Web 技術を用いて異種アプリケーション情報を統合するセマンティックグループウェア: WorkWare++と、知識を持った従業員を検索する KnowWho システム: ヒューマンナレッジ・ナビゲータを紹介した。

WorkWare++の今後の展開として、大きく、(1)OWL 定義の拡張、(2)実装面の検討、(3)インターネットへの展開、の3つの方向がある。

今回示した OWL 定義は、KnowWho ドメインに限った OWL 定義であり、クラス定義も Person (Employee), Group (Project), Event (Meeting), Document など単純なクラスであった。一般に、企業活動で生産される知識は、今回示した OWL 定義だけで記述できるほど単純ではなく、適用範囲を広げる意味でも、OWL 定義の拡張が必要となるだろう。ただし、この場合も、新規のポキャブラリを開発するのではなく、既存のポキャブラリをベースにした拡張が望ましいと考えている。例えば、企業活動におけるサービスやプロセスを記述するビジネスオントロジ [Izumi04]が、既に開発されている。

次に、実装面の検討であるが、現在の WorkWare++では、XML 検索エンジンや Text Mining 専用の描画クライアントソフトを用いているなど、純粋な Semantic Web 技術のみで構成されているわけではない。これに関し、既存 RDF エンジンの利用、または新規開発などがあるが、メタデータの効果拡大をより多く得るためには、Rule 層/Logic 層の規格制定や関連 OWL ツールの充実を待つ必要がある。

最後に、WorkWare++の適用範囲をインターネットまで広げるためには、Trust や URI の問題を解決する必要がある。Trust 層は、Semantic Web レイヤにおける最上位層に位置付けられていることから分かるように、極めて難しい問題である。また、「人」や「もの」など、実世界に存在する対象物への URI 付与問題に関して、さらに議論が必要であろう。

参考文献

[Garner02] "The Knowledge Worker Investment Paradox", Garner research, 2002.
[SW] W3C Semantic Web Activity: "Semantic Web", <http://www.w3.org/2001/sw/>
[Igata03-1] Igata, Tsuda, Watanabe, Matsui: "Semantic Groupware: an approach to KNOW-WHO using RDF", Proceedings of the 6th SANKEN (ISIR) international symposium, pp.77-80, 2003.

[Kozakura03] 小櫻, 片山, 井形, 渡部, 津田: "Semantic Web 技術を利用したグループウェア WorkWare++", 情報処理学会 65 回全国大会, 2003.

[Katayama03] 片山, 小櫻, 井形, 渡部, 津田: "セマンティックグループウェア WorkWare++と KnowWho 検索への応用", 情報学基礎研究会, 2003-FI-71-2, 2003.

[Igata03-2] Igata, Tsuda, Katayama, Kozakura: "Semantic groupware and its application to KnowWho using RDF", ISWC 2003, Posters and Demonstrations, pp.53-54, 2003.

[Tsuda03] 津田: "ヒューマンナレッジ・ナビゲータ", セマンティック Web コンファレンス 2003, INTAP, p.7-1~p.7-5, 2003

[RDF] W3C RDF Working Group: "Resource Description Framework", <http://www.w3.org/RDF>

[OWL] W3C Web-Ontology (WebOnt) Working Group: <http://www.w3.org/2001/sw/WebOnt/>

[Igata00] 井形, 難波: "大規模な構造化文書データベースにおけるインデクシングと検索の手法", 情処情報学基礎研究会, 2000-FI-57-2, 2000.

[Watanabe02] 渡部: "富士通研究所による特許検索・分析支援システム「ACCENT」", INFOSTA2002, A-1, 2002.

[Okuyama01] 奥山, 角田, 岩川: "新しいモバイルサービスを支えるインスタントメッセージ技術", 雑誌 Fujitsu, Vol.52, No.4, pp.262-207, 2001.

<http://magazine.fujitsu.com/vol52-4/paper02.pdf>

[OWL-UseCases] J.Heflin: "OWL Web Ontology Language Use Cases and Requirements", W3C Recommendation 10 February 2004.

<http://www.w3.org/TR/2004/REC-webont-req-20040210/>

[Hoshiai03] Hoshiai, Yamane, Tsuda: "Category-level Retrieval among Heterogeneous Information Sources based on Category Matching", Proceedings of the 6th SANKEN (ISIR) international symposium, pp.73-76, 2003.

[DAML-OntoLib] DAML: <http://www.daml.org/ontologies/>

[RFC2426] F. Dawson, T. Howes: "vCard MIME Directory Profile", 1998. <ftp://ftp.isi.edu/in-notes/rfc2426.txt>

[RFC2445] F.Dawson, D. Stenerson: "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", 1998. <ftp://ftp.isi.edu/in-notes/rfc2445.txt>

[DC] Dublin Core Metadata Initiative: "Dublin Core Metadata Element Set, Version 1.1: Reference Description", 1999.

<http://dublincore.org/documents/1999/07/02/dces/>

[RDF-vCard] R.Innella: "Representing vCard Objects in RDF/XML", W3C Note 22 February 2001, <http://www.w3.org/TR/vcard-rdf>

[RDF-iCal] D.Connolly, L.Miller: "RDF Calendar Workspace", 2001. <http://www.w3.org/2002/12/cal/>

[RDF-DC] D.Beckett, E.Miller, D.Brickley: "Expressing Simple Dublin Core in RDF/XML", 2002.

<http://dublincore.org/documents/dcmes-xml/>

[N3] T.B.Lee: "Primer: Getting into RDF & Semantic Web using N3", <http://www.w3.org/2000/10/swap/Primer.html>

[Sherpa] Semaview: <http://www.eventsherpa.com/>

[FOAF] foaf project: "Friend Of A Friend", <http://www.foaf-project.org/>

[HR-XML] HR-XML Consortium: <http://www.hr-xml.org/>

[Izumi04] 和泉, 吉岡, 山口: "ビジネス指向のサービス関連オントロジー", 人工知能学会誌, Vol.19, No.2, pp.151-158, 2004.

付録 : WorkWare++における共通 OWL 定義 (抜粋)

```

@prefix kw:
<http://kw.example.org/OWL/KnowWho/0.1#>.
@prefix dc:
<http://purl.org/dc/elements/1.1/>.
@prefix dcterms:
<http://purl.org/dc/terms/>.
@prefix i:
<http://www.w3.org/2002/12/cal/ical#>.
@prefix v:
<http://www.w3.org/2001/vcard-rdf/3.0#>.
@prefix owl:
<http://www.w3.org/2002/07/owl#>.
@prefix rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-
ns#>.
@prefix rdfs:
<http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd:
<http://www.w3.org/2001/XMLSchema#>.

kw:Person
  rdf:type owl:Class .
kw:PersonName
  rdf:type owl:Class .
kw:Employee
  rdf:type owl:Class;
  rdfs:subClassOf kw:Person .
v:FN
  rdf:type owl:DatatypeProperty;
  rdfs:domain kw:PersonOrGroup;
  rdfs:range xsd:string .
v:N
  rdf:type owl:ObjectProperty;
  rdfs:domain kw:Person;
  rdfs:range kw:PersonName .
...
v:ORG
  rdf:type owl:ObjectProperty;
  rdfs:domain kw:Person;
  rdfs:range kw:Group .
...
kw:knows
  rdf:type owl:ObjectProperty;
  rdfs:domain kw:Person;
  rdfs:range kw:Person .
kw:related
  rdf:type owl:ObjectProperty;
  rdfs:domain kw:PersonOrGroup;
  rdfs:range kw:Keyword .
...
kw:Group
  rdf:type owl:Class .
kw:Canpany
  rdf:type owl:Class;
  rdfs:subClassOf kw:Group .
kw:Division
  rdf:type owl:Class;
  rdfs:subClassOf kw:Group .
kw:member
  rdf:type owl:ObjectProperty;
  rdf:type owl:TransitiveProperty;
  rdfs:domain kw:Group;
  rdfs:range kw:PersonOrGroup .
kw:leader
  rdf:type owl:ObjectProperty;
  rdfs:subPropertyOf kw:member .
...

kw:Project
  rdf:type owl:Class;
  rdfs:subClassOf kw:Group .
kw:hasEvent
  rdf:type owl:ObjectProperty;
  rdfs:domain kw:Project;
  rdfs:range kw:Event .
kw:subProject
  rdf:type owl:ObjectProperty;
  rdfs:subPropertyOf kw:member;
  rdf:type owl:TransitiveProperty;
  rdfs:domain kw:Project;
  rdfs:range kw:Project .
...
kw:Event
  rdf:type owl:Class .
kw:Meeting
  rdf:type owl:Class;
  rdfs:subClassOf kw:Event .
i:summary
  rdf:type owl:DatatypeProperty;
  rdfs:domain kw:Event;
  rdfs:range xsd:string .
...
i:attendee
  rdf:type owl:ObjectProperty;
  rdfs:domain kw:Meeting;
  rdfs:range kw:PersonOrGroup .
i:organizer
  rdf:type owl:ObjectProperty;
  rdfs:domain kw:Meeting;
  rdfs:range kw:PersonOrGroup .
i:attach rdf:type owl:ObjectProperty;
  rdfs:domain kw:Meeting;
  rdfs:range kw:Document .
...
kw:Document
  rdf:type owl:Class .
kw:TechnicalMemo
  rdf:type owl:Class;
  rdfs:subClassOf kw:Document .
kw:Material
  rdf:type owl:Class;
  rdfs:subClassOf kw:Document .
kw:Keyword
  rdf:type owl:Class .
...
dc:creator
  rdf:type owl:ObjectProperty;
  rdfs:domain kw:Document;
  rdfs:range kw:PersonOrGroup .
dc:subject
  rdf:type owl:ObjectProperty;
  rdfs:domain kw:Document;
  rdfs:range kw:Keyword .
...
kw:PersonOrGroup
  rdf:type owl:Class;
  owl:unionOf kw:Person, kw:Group .
...

```