

ビジネスアプリケーション構築支援ツール (LordBader)

寺井 公一^{*1}, 杉浦 直樹^{*1}, 澤井 雅彦^{*1}, 和泉 憲明^{*2}, 山口 高平^{*1}

^{*1}静岡大学 情報学部

^{*2}産業技術総合研究所サイバーアシスト研究センター

本稿では、多粒度リポジトリを用いたビジネスアプリケーション構築支援環境とそれに基づく支援ツールを提案する。多粒度リポジトリは、粗粒度、中粒度、粗粒度リポジトリから構成され、それぞれ、エンタープライズモデリング、ワークフローモデリング、アプリケーション構築を支援する。本支援環境により、ソフトウェア開発における分析、設計、実装を一貫して支援することができる。本研究ではプロトタイプ支援ツールを JAVA と XML により開発を行った。そして、プロトタイプ支援環境を Web ショップアプリケーション構築に適用したところ、本支援環境の妥当性を確認した。

Layer Organized Repository based Business Application Development Environment (LordBader)

Koichi TERAII^{*1}, Naoki SUGIURA^{*1}, Masahiko SAWAI^{*1}, Noriaki IZUMI^{*2}
and Takahira YAMAGUCHI^{*1}

^{*1}Faculty of Information, Shizuoka University

^{*2}Cyber Assist Research Center, AIST Tokyo Waterfront

This paper proposes an integrated support environment for business application development. This environment is based on multi-layered repository, which consists of coarse-, middle-, and fine-grain-size of repository that correspond to enterprise models, workflow models, and software models, respectively. This environment helps us to develop business applications in incremental deployment of analysis, design, and implementation. We have implemented this environment by JAVA language, and described repository data by XML so that each repository can be interoperated, and evaluated by applying to Web Shop development. As the result, we have confirmed that the business logic of applications can be reflected on correctly with its changes of business models.

1. はじめに

迅速な e-ビジネスの実現のためには、ビジネスアイデアを基に素早くビジネスモデルを構築し、モデルに基づいて正確にビジネスアプリケーションへと展開することが重要である。

そこで、本稿では、ビジネスモデル構築からビジネスアプリケーションへの展開を一貫して支援する環境を提案する。本研究では、ビジネスアプリケーション開発を、エンタープライズモデリング、ワークフローモデリング、アプリケーション構築の 3 工程から構成する。そして、アプリケーション開発に必要な経験やノウハウなどの知識を収集し、各開発工程毎の知識に分類し、それぞれをモデル化する。このモデルに従い、コンテンツをライブラリ化し、各開発工程毎に利用可能とすることで、それぞれ、粗粒度、中粒度、細粒度リポジトリを構築する。

関連研究として、ビジネスに特化したオントロジーやリポジトリは、既にいくつか提案されているが、形式性の問題などから、そのまま利用するのは難しい。例えば、エジンバラ大学のエンタープライズオントロジー [3] は、形式性が高いものの、抽象的な概念のみが定義されているため、具体的なビジネスモデルの構築には不向きである。SCC(Supply-Chain Council) の SCOR[4] は、ビジネスアクティビティが、抽象的なものから具象的なものまで多数定義されているが、製造業に特化したものである。また、MIT の e-Business Process Handbook [5] は、多くの具象プロセスを含むが、形式性に欠ける。

e-Business Process Handbook は形式性には欠けるものの、豊富な数のプロセスを定義し、ビジネスモデル構築にふさわしい抽象度を備えている。そこで、本研究では e-Business Process Handbook の再構築を行う。粗粒度、中粒度リポ

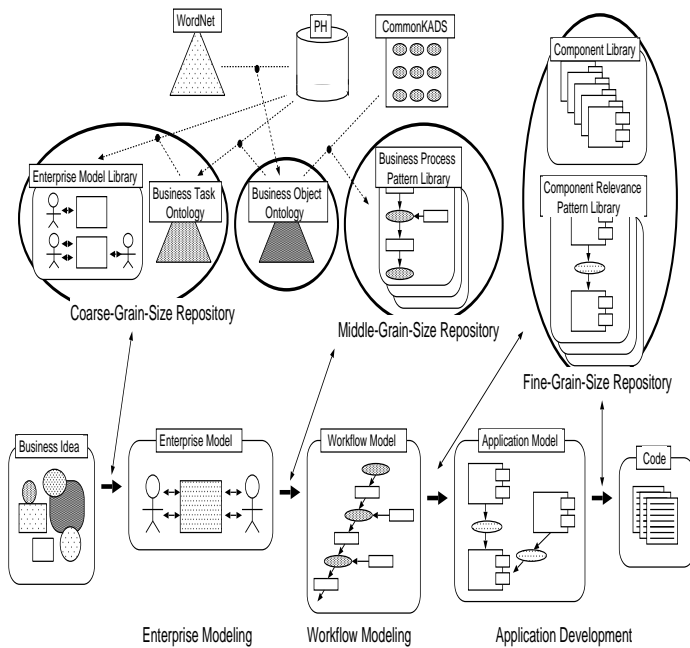


図 1: 多粒度リポジトリの設計



図 2: ビジネスオブジェクトオントロジー (一部)

ジトリは、*e-Business Process Handbook* と、WordNet[6]、CommonKADS[2] などの既存ライブラリを基に構築され、細粒度リポジトリは細利用可能なソフトウェア部品であるビジネスコンポーネント [1] を基に構築される。

本研究では、プロトタイプ支援環境を JAVA により開発した。このツールを用いることにより、エンタープライズモデリング、ワークフローモデリング、アプリケーション構築を直感的な操作で行うことができる。また各リポジトリの内容を XML を用いて記述することにより、各リポジトリ間の相互運用が可能となる。

2. 多粒度リポジトリの設計と構築

2.1 多粒度リポジトリの設計

本研究では、ビジネスアプリケーション開発を、エンタープライズモデリング、ワークフローモデリング、アプリケーション構築の 3 工程から構成し、各々の工程における参照情報として、それぞれ、粗粒度リポジトリ、中粒度リポジトリ、細粒度リポジトリを構築する。ビジネスアプリケーション開発は、これらのリポジトリを利用しながら、段階的にモデルを細かくしてゆくことにより行われる。

各ビジネス間、各開発工程間での語彙の一貫性を保つために、本研究では、ビジネスオブジェクトオントロジーを構築する。そして、そのオントロジーを中核に、各リポジトリを、*e-Business Process Handbook*、WordNet、CommonKADS などの既存のリポジトリをベースに構築する (図 1 参照)。

粗粒度リポジトリは、エンタープライズモデルとビジネスタスクオントロジーから成る。中粒度リポジトリは、ビジネスオブジェクトと、CommonKADS で定義されている推論プリミティブから構成されるビジネスプロセスパターンライブラリから成る。細粒度リポジトリは、再利用可能なソフトウェア部品であるビジネスコンポーネントと、その利用の典型的なパターンから成る。

2.2 ビジネスオブジェクトオントロジーの構築

ビジネスモデルは様々な語彙を用いて記述されるが、個々のビジネスごとに固有の語彙を用いて記述すると、どれが共通概念であり、どれが非共通概念であるかということを判別することができなくなる。そこで、この語彙に関する合意を本研究ではビジネスオブジェクトオントロジーとして、概念定義と概念階層を与える。

本研究では、一般的な語彙の体系であるプリンストン大学により提供されている WordNet をベースとして、ビジネスオブジェクトオントロジーを構築する。まず、主要な名詞概念として、ビジネスプロセスの入出力に使用され出現頻度の高い名詞に注目し、*e-ビジネスプロセスハンドブック* から名詞概念を抽出する。こうして得られた主要名詞概念は Attribute, Person or Organization, Resource, Product, Service, Information, Act から成る。次に、これらの主要名詞概念をオントロジー中の最上位に配置し、他の名詞概念を WordNet 中での位置を参照しながら下位に配置する。

ここで、WordNet は一般的な語彙の体系であり、ビジネスドメインに対して適切でない概念階層を持つことがある。そこで、このような概念に対しては、それぞれ適切な位置へ配置する必要がある。

構築したビジネスオブジェクトオントロジーの一部を図 2 に示す。

2.3 粗粒度リポジトリの構築

2.3.1 エンタープライズモデルライブラリの構築

ビジネスアプリケーション開発を前提とするビジネスアイデアの想起段階では、ビジネス定義のための詳細な概念が用いられることはあまりない。その一方で、過去に利用したエンタープライズモデルや流通しているベストプラクティスから新しいアイデアを想起することが多い。そこで、まず、米国ビジネスモデル特許を含むような典型的なエンタープライズモデルを対象に、MIT にて提供されているケーススタ

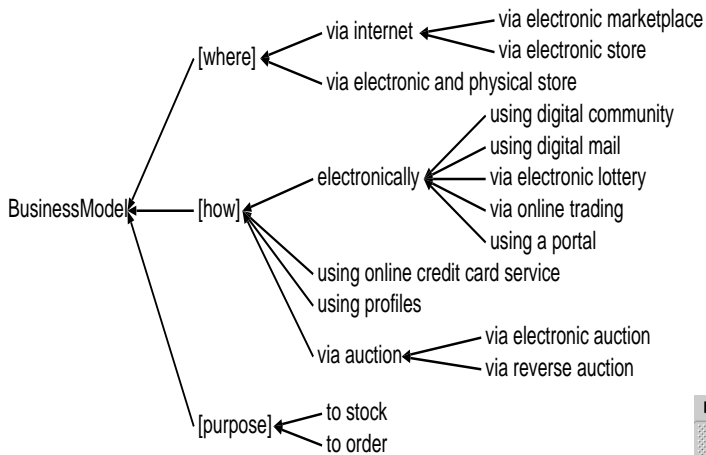


図 3: ビジネスタスクの特徴分類

ディを分析することにより、エンタープライズモデルを再利用するためのライブラリを構築する。

本研究で構築するエンタープライズモデルは、ビジネス中に登場するアクタと、その間の関係によって表される。そこで、エンタープライズモデルのケーススタディから、顧客 (customer) やサプライヤ (supplier) などのアクタを抽出する。そして、それらアクタ間のインタラクションのタイプにより、エンタープライズモデルを Distributor, Broker, Creator, Extractor, Service Provider の 5 つのタイプに分類する。さらに、分類ごとに含まれるタスクの集計を行い、使用頻度の高いタスクからデフォルトエンタープライズモデルを構築し、そのエンタープライズモデルとの差分となるビジネスタスクを分析することにより、タスクの特徴分類階層を構築した (図 3 参照)。

2.3.2 ビジネスタスクオントロジーの構築

エンタープライズモデルには、そのビジネスが持つタスク階層が含まれており、本研究では、各ビジネスが持つタスクをその階層中に明示化することにより、ビジネス間の差異を表す。そこで、ビジネスタスク間の関係を明確にするために、ビジネスタスクの概念階層が必要となる。e-Business Process Handbook の階層構造は、8 つの基本タスクを上位として、数多くのビジネスタスクが定義されている。しかし、階層化の観点が本研究の目的と異なることから、e-ビジネスプロセスハンドブックをそのまま用いることは難しい。そこで、本研究では e-Business Process Handbook を再構築することにより、ビジネスタスクリポジトリを構築する。

具体的には、e-Business Process Handbook の全ケーススタディからビジネスタスクを抽出し、ビジネスタスクごとの利用頻度を算出する。そして、利用頻度の高いビジネスタスクを概念階層の上位に配置し、それらの下位にケーススタディで用いられている他のビジネスタスクを配置する。さらにその下位に、ケーススタディでは用いられていないが e-Business Process Handbook に登場するビジネスタスクを配置する。

構築したビジネスタスクオントロジーは約 500 のタスク定義を含んでおり、その概要を図 4 に示す。

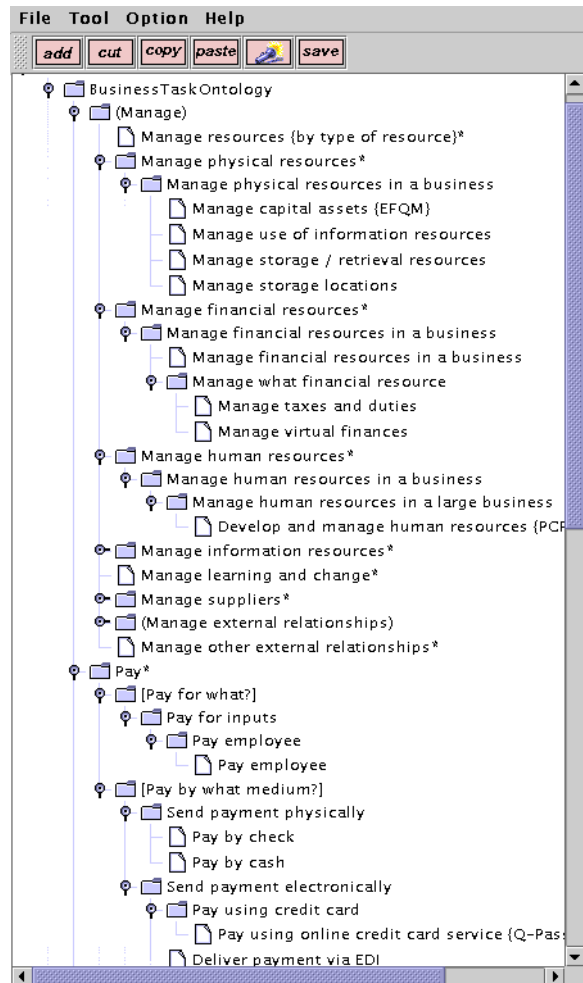


図 4: ビジネスタスクオントロジー

Deliver product or service

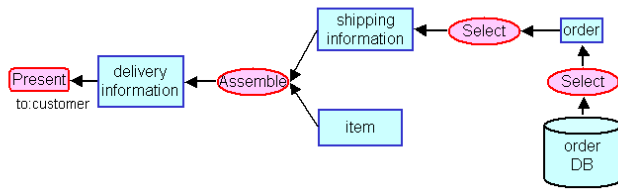


図 5: ビジネスプロセスパターンの例

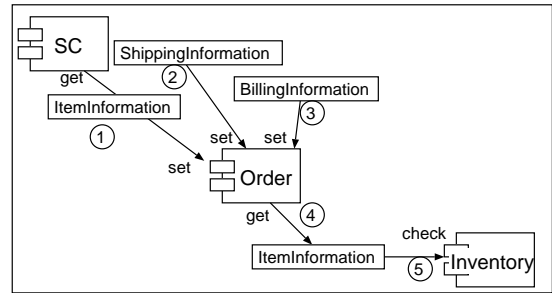


図 6: コンポーネント関連パターンの例

2.4 中粒度リポジトリの構築

粗粒度におけるビジネスの特徴を反映したビジネスアプリケーションを開発するためには、特徴とするビジネスタスクを、企業固有の業務手順を明示化し、下位工程に反映するための記述である中粒度のワークフローモデルへと展開する必要がある。そこで、展開時の参照情報となる中粒度リポジトリを構築する。

本研究では、CommonKADS をベースとして推論プリミティブを用意し、これをビジネスプロセスプリミティブと呼ぶ。CommonKADS は知識システム構築方法論であり、その推論プリミティブを用いてワークフローモデルを記述することにより、知識を背景としたモデリングを効率よく行うことが出来る。推論プリミティブに対して、その入出力オブジェクトを、ビジネスオブジェクトオントロジー中の名詞概念を用いて設定することにより、ビジネスタスクからの展開パターンを定義する。これをビジネスプロセスパターンと呼び、中粒度リポジトリに蓄える(図 5 参照)。

2.5 細粒度リポジトリの構築

明示化されたワークフローをビジネスアプリケーションへと展開するための参照情報である細粒度リポジトリを構築するために、本研究では、再利用可能なソフトウェア部品であるビジネスコンポーネントを利用する。コンポーネントを利用することにより、ワークフローモデルでは記述することのできない、ビジネスアプリケーション固有のシステムレベルの処理(トランザクション管理、データベースへのアクセス)を隠蔽することができる。本研究では、ケーススタディとして Web ショップを取り上げ、Web ショップを構築するための 7 種類のコンポーネントのロール(Order, Shopping Cart, Inventory, Customer, Delivery, Catalog, Payment)を抽出した。各々のコンポーネントは、コード化時に利用されるメタデータとして、JAVA の API 情報を XML 形式で保持している。本研究では、これらのコンポーネント間の典型的な関連パターン(図 6 参照)をコンポーネント関連パターンとして定義する。このパターン中には、各々のコンポーネントがどんな順序でどのように関わることが記述されている。細粒度リポジトリでは、このコンポーネント関連パターンをライブラリとして蓄え、ワークフローモデルとのマッピングを与えることによりビジネスアプリケーションを構築する。

3. プロトタイプ支援環境 (LordBader)

3.1 支援環境の設計と構築

以上より、ビジネスアプリケーション構築を支援するための多粒度リポジトリの構築を行うことができた。粗粒度リポ

ジトリはビジネスタスクの HAS-A の階層であるビジネスタスクオントロジーと、そのタスクオントロジーをベースに構築されているエンタープライズモデルライブラリから成り、エンタープライズモデリングを支援する。中粒度リポジトリは、ビジネスオブジェクトの IS-A の階層であるビジネスオブジェクトオントロジーと、ビジネスオブジェクトに対する操作である推論プリミティブから構成されるビジネスプロセスパターンから成り、ワークフローモデリングを支援する。細粒度リポジトリは、細利用可能なソフトウェア部品であるビジネスコンポーネントと、その間の典型的なパターンであるコンポーネント関連パターンから成り、アプリケーション構築を支援することができる。

本研究では、これら 3 つのリポジトリを相互運用しながらビジネスアプリケーションを構築するための、プロトタイプ支援環境(LordBader: Layer Organized Repository baseD Business Application Development EnviRonment)を JAVA と XML を利用して構築した。構築したプロトタイプは、ビジネスアプリケーション開発における 3 つの工程をそれぞれ支援するエンタープライズモデルエディタ、ビジネスプロセスエディタ、コンポーネント関連パターンエディタから成る(図 7, 8, 9 参照)。プロトタイプ支援環境の概要を図 10 に示す。

プロトタイプ支援環境では、リポジトリのコンテンツ、及び各工程で扱われるモデルはすべて XML により記述されている。そして、それらは基本的に、アクタ、ビジネスオブジェクト、コンポーネントのような“オブジェクト”と、それらの間の“関係”から構成されている。これにより、各モデルに対して、DOM や SAX などを用いて統一的にアクセスすることが可能であり、各ツールは各々のリポジトリを相互に運用することができる。また、その基本的なモデルのために、各ツールにおいて、モデルのビューは非常に理解しやすくなっている。さらに、モデルの編集はオブジェクトのクリック、ドラッグといった、直観的な方法で行うことができる。

3.2 プロトタイプ支援環境によるアプリケーション構築

3.2.1 エンタープライズモデルの構築

エンタープライズモデルエディタではエンタープライズモデルを構築する。

エンタープライズモデルは最も抽象化されたモデルであり、各ビジネスの核となる特徴を明確にする。本研究においてエンタープライズモデルは、ビジネス中で実行されるタスクの階層構造により表され、各々のビジネスの特徴は、そのタスク階層がどのようなタスクで構成されているかにより表される。本支援環境でのエンタープライズモデリングの目的

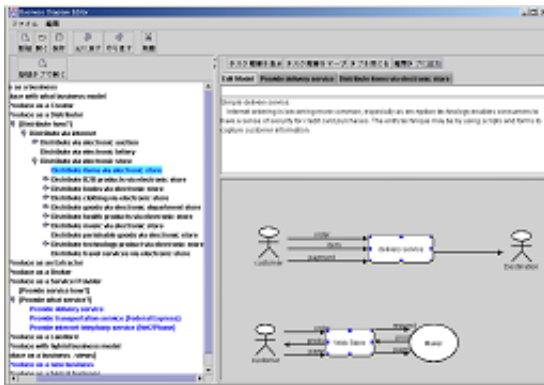


図 7: エンタープライズモデルエディタ

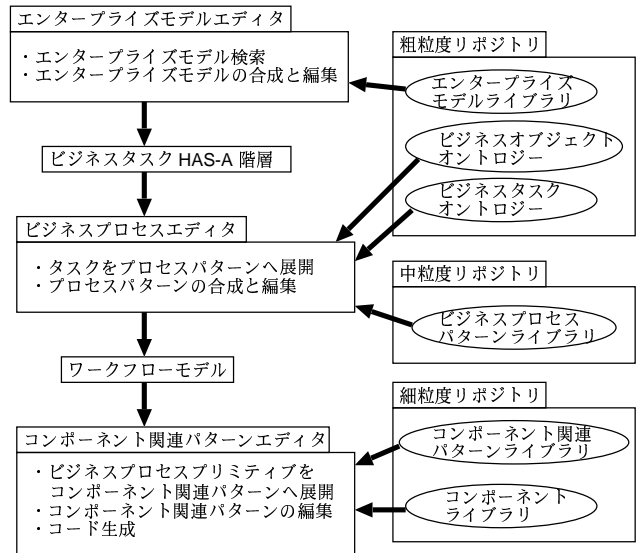


図 10: プロトタイプ支援環境の概要

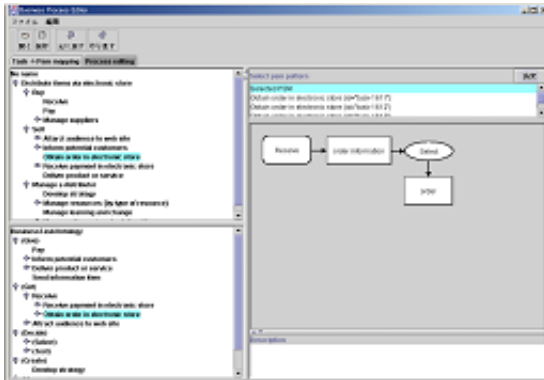


図 8: ビジネスプロセスエディタ

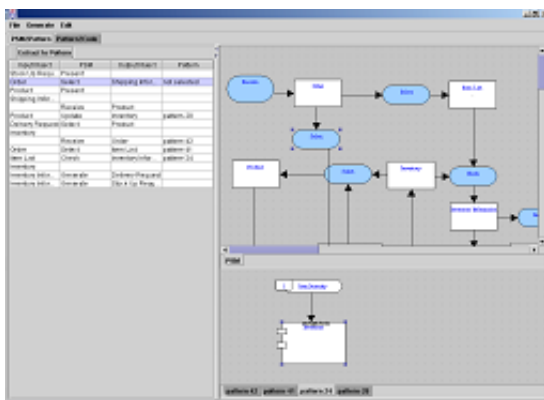


図 9: コンポーネント関連パターンエディタ

は、新たなエンタープライズモデルであるタスク階層を構築することである。

新たなエンタープライズモデルを構築するために、我々はエンタープライズモデルライブラリを用いる。エンタープライズモデルライブラリには、過去のビジネスの成功事例が豊富に含まれており、その各々のビジネスの特徴（タスク階層）を組み合わせることで新たなビジネスアイデアを生み出すことができる。

エンタープライズモデルエディタでは、まず、ユーザはエンタープライズモデルライブラリを参照し、既存のエンタープライズモデルから、これから作成する新規エンタープライズモデルのベースとするものを選択する。ベースとするモデルを複数選択した場合、それらの合成作業を行う。その際、エンタープライズモデル同士を比較し、各モデルの特徴となるアクタやビジネスタスクを採用し、重複部分や削除すべきビジネスタスクは取捨選択する。特に、新たなビジネスタスクを追加する際は、ビジネスタスクオントロジーを参照して、適切なものを選択する。こうして構築されたエンタープライズモデルを既存のモデルと比較・分析することにより、ビジネスの特徴を精錬することが可能である。

3.2.2 ワークフローモデルの構築

ワークフローモデルエディタでは、エンタープライズモデルエディタの利用により得られるエンタープライズモデルからワークフローモデルを構築する。

エンタープライズモデル中のビジネスタスクは“何を”行うかを示す抽象的な概念であり、“どう”行うかという情報は含んでいない。そこで、あるタスクに対して、それを実現するワークフローは、利用できる資源や企業固有のノウハウといった観点から、いく通りも存在する。ワークフローモデリングでは、この方法を各々のビジネスに適した形でワークフローモデルとして一意に定めることを目的とする。

ワークフローモデルは、ワークフローの断片であるビジネスプロセスパターンから構築される。本支援環境では、“リポジトリを利用してアプリケーションを構築”、“自らの経験・知識を基にリポジトリのコンテンツを管理する”という

2つの役割を想定しており、エンタープライズモデルとビジネスプロセスパターンとのマッピングは後者によりあらかじめ与えられていると想定している。

ビジネスプロセスエディタでは、まず、エンタープライズモデルエディタの利用により得られたエンタープライズモデルから、ビジネスプロセスパターンを抽出する。次に、同定されたビジネスプロセスパターン群を合成し、一連のワークフローモデルを構成する。その際、各パターンが入出力として持つオブジェクトに着目し、ビジネスオブジェクトオントロジー中での位置情報を参照しながら、距離の近い概念同士を結びつける。このビジネス全体のワークフローモデルにより、ビジネス全体の流れを把握し、ボトルネックとなる個所を診断したり、重複した作業を一本化したりする、といったことが可能となる。これを、中粒度リポジトリによるワークフローモデルとする。

3.2.3 アプリケーションの構築

コンポーネント関連パターンエディタでは、ビジネスプロセスエディタから出力されるワークフローモデルからビジネスアプリケーションを構築する。そのために、ワークフローモデル中の各々の推論プリミティブから、コンポーネント関連パターンを抽出する。ワークフローモデルとコンポーネント関連パターンとのマッピングは、ワークフローモデルの構築時と同じようにあらかじめ与えられていると想定している。抽出されたパターンは、パターン間の共通オブジェクトを結合することにより、アプリケーション単位へとまとめられる。まとめられたパターンは、パターン中のメタデータを参照することにより各々コードへと展開される。

4. 評価と検討

今回、アプリケーション構築の工程ではコンポーネントの同定を主に行っており、生成されるコードはコンポーネントのメソッドが並んでいるにすぎない(図 11 参照)。しかし、プロトタイプ支援環境を Web ショップ構築に適用したところ、ビジネスアイデアの変化が、生成されるコードのメソッドの構成や並びに適切に反映されたことから、妥当性は確認することができた。

より詳細な処理を実現するコードを生成するための今後の課題としては、モデルの表現能力の強化と、アプリケーションモデルの3層分離化が上げられる。現在、ワークフローモデルと、コンポーネントを用いたアプリケーションモデルの間には、埋める必要のあるギャップが存在する。ワークフローモデルは、ノウハウや過去の経験を活かした仕事の流れを表すことができるが、ビジネスアプリケーション構築に必要な細かいルールを記述することができない。例えば、注文を受け取り、在庫をチェックし、注文を確定するということは表現できるが、“在庫数が5以下であれば、~を行う”ということは表現できない。また、より効果的なビジネスアプリケーションを構築するには、アプリケーションを、プレゼンテーション、ビジネスロジック、データマネジメントという3つの層を独立して考える必要があると思われる。

1つ目の課題の解決策として、BPMP(Business Process Management Initiative)により策定が行われているBPML(Business Process Modeling Language)[7]を利用して、ルールをワークフローモデル中に埋め込むことを検討中である。BPMLではXMLシンタックスにより、ビジネス上の細かいルールを表現することができる。2つ目の課題に関しては、それぞれの層を表すモデルと、アプリケーション

```
NewPatternActionForm newPatternActionForm = (NewPatternActionForm) form;

newPatternActionForm.getEmail();
// Got Email
newPatternActionForm.getBillingInformation_Telephone();
newPatternActionForm.getBillingInformation_Address_StreetAddr();
newPatternActionForm.getBillingInformation_Address_State();
newPatternActionForm.getBillingInformation_Address_Country();
newPatternActionForm.getBillingInformation_Address_ZipCode();
newPatternActionForm.getBillingInformation_CreditCard_CardExpiryDate();
newPatternActionForm.getBillingInformation_CreditCard_CardNum();
newPatternActionForm.getBillingInformation_CreditCard_CardType();
// Got BillingInformation
newPatternActionForm.getShippingInformation_Telephone();
newPatternActionForm.getShippingInformation_Address_StreetAddr();
newPatternActionForm.getShippingInformation_Address_State();
newPatternActionForm.getShippingInformation_Address_Country();
newPatternActionForm.getShippingInformation_Address_ZipCode();
// Got ShippingInformation

// create Order

// order: getLinItem
order.getLinItem();

// linItem: getItemQuantity
linItem.getItemQuantity();

//inventory: checkInventory
```

図 11: 生成されるコードの例 (一部)

構築時に、3つの層をアプリケーションへと統合するための方法を検討中である。

5. まとめ

本稿では、多粒度リポジトリを用いて、ビジネスアイデアの構築からアプリケーションへの展開を一貫して支援する支援環境を提案した。今回、プロトタイプ支援環境を構築し、Web ショップアプリケーション構築に適用したところ、本支援環境の妥当性を確認した。今後は、より効率的、効果的なビジネスアプリケーション構築を目指し、本支援環境を拡張させていく予定である。

参考文献

- [1] Herzum, Peter, and Oliver Sims: “Business Component Factory”. John Wiley & Sons, 1999.
- [2] Schreiber, Guus, et al: “KNOWLEDGE ENGINEERING AND MANAGEMENT The CommonKADS Methodology”.
- [3] Ushold, U., et al: “The Enterprise Ontology, Knowledge Engineering Review”, Vol.13, Special Issue on Putting Ontologies to Use(1998).
- [4] See <http://www.supply-chain.org/>
- [5] See <http://ccs.mit.edu/ph/>
- [6] See <http://www.cogsci.princeton.edu/~wn/>
- [7] See <http://www.bmpi.org/>