修士論文

表構造の解釈に基づくオントロジーの獲得

指導教官　　石田 亨 教授

京都大学大学院情報学研究科
修士課程社会情報学専攻

田仲 正弘

平成17年8月1日

表構造の解釈に基づくオントロジーの獲得

田仲 正弘

内容梗概

現在の Web のコンテンツは主に人間が読むためのものであり，基本的に機械が自動的に収集して処理することは想定されていない．そのため，例えば Web ページを検索する場合にも主にキーワードに基づく検索が用いられている．そこで次世代の Web として，セマンティック Web の構想が唱えられている．セマンティック Web では，Web 上のリソースのクラスや関係を定義するオントロジーを用いることで，意味に基づいて Web サイトを検索できるとされている．

セマンティック Web の実現には，オントロジーの蓄積が必要となるが，一般的な Web のユーザにとってセマンティック Web のためのオントロジーを記述するのは困難であり，そのため現時点ではセマンティック Web はそれほどの広がりを見せていない．今後現在の Web のように多くのユーザによってセマンティック Web のためのコンテンツが蓄積されていくには，HTML のような人間が作成しやすい構造を持つコンテンツから，そのコンテンツの持つ情報構造を利用してオントロジーを自動的に生成するという方法が考えられる．

本研究では，表形式のデータに注目してオントロジーを獲得する手法を提案する．表は作成が容易であることや，直感的に理解しやすいことなどの理由により，多くのデータをまとめて示す際に非常に広く用いられている．Web においてもオンラインショップのカタログ等に広く用いられており，表構造の特徴を利用して自動的にオントロジーが獲得できれば，その有用性は大きい．

表ではその構造により，表中のセルに記述されたデータ間の関係が表現されている．しかし広く Web か ら表を集められた表を対象としてオントロジーの獲得を行うには，以下の問題 がある．

表構造の多義性　表ではその構造によって記述されたデータ間の関係が表現されるが，同じ表構造でも表によって表す関係は異なる．ある構造をどう解釈するのが適当かは，表中に記述されたデータの意味によって決まるため，データの字句的な特徴や表の認知モデルなどに基づくアプローチでは詳細な構造の解釈が難しい．

コンテンツの多様性　ドメインに特化した知識ベースを用いて表中に記述されたデータの意味を理解することにより，より詳細な表の解析をおこなう手

法が提案されている．しかし，Web から広く収集された表には，多様なドメインのコンテンツが含まれることが多いため，知識ベースを前もって作成しておく方法は大きなコストがかかる．

以上の点の解決のため，本研究では以下の手順からなるアプローチを提案する．

1. 構造の解釈を与える
2. 解釈を与えた構造を一般化する
3. 表全体を解釈する

表ごとに特定の表構造がどのような関係を表現しているかということを人間が解釈して与えることにより，それぞれの表にあわせて構造を解釈できる．本研究では，表中に記述された一部のデータ間の関係を記述して与えることで，そのデータが記述された部分の構造の解釈を与える．また，解釈が与えられた構造を，その構造が表す意味が変わらないように一般化することにより，より多くのデータ間の関係が獲得できる．一般化された表構造が表中で一致する箇所を探すことで，データ間の関係を獲得できる．本研究では，セルの隣接関係や繰り返し構造に注目して表構造の形式化・一般化を行う．

本研究の貢献は以下の通りである．

表に応じた構造の解釈　表ごとに表中の一部のデータ間の関係を人手で記述して与えることにより，データの意味まで考慮した，それぞれの表に応じた構造の解釈が可能である．これにより，従来の研究の手法では得られなかった，そのまま RDF/OWL の記述として利用できるより詳細な関係が得られる．

様々なドメインへの適用　本研究で提案する手法では，人手で与えられる構造の解釈に従って表中のデータ間の関係を得る．しかし，構造の解釈を与えるコストは小さく，また表中のデータの意味を理解するための知識ベースを必要としないために，様々なドメインに容易に適用することができる．

従来の手法と比較すると，それぞれの表について人手で解釈を与えるため，完全に自動的な処理ができないが，RDF/OWL でのオントロジー記述に必要とされる詳細な関係を得ることができる．また，表中に多くの情報が繰り返し構造で記述されている場合には，表構造の一般化によって，その情報の RDF/OWL による記述を少ないコストで得られるため，セマンティック Web のためのコンテンツ蓄積のために有用である．

Ontology Extraction Based on Interpretation of Table Structures

Masahiro TANAKA

Abstract

The most of documents on the current Web are provided for humans, but not for agents. Agents cannot understand web pages described in HTML and keyword search is used for information retrieval on the Web To solve this problem, the concept of Semantic Web is coming up. Semantic Web enables agents to understand the meanings of contents of web pages and make it possible to search for web pages based on meanings. However, Semantic Web has not been spread yet because of the high cost of metadata creation. Therefore it is useful to translate existing data into ontologies using the structures of the data.

We propose a method to extract ontologies from tables using table structures. Tables are widely used because they are easy to construct and understand. A large amount of tables such as price lists and timetables are available on the Web. So it is profitable to extract ontologies from tables using the structures of the tables.

Tables contains various structures. The structures of tables of Relational Database are quite simple, but some tables have complex structures consisting of cells which vary in size. Table structures represent relations between data in the tables However, extraction of ontologies from tables on the Web have problems as follows:

Ambiguity of table structures The relation between data which a table structure represents depends on each table. The interpretations of table structures are determined by the semantics of data. It is the reason approaches based on lexical features of data or a table cognition model extract detailed relations.

Diversity of contents Domain-specific knowledge bases enables us to understand meanings of data in tables in order to extract detailed relations. However, tables available on the Web often contain diverse range of contents. It is the reason approaches using knowledge bases are not effective.

To solve these problems, we propose a method which consists of three steps

shown below:

1. Give an interpretation of a table structure
2. Generalize the table structure which is given an interpretation
3. Extract relations from the whole table

Giving an interpretation to each table by humans enables us to interpret the table structure in each table differently. We give relations between some data in tables as interpretations of the structures which contain the data. The structures which are given interpretations are generalized in order to extract relations from the whole table. On the basis of the observation of table structures, we generalize the table structures, which represent particular relations, focusing on the adjacency of cells and iterative structures.

The contributions of this work are as follows:

Extracting Relations Based on Given Interpretation Our method extracts relations using interpretations of table structures given by humans in order to interpret table structures differently in each table. This make it possible to extract detailed relations which are available as descriptions in RDF/OWL.

Applicability to Tables in Various Domains Our method is easy to apply to tables in various domains because it uses interpretations given by humans and instead of a domain-specific knowledge base for understanding meanings of data in tables.

We applied our method to many kinds of tables in order to show its usefulness and to clear up possible issues. As a result of our experiments, it was clarified that our method can mainly extract relations between a property and a property value and relations between a class and a instance. Additionally we confirmed that our method can extract detailed relations using interpretations given by humans compared with the automatic approach based on prior knowledge such as table cognition model and features of types and location of data. This shows that our method is suited to extraction of various relations which are needed for descriptions in RDF/OWL such as property-property value pairs, class hierarchies and property hierarchies.

表構造の解釈に基づくオントロジーの獲得

目次

第1章　はじめに

現在のWebのコンテンツは主に人間が読むためのものであり，基本的に機械が収集して自動的に処理することは想定されていない．そのため，例えばWebページを検索する場合にも，主にキーワードに基づく検索が用いられている．そこで次世代のWebとして，セマンティックWebの構想が唱えられている．セマンティックWebでは，Web上のリソースのクラスや関係を定義するオントロジーを用いることで，意味に基づいてWebサイトを検索できるとされている．

セマンティックWebを実現するには，オントロジーの蓄積が必要となる．オントロジーの記述のために，RDF/OWLなどの記述言語が標準化されているが，これらを用いたオントロジーの記述は非常に煩雑である．また，オントロジーにおける論理的に厳密な関係の定義は人間の直感と一致しないことも多く，一般的なWebのユーザにとってはセマンティックWebのためのオントロジーを記述するのは困難である．そのため現時点ではセマンティックWebはそれほどの広がりを見せていない．

現在のWebの爆発的な広がりは，Webコンテンツを記述するのに主に用いられるHTMLのような言語が，ある程度機械が解釈するために必要な構造を持っていながらも，人間が自然に作成できるものであったことが大きい．このことからも，セマンティックWebを実現させるためには，人間にとってコンテンツ作成が容易になり，専門家以外の多くのユーザの手によって次第にセマンティックWebのためのコンテンツが蓄積されていくことが不可欠であると考えられる．

これまでオントロジー構築を容易にするため，メタデータの入力支援・管理ツールが提案されてきている[1, 2, 3, 4]．これらのツールは，直感的なインターフェースにより記述の煩雑さを軽減し，また入力したオントロジーの論理的な整合性をチェックするなどの支援を行う．しかしこれらは基本的に，専門的な知識を持つ者が論理的に厳密なデータを作成するのを助けることを目的とするものである．

多くのユーザによってセマンティックWebのためのコンテンツが蓄積されていくようにするためには，HTMLのような人間が作成しやすい構造を持つコンテンツから，そのコンテンツの持つ情報構造を利用してオントロジーを自動的に生成するという方法が考えられる．例えばHTMLによって記述されたWeb

ページでは，タグによって見出しや文章の区切りなど，ある程度の構造が表現されている．そのため，タグによるWebページの構造に注目して，Webページからの自動的なメタデータ作成を行う研究が行われてきた．用いる技術として，タグの出現パターンを利用するもの [5]，ラッパを利用して情報抽出を行う方法 [6, 7] などがある．また，ブートストラップ的手法を用いて，多義語の曖昧さを解決しながら非常に大量のコンテンツに対して自動的にアノテーションを行う手法も報告されている [8].

本研究では，表形式のデータから，表の構造に注目してオントロジーを獲得する手法を提案する．表は作成が容易であることや，直感的に理解しやすいことなどの理由により，多くのデータをまとめて示す際に非常に広く用いられている．Webにおいてもオンラインショップのカタログ等に広く用いられており，表構造の特徴を利用して自動的にオントロジーが獲得できれば，その有用性は大きい．

表ではその構造により，表中のセルに記述されたデータ間の関係が表現されている．表は関係データベースの表のように単純な構造をしたものから，異なる大きさのセルが複雑に配置されたものまでさまざまである．

しかし広くWebから表を集められた表から，その構造を利用してオントロジーの獲得を行うには，以下の問題がある．

表構造の多義性　表ではその構造によって記述されたデータ間の関係が表現されるが，同じ表構造でも表によって表す関係は異なる．ある構造をどう解釈するのが適当かは，表中に記述されたデータの意味によって決まるため，データの字句的な特徴や表の認知モデルなどに基づくアプローチでは詳細な構造の解釈が難しい．

コンテンツの多様性　ドメインに特化した知識ベースを用いて表中に記述されたデータの意味を理解することにより，より詳細な表の解析をおこなう手法が提案されている．しかし，Webから広く収集された表には，多様なドメインのコンテンツが含まれることが多いため，知識ベースを前もって作成しておく方法は大きなコストがかかる．

表からの情報抽出を目的とする従来の研究では，表を解釈するために，与えられた表を典型的な表構造のいずれかに帰着するもの [9, 10] や，表の認知モデル [11] に基づいて表を解釈するもの [12] などがある．これらの方法は，表の構造と表中のデータの字句的な特徴に基づいて関係を獲得し，表中に記述された

データの意味内容は考慮しない．例えば，表の属性と属性値の対応を得ることはできるが，その属性と属性値がプロパティとプロパティ値の関係を表すのか，それぞれ別のプロパティのプロパティ値となるのかを区別することが出来ない．しかし，RDF/OWL のようなセマンティック Web のためのオントロジー記述においては，属性と属性値の対応関係を得るだけでは不十分であり，クラス-インスタンス関係や，プロパティ-プロパティ値などの関係を得る必要がある．

一方表からより詳細な関係を獲得するために，対象ドメインの知識ベースを用いる手法も提案されている [13, 14]．これらの手法では，属性と属性値の対応にとどまらず，プロパティとプロパティ値の関係や概念の包含関係など，より詳細な関係を得ることができるが，表の解析のためにドメインに特化した知識ベースを用意しておく必要があるため，幅広いドメインへの適用が難しい．

そこで，本研究では以下の手順からなるアプローチを提案する．

1. 構造の解釈を与える
2. 解釈を与えた構造を一般化する
3. 表全体を解釈する

ある表構造がどのような関係を表すかは，その表構造だけでなく記述されたデータの意味によるため，知識ベースを用いずに自動的に行うことは困難である．そこで，表ごとに特定の表構造がどのような関係を表現しているかということを人間が与えることにより，それぞれの表に応じた構造の解釈を行う．人手による構造の解釈を用いるため，知識ベースが不要であり，多くのドメインに容易に適用できる．

人間による解釈を反映させるため，表中に記述された一部のデータ間の関係を記述して与え，そのデータが記述された部分の構造の解釈とする．また，解釈が与えられた構造をその構造が表す意味が変わらないように一般化することにより，より多くのデータ間の関係が獲得できる．一般化された表構造が表中で一致する箇所を探すことで，データ間の関係の獲得を行う．本研究では，セルの隣接関係や繰り返し構造に注目して表構造の形式化・一般化を行う．

表の認知モデルやデータの字句的特徴に基づいて表を解析する手法と比較すると，それぞれの表について人手で解釈を与えるため，完全に自動的に表を処理することは出来ないが，プロパティとプロパティ値など，RDF/OWL でのオントロジー記述に必要とされる詳細な関係を得ることができる．また，表中に多くの情報が繰り返し構造で記述されている場合には，表構造の一般化によっ

て，その情報の RDF/OWL による記述を少ないコストで得られるため，セマンティック Web のためのコンテンツ蓄積のために有用である．

一方，知識ベースを用いて解析する手法と比較すると，人手で表構造の部分的解釈を与えることは知識ベースを用意するよりもコストが小さく，さまざまなドメインのコンテンツを含む表を柔軟に処理できる．

以降では，第 2 章で表構造の観察について述べる．次に第 3 章で，本研究で用いる表構造の形式的表現について述べる．第 4 章では，表からのデータ間の関係の獲得の処理について説明する．第 5 章で，提案手法を従来の研究で提案された手法と比較することで有用性を確認する．第 6 章で，表のモデルや表からの情報獲得を扱った従来の研究について述べる．第 7 章で，提案手法を実装したシステムについて説明する．最後に第 8 章で結論を述べる．

第2章　表構造の観察

本研究では，表構造が表すセル中のデータ間の関係に注目し，表からのオントロジーの獲得を試みる．そのためには，表構造が表すデータ間の関係がどのような特徴によって決定されるかに基づいて，表構造の形式的表現を定義する必要がある．形式的に表現された表構造に構造が表す関係を対応付け，その構造に一致する箇所を表中から探せば，表中のデータ間の関係を得ることができる．また，ある表構造が表すデータ間の関係が明らかである場合に，表す関係を変えることなくその表構造を一般化することができれば，より多くの箇所から新たに表中のデータ間の関係を獲得することができると考えられる．

本章では，表構造を一般化し，形式的表現を定義するため，表構造で表される関係と表構造の特徴についての観察を述べる．

2.1　表構造のセマンティクス

表では，その構造によってセル中のデータ間の関係（クラス-インスタンス関係・クラス階層関係・プロパティ-プロパティ値の組）が表される．表構造の観察の結果，多くの場合一つの表の中では同じ構造が表す関係は一定であることがわかった．ここでは表構造が表すデータ間の関係を，その表構造のセマンティクスと呼ぶ．

例として表1を取り上げる．表1はコンピュータ部品の価格表である．一行目の “Processor” は，この表に記述されている商品の分類を示している．また2行目の “ProductID”, “ProductName”, “Price” は，この表に記述されている商品の属性である．3-5行目は，この表に記述された個々の商品の情報に対応する．1行が一つの製品を表し，各列は2行目に記述された属性に対応する属性値を表している．

この表の情報をオントロジーとして記述することを考える．このとき，各商品を一つの個体とみなすことができる．“Processor” という商品の分類は，個体が属するクラスであると考えられる．また2行目は個体の持つプロパティであり，3-5行目の各列にはそれぞれの個体のプロパティ値が記述されていると考えられる．

ここで，表1において “Pentium 4 2.80A GHz” がある個体の “ProductName” というプロパティの値であること，“Processor” がその個体の属するクラスであ

表 1: PC 部品の価格表

Processor		
ProductID	ProductName	Price
P4_340	Pentium 4 3.40E GHz	$260
P4_280	Pentium 4 2.80A GHz	$140
A64_320	Athlon 64 3200+	$160

ることが既知であるとする．このとき表 1 の構造と，記述された情報の種類の対応について観察すると，“Processor” というクラスの記述されたセルの特徴として，幅が広く表の上端に位置していることが挙げられる．また “ProductName” と “Pentium 4 2.80AGHz” というプロパティとプロパティ値の記述されたセルを含む部分の特徴として，“ProductName” が表の二行目におかれて上の辺で幅の広いセルに隣接しており，“Pentium 4 2.80A GHz” は対応するプロパティ “ProductName” と同じ列の下の部分におかれていることが挙げられる．

以上を一般化すると，表 1 の表構造のセマンティクスを以下のように書くことができる．

- 他のセルに比べて幅の広い，表の上端のセルに，表に記載されている個体の属するクラスが記述される．
- 上の辺で幅の広いセルに隣接するセルに，個体のプロパティが記述される．
- プロパティが記述されたのと同じ列で，それよりも下に位置するセルにプロパティ値が記述される．プロパティ値が記述されるセルは全て同じ形状で，縦に連続して配置される．

このような表構造のセマンティクスが得られれば，同じ特徴を持つ構造の部分に含まれるデータの関係を知ることができる．すなわち，“Pentium 4 2.80A GHz” だけではなく，3 行目に記述された “Pentium 4 3.40E GHz” や，5 行目に記述された “Athlon 64 3200+” も “Processor” クラスに属する個体のプロパティ “ProductName” のプロパティ値であることがわかる．

2.2 表構造の仮定

ある表構造のセマンティクスが明らかなら，表中でその表構造が出てくる部分からは，その部分に含まれるデータについての関係が得られる．

本研究では表構造のセマンティクスを得るために，表中の特定の構造を持つ部分に記述されたデータ間の関係の例を人手によって記述することでその構造の解釈を与える．さらに解釈を与えられた構造と一致する箇所を表中から探すことで，表に記述されているデータ間の関係を新たに得ることができる．ただし表中のより多くの箇所からセルのデータ間の関係を得るためには，表構造をその特徴に基づいて一般化する必要がある．2.1 節で述べた表 1 のセマンティクスの記述では，まず “Processor” と記述されたセルの位置と隣接するセルに比べて幅が広いという点に注目している．次に，“ProductName” というプロパティとなる語が上辺では他の辺と比べてより幅の広いセルに隣接しているという点に注目している．さらに，プロパティ値となる語がプロパティとなる語と同じ列の下の部分に，おなじ特徴を持つ連続したセルとして記述されているという点に注目している．

本研究では，表構造の観察の結果から，以下の点に注目し，表構造の記述に関する仮定に基づいて一般化された表構造の形式的表現を定義する．

セルの隣接　表 1 のように，表には複数の行や列にまたがるセルが含まれることがある．観察の結果，隣接する 2 つのセルで幅が異なる場合，それらのセルには異なる種類のデータが含まれることが多いことがわかった．表 1 において，一行目のセルと 2 行目のセルにはそれぞれクラスとプロパティが記述されているが，これらのセルは互いに幅が異なっている．そこで，行や列の構造を表現する際，その行や列に含まれるセルとその周囲のセルとの幅の大小関係に注目するべきであると考えられる．

同じ行や列内のセルの関係　観察の結果，プロパティ名とそのプロパティの値など，関係する 2 つのセルは，ふつう同じ行や列にあることがわかった．表 1 において，2 行目のセルに記述されたプロパティ “ProductName” と，そのプロパティに対応するプロパティ値 “Pentium 4 3.40E GHz”，“Pentium 4 2.80A GHz” などは同じ列に位置している．そのため，同じ行や列内のセルの関係は，その行や列の構造と行や列内でのそれらのセルの位置によって表されると考えられる．また，同じ行や列内で同じ特徴を持つセル（もしくは複数のセルが集まったブロック）が連続して出現する場合には，それらのセル（ブロック）は出現回数によらず似た種類のデータを表していると考えられる．

異なる行や列にあるセルの関係　表中で異なる行や列にある 2 つのセルが互い

に関連を持つ場合，それらのセルはその両方のセルと同じ行や列にあるセルによって関連付けられる．よって互いに関連を持つ 3 つ以上のセルが，異なる同じ行や列に位置する場合は，それらの関係は，同じ行や列にある 2 つのセルの関係を表す構造を組み合わせた構造で表現できる．行と列の構造を組み合わせて表現した際に，その行や列と隣接して同じ特徴を持つ行や列が続く場合がある．この場合も，連続して現れる同じ特徴を持つ行や列は，その出現回数に依らず同じ種類のデータが記述されていると考えられる．

以上の観察に基づいて，第 3 章でデータ間の関係を表す一般化された表構造の形式的表現を定義する．

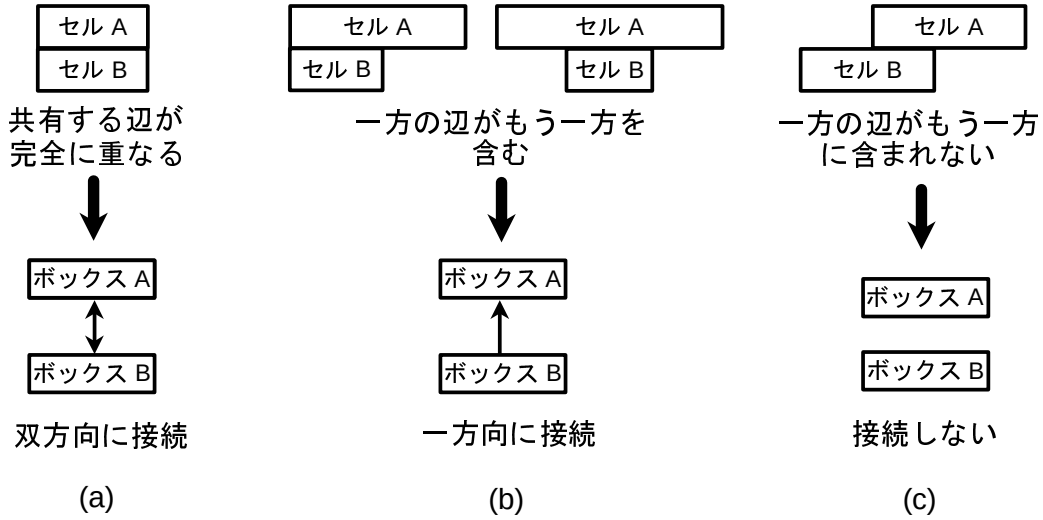

図 1: セルの隣接関係と対応するボックスの接続

第 3 章　表構造の形式化

表構造のセマンティクスを用いて表全体からデータ間の関係を獲得するには，表構造のセマンティクスを決定する特徴に注目した，表構造の形式的な表現が必要になる．本研究の手法では，表中の一部の構造の解釈を与えることで，どの部分にどのような情報が記述されているかを表構造と対応付ける．さらに，同じ情報が記述されていると考えられる部分をまとめて表現することで，表構造の一般化を行う．そのため，2.2 節で述べた表構造に関する仮定に基づき，セルの隣接関係や同じ特徴を持つセルによる繰り返し構造に注目して，表中のセルに記述されたデータ間の関係を表す表構造の形式的表現を定義する．

3.1　セルの隣接

表の一部の領域を表現するために，まず行や列をセルの配列とみなし，セルの隣接関係に注目して表現することを考える．

はじめに，表の一つのセルに相当するボックスと呼ぶ要素を定義する．さらにボックスに対応するセルの周辺のセルとの辺の重なり方によって，セルの隣接関係を一方向または双方向のボックス間の接続としてあらわす．セルの隣接関係とボックスの接続の対応を図 1 に示す．

図 1 の上部は元の表の隣接する 2 つのセルを示している．図 1 の下部はそれらのセルに対応するボックスとそれらの接続を表している．図の下部の四角は

ボックスを表し，上下に並んだ二つのボックス間の接続は，対応するセルの隣接関係を表現している．

図 1(a) のように，元の表においてセル A とセル B の二つのセルが隣接しており，重なっている辺が完全に一致している場合には，セル A とセル B に対応する二つのボックス (ボックス A とボックス B) を双方向に接続して表す．図 1(b) のように，セル A とセル B の二つのセルが隣接しており，重なっている一方の辺がもう一方の辺を含む場合には，短い辺を持つセルに相当するボックス (ボックス B) から長い辺を持つセルに対応するボックス (ボックス A) に一方向に接続して表す．図 1(c) のように，セル A とセル B の二つのセルが隣接しており，重なっている辺の一方が，もう一方に含まれない場合には，それらのセルに相当するボックスは接続しない．また，表において隣接する 2 つのセルのどちらが上（左）でどちらが下（右）かということは重要であるため，接続の上下左右の向きも区別する．

従来の研究では，表の解析の際には，複数の行や列にまたがる幅の異なるセルは分割してから処理を行うことが多い．これは表全体の中でのセルの位置や，セル中のデータの字句的な特徴に基づいて表を解析するためである．本研究では，人手によってあるセルにどのようなデータが記述されているかが与えられたとき，そのセルに隣接する領域のどこまでが同じ種類のデータが記述されたものかをセルの幅の違いによって決定するため，以上のようなセルの幅の違いの表現を用いている．

3.2 同じ行や列のセルの関係

2.2 節で，同じ行や列に含まれる 2 つのセルに記述されたデータが何らかの関係を持つとき，その関係はそれらのセルを含む行や列の構造で表現されると仮定した．そこで，3.1 節で定義したボックスとボックスの接続によるセルの隣接関係の表現を用いて，行や列の構造を表現する．

例として表 1 の構造を考える．2.1 節で述べたように，表 1 において，“Processor”, “ProductName” はそれぞれ，表に記述されている個体のクラスとプロパティであると解釈することができる．また，“Pentium 4 2.80A GHz” は，2 行目に記述された個体のプロパティ “ProductName” のプロパティ値であると解釈できる．そこで “Processor”, “ProductName”, “Pentium 4 2.80A GHz” の間の関係を表す構造として，表 2 に網掛けして示したこれらのセルを含む領域に

表 2: 同じ列内のセルの関係を表す領域

Processor		
ProductID	ProductName	Price
P4_340	Pentium 4 3.40E GHz	$260
P4_280	Pentium 4 2.80A GHz	$140
A64_320	Athlon 64 3200+	$160

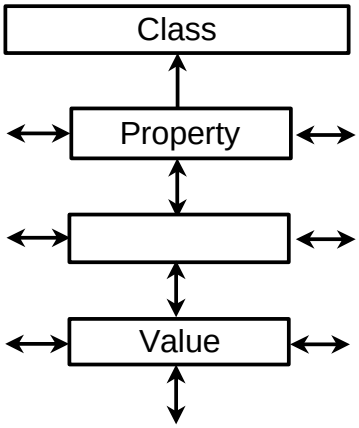

図 2: 表 2 で網掛けした部分の構造

注目する．

このとき，表 2 に網掛けして示した領域を，それぞれのセルに対応するボックスとその接続で表現すると，図 2 のようになる．

“Processor”, “ProductName”, “Pentium 4 2.80A GHz” のセルについては，これらの語の間の関係が与えられているので，これらの語が記述されたセルの相当するボックスには，それぞれの語の意味を表すラベルをつける．図 2 において “Processor” のセルに相当するボックスには “Class” というラベルがつけられる．同様に，“ProductName”, “Pentium 4 2.80A GHz” に相当するボックスには “Property”, “Value” というラベルがつけられる．“Class” とラベルの付けられたボックスは，表 2 の “Processor” と記述されたセルに相当する．“Processor” と記述されたセルは上・左・右の辺に隣接するセルを持たないので，対応するボックスの上・左・右に矢印で表される接続を持っていない．一方下の辺では “ProductName” と記述されたセルに隣接している．“ProductName” と記述されたセルの上の辺は “Processor” と記述されたセルの下の辺に含まれているた

め，“ProductName” のセルに相当する “Property” とラベルの付けられたボックスから，“Processor” のセルに相当する “Class” とラベルの付けられたボックスへ一方向に接続されている．“ProductName” のセルとその下の “Pentium 4 3.40E GHz” のセルは重なっている辺の長さが同じであるため，それらのセルに相当するボックスは双方向に接続される．“Pentium 4 3.40E GHz” のセルと “Pentium 4 2.80A GHz” のセルについても同様に，それらのセルに相当するボックスは双方向に接続される．“Processor” のセルは個体のクラスに相当するので，また図 2 に示したように，各ボックスでは，取り出した列に含まれないセルとの隣接関係も表現する．

さらに，2.2 節で述べたように，同じ行や列内で周囲のセルとの隣接関係が同じセルが連続する場合には，出現回数によらずそれらのセルには似た種類のデータが記述されていると考えられる．そこで，行（列）で連続して出現するボックスが同じ接続を持ち，かつそれぞれのボックスで隣接する行（列）のボックスとの接続が同じ場合には，その出現回数によらずそれらのボックスに相当するセルには同じ種類のデータが記述されているものと考える．

以上を表現するために，ボックスを用いた表構造の表現に繰り返し構造を表す+記号を導入することで，同じ接続を持つボックスが連続して出現する構造を表現する．図 2 で同じ接続を持つボックスの連続する部分を+記号を用いて表すと，図 3 のようになる．+記号の横の矢印は，+記号の中のボックスが縦向き双方向の接続で 1 度以上連続して現れることを意味している．ただし，ラベルの付けられたボックスについては，同じ接続を持つ複数のラベルつきボックスが連続して現れる場合でも，一つの繰り返しの中には含めないものとする．これは，プロパティとプロパティ値のようなラベルのついたボックスは，データの種類を指定された時点で，それらのボックスに記述されているデータの種類が異なっていることがわかっているからである．

+記号を用いて繰り返し現れるボックスをまとめて表現する場合，まとめられるボックス間の接続が同一であることが必要であるが，まとめられる連続したボックスの両端の接続に関しては異なっていてもよいものとする．これは+記号を用いてまとめられたボックスは，表の中での領域に相当するためである．+記号によりまとめられる連続したボックスの両端の接続はカッコの外側に付けられて，+記号によって表される領域と，その領域に隣接する部分との接続を表す．

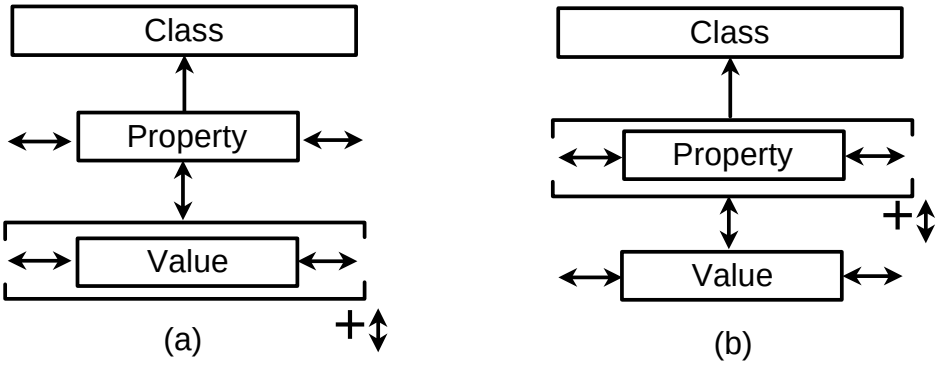

図 3: 表 2 の網掛けした部分の一般化表構造

例えば図 2 で “Property” というラベルを付けられたボックスは上端で上向きの接続を持つ．一方 “Property” と “Value” のボックス及びその間のボックスは，互いに垂直方向で双方向の接続を持つ．このような場合にも，“Property” と “Value” のボックス及びその間にあるボックスという 3 つのボックスの間にある接続のみに注目する．“Value” のラベルを持つボックスとその上にあるラベルのないボックスは，左右の辺にに同じ接続を持つ．さらにこれらのボックスは間に垂直・双方向の接続を持つ．そのため，“Value” のラベルを持つボックスとその上にあるラベルのないボックスを+記号を用いてまとめる．これらのボックスは上端と下端で異なる接続を持つが，その接続は+記号で表現されたボックスのまとまりに対する接続として表される．

図 3(a) は，このようにして得られた構造の表現を示している．同様に “Property” のボックスとその下のラベルの無いボックスと左右の辺に同じ接続を持つため，図 3(b) のように，+記号を用いてまとめることができる．

表 2 の網掛けした部分の構造を表現する場合，“ProductName”, “Pentium 4 2.80A GHz” のセルの左右のセルとの隣接関係が同じであるため，このように 2 通りの表現が得られることになる．

図 3 のようなボックスと+記号を用いた，一般化された表構造の表現を，一般化表構造と呼ぶものとする．

従来の研究ではある属性に対して複数の対応する属性値があるような場合には，どのセルに属性値が記述されているかはセル中のデータの字句的な特徴に基づいて決定されるものが多い．本研究においては，ある属性に対応する属性値が複数ある場合には，図 2 のようにそのうち一つの属性値が人手によって指

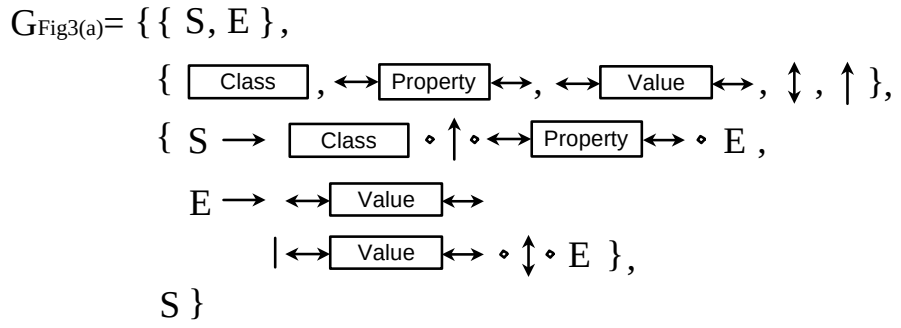

図 4: 図 3(a) に対応する文法

定され，それを元に表全体を解釈するための一般化を行うため，このようにセルの隣接の特徴に基づいてボックスをまとめて表現している．

図 3(b) で表現される構造は，複数のプロパティに対して一つのプロパティ値が対応するという誤った関係を表している．また同様に，複数のスーパークラスに対して一つのサブクラスが対応する関係など，表での表現においては不自然な関係が得られる場合もある．このような誤った一対多の対応を持つ一般化表構造が得られたときには，その一般化表構造は利用しないものとする．

図 3 に示した表の列の構造を表す一般化表構造は，縦方向に並んだボックスとその接続からなる特定の配列を表していると見なすことができる．すなわち図 3(a) であれば，上から順に “Class” というラベルのボックス，上向きの接続，“Property” というラベルの左右に双方向の接続を持つボックス，縦向き双方向の接続，括弧の中の “Value” というラベルのついた左右に双方向の接続を持つボックスの一度以上の連続という配列を表す．この配列は，ボックスとボックスの接続を終端記号とする文法で表現することができる．図 3(a) に対応する文法 $G_{Fig3(a)}$ を $G = (N, \Sigma, P, S)$ （N: 非終端記号の有限集合，Σ: 終端記号の有限集合，P: 生成規則の有限集合，S: 開始記号）の形で与えたものを図 4 に示す．3 種類のボックスと，2 種類のボックス間の接続が終端記号となる．また 2 つの生成規則を持ち，非終端記号 E によって，縦向き双方向に接続される連続する “Value” のラベルつきボックスが表現されている．

表 3: 異なる行や列にある関連する 3 つのセル

	CPU	RAM	HDD
OptiPlex240	P4 1.7GHz	512MB	60GB
Dimension8300	P4 2.4GHz	256MB	80GB
Pavilion505	Cel 2.4GHz	256MB	40GB

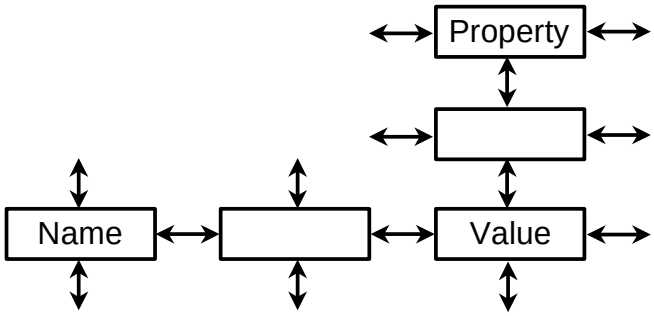

図 5: 表 3 の網掛けした部分の構造

3.3 異なる行や列内のセルの関係

2.2 節で述べたように，表中で異なる行や列にある 3 つ以上のセルが互いに関連を持つ場合，同じ行や列にある 2 つのセルの関係を表す構造を組み合わせた構造で表現できると考えられる．

例として表 3 を考える．表 1 同様に，コンピュータ部品の価格表である．表 3 に記述された “Dimension8300”, “RAM”, “256MB” は，それぞれある個体の名前 (名前というプロパティのプロパティ値)，プロパティ，プロパティ値であると解釈できる．このとき，まず同じ行にある “Dimension8300”, “256MB” の 2 つのセルについて，それらを含む行の構造を得る．同様に，同じ列にある “RAM”, “256MB” のセルについて，それらのセルを含む列の構造を得る．“Dimension8300”, “RAM”, “256MB” という 3 つのセルの関係を表現する構造を得るには，“Dimension8300” と “256MB”, “RAM” と “256MB” のそれぞれについて得られた構造を組み合わせればよい．よって “Dimension8300”, “RAM”, “256MB” の関係を表す部分は表 3 の網掛けした部分となり，この部分の構造をボックスとその隣接関係で表すと図 5 が得られる．

図 5 を見ると，“Value” というラベルの付けられたボックスとその左にある

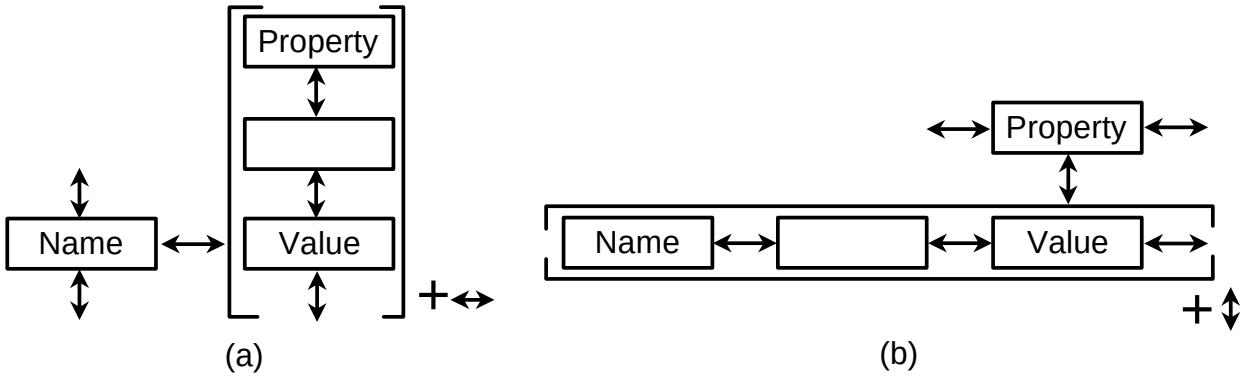

図 6: 表 3 の網掛けした部分の一般化表構造

ボックスは，周囲のボックスと同じ接続を持っている．そのため，図 5 で表される構造には繰り返し同じ種類のデータが記述されている可能性がある．また “Value” というラベルの付いたボックスにはその上にさらに 2 つのボックスが接続されており，これら 3 つのボックスは列を表す．そこで，“Property” というラベルのボックス，“Value” というラベルのボックス，その間にあるボックスという 3 つのボックスで表される構造を持つ列が繰り返し現れる構造を表現するため，+記号でこの列の構造を表す 3 つのボックスをまとめて表現する．これを表した一般化表構造が図 6(a) である．

また図 5 において，“Value” というラベルのボックスとその上のボックスで周囲との接続が同じであることから，“Name” というラベルのボックス，その間にあるボックスという 3 つのボックスからなる構造を持つ行が繰り返し現れる構造を表現すると，図 6(b) の一般化表構造が得られる．

このように図 5 では，どの箇所を繰り返し構造として表現するかによって，図 6(a)(b) に示した 2 通りの一般化表構造も得ることができる．図 6(a) はある行に注目して横向きに表を読んでいくことに相当し，図 6(b) はある列に注目して縦向きに表を読んでいくことに相当する．+の横の双方向の矢印は，カッコ中のボックスに相当するセルが縦向きに繰り返し現れる場合，全てのボックスは双方向の接続を持つことを表している．

ただし，列の構造と行の構造を組み合わせる場合には，+記号による繰り返し構造の表現をどちらかに限る．これは複数の行や列での繰り返しを考えると，“Name”, “Property” のボックスに相当するセルが表の端にあるという情報が失われるためである．

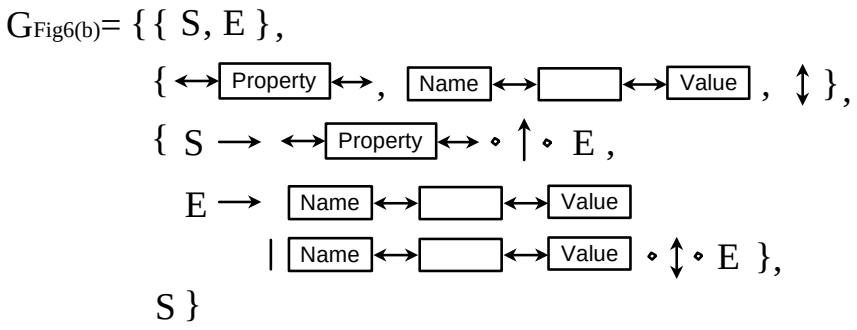

図 7: 図 6(b) に対応する文法

図 6(b) を，縦向きにボックスと接続からなる配列と見なすと，順に “Property” というラベルのボックス，縦向き双方向の接続，括弧の中の 3 つのボックスからなっている．図 6(b) が表す配列に対応する文法 $G_{Fig6(b)}$ は図 7 のようになる．“Property” というラベルのボックス，6(b) の括弧の中の互いに接続された 3 つのボックス，縦向き双方向の接続が終端記号となる．図 4 に示した文法と同様に 2 つの生成規則を持ち，非終端記号 E によって，縦向き双方向に接続される 3 つのボックスの繰り返し構造が表現されている．

図 6(a)(b) に示した一般化表構造は，その両方を用いても表 3 に記述されている全ての個体について，全てのプロパティとプロパティ値のペアを表現していない．図 6(a) では，“Name” や “Value” のラベルの付いたボックスは下の辺に双方向の接続を持っており，これらのボックスは表の最下行のセルと一致しない．よって図 6(a) は，ある一つの個体に関する複数のプロパティとプロパティ値が記述された構造を表現していることになる．また図 6(b) では，“Property” や “Value” のラベルの付いたボックスは右の辺に双方向の接続を持っており，これらのボックスは表の最も右の列にあるセルと一致しない．よって図 6(b) は，複数の個体に関して，その個体の名前と，ある特定の一つのプロパティとそのプロパティ値が記述された構造を表現していることになる．

この問題は，図 6 の一般化表構造により解釈されたデータ間の関係を用いて，ふたたび一般化表構造の獲得を行うことで解決できる．詳細については，4.1 節で述べる．

第4章　表構造の抽出

第1章で述べたように，本研究では表中のデータの関係を獲得するために，以下の手順からなるアプローチを取る．

1. 構造の解釈を与える
2. 解釈を与えた構造を一般化する
3. 表全体を解釈する

本研究では，ある構造がどのような関係を表すかという解釈を人手で与える．そのため，表中の一部のデータ間の関係を与える．関係を与えられたデータを含む領域の構造がその関係を表していると考え，構造とデータ間の関係を対応付ける．さらに，その領域の構造を，第3章で述べたように形式的に表現し，一般化を行う．これにより，初めに人手でデータ間の関係を与えた部分と同様の関係を表す構造の，一般化表表現が得られる．最後に，得られた一般化表構造が一致する構造を表全体から探す．一致した部分からは新たにデータ間の関係を得ることができる．

以下ではまず，上に示したそれぞれの処理について詳しく述べる．次にいくつかの表に対する適用例から，提案手法の問題点を示す．さらに，複数の表から得られる情報を利用する方法と，その問題点について述べる．

4.1　処理の手順

以下に本研究のアプローチの各ステップについて，順に説明する．

構造の解釈を与える　本研究では，表中の一部のデータ間の関係をRDF(Resource Description Framework) ステートメントで記述することにより，それらのデータが含まれる構造の解釈を与えるものとする．RDFはウェブ上にあるリソースを記述するための統一された枠組みであり，W3Cにより規格化がなされている．RDFは特にメタデータについて記述することを目的としており，セマンティックWebを実現するための技術的な構成要素の一つとなっている．またオントロジー記述言語であるOWLもRDFに基づく．本研究では解釈として与えた関係に従って表から新たな関係を獲得するため，セマンティックWebのためのオントロジー獲得に，RDFステートメントによる構造の解釈の記述を与えることは適当であると考えられる．

表1では，“Processor”と記述された一行目のセル，“ProductName”と記

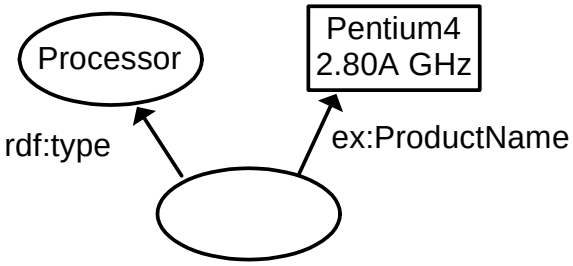

図 8: 表 1 中のデータの関係

述された 2 行目 2 列目のセル，“Pentium 4 2.80A GHz” と記述された 4 行目 2 列目のセルはそれぞれある個体のクラス・プロパティ・プロパティ値を表していると解釈できる．この関係を RDF のグラフで表すと図 8 のようになり，2 つの RDF ステートメントによって記述される．

このように表構造が表す関係は，単一の RDF ステートメントでは表現できないことが多い．そこで図 8 のように，特定の表構造があらわす関係を記述する RDF ステートメントの集合を，エピソードと呼ぶものとする．

解釈を与えた構造を一般化する　解釈が与えられた構造を一般化し，一般化表構造を得る手順は以下のようになる．まず，与えたエピソードに含まれる RDF ステートメントのリソースやプロパティが表中で出現するセルを探す．次に, 見つかったセルを含む行や列の構造をボックスとその接続で表現し，さらに同じ接続を持つボックスが繰り返し現れる部分を+記号によってまとめ，一般化表構造を得る．繰り返し現れている部分を発見するために，以下のような処理を行う．まずボックスとその接続によって表現された構造を，行や列の配列とみなす．次にその部分配列が連続して現れる部分を探し，見つかった場合は+記号による繰り返し構造に置き換える．見つからなかった場合は，部分配列の長さを大きくして同じ処理を繰り返す．

与えたエピソードのリソースやプロパティが現れたセルが，一般化表構造のラベルつきボックスとなり，クラス・プロパティ・プロパティ値などの与えたエピソードにおける役割がラベルとしてつけられる．ただし subClassOf や instanceOf のようなプロパティは，プロパティ名が表の中に現れるわけではなく，表の構造それ自体によって表現される．そのため表中で現れるセルを探さず，一般化表構造中のラベルつきボックスとして含めない．

表 1 に対して図 8 のエピソードを与えた場合には，得られる一般化表構造

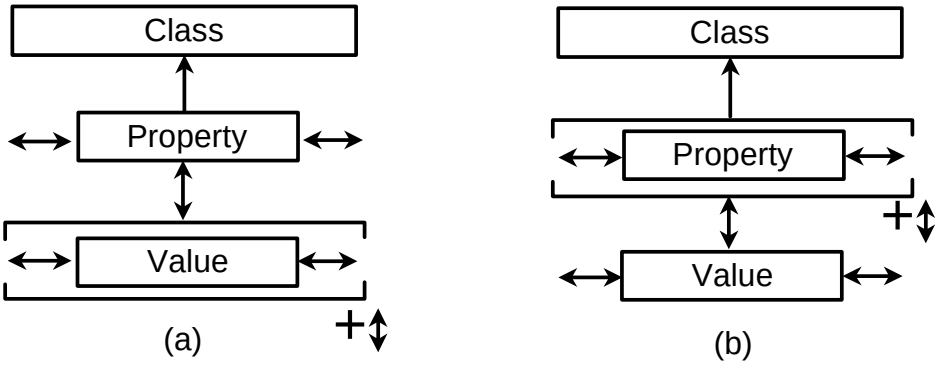

図 9: 表 1 から得られる一般化表構造

は図 9(a)(b) のようになる．2 通りの一般化表構造が得られるのは，プロパティ名である “ProductName” が記述されたセルと，プロパティ値である “Pentium 4 2.80A GHz” が記述されたセル，及びそれらの間に配置されたセルで，周囲のセルとの隣接関係が同じであるため，プロパティ名とプロパティ値のどちらが連続して出現しているのかが表構造からは判断できないためである．

しかし図 9(b) は一つのプロパティ値に対して複数のプロパティが対応するという，誤った関係を表している．このように，与えたエピソードに対して繰り返し構造があらわす 1 対多の関係が誤っている一般化表構造は，以降の処理で利用しないものとする．また与えたエピソードがあらわす関係が表中で特徴的な表構造と対応していない場合には，一般的過ぎてどのような箇所にもマッチする一般化表構造が得られることがある．例えば，全てのボックスでその隣接関係が同じ一般化表構造が得られた場合には，特定の関係に対応する表構造を表現しているとは言えないため，その一般化表構造は利用できない．

表全体を解釈する　得られた一般化表構造が表中でマッチする構造を探すことで，新たなエピソードが得られる．一般化表構造のマッチは，表中で連続するセルのボックスによる表現と，一般化表構造中のボックスを順に比較していくことで行う．二つのボックスが同じ接続を持つ場合に，それらのボックスは一致するとする．与えられたセルの配列のボックスによる表現が一般化表構造が表すボックスの配列とマッチすれば，一般化表構造の各ラベルつきボックスと一致したセルのデータは，一般化表構造獲得に用い

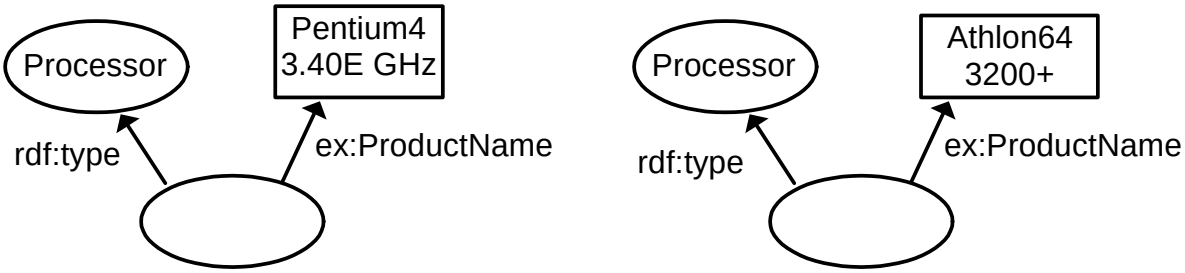

図 10: 表 1 から新たに得られるエピソード

た元のエピソードの対応するリソースやプロパティと同じ関係にあると考えられる．

例として，表 1 から得られた一般化表構造である図 9(a) を，“Processor” のセルを起点として表 1 にマッチさせることを考える．図 9(a) の “Class” のボックスは，一行目の “Processor” のセルをボックスで表現したものと一致する．ここで “Class” のボックスは下の辺で “Property” のボックスと接続されているため，次に Processor の下にあるセルをボックスで表現したものとを比較する．その結果 “ProductName” のセルが一致する．さらに “Property” のボックスも下の辺で “Value” のボックスと接続されているため，“ProductName” の下の “Pentium 4 3.40E GHz” のセルをボックスで表現したものと Value のボックスを比較する．これらも一致するので，クラス，プロパティ，プロパティ値を “Processor”, “ProductName”, “Pentium 4 3.40E GHz” とするエピソードが得られる．図 9(a) は，“Value” のボックスは，＋記号により任意の回数繰り返し出現することを表しており，さらに下のセルを調べていくと，“Pentium 4 2.80A GHz”, “Athlon 64 3200+” のセルのボックスによる表現とも一致する．“Pentium 4 2.80A GHz” は最初に与えたエピソードに含まれているので，最終的に図 10 のグラフで示した二つのエピソードが新たに得られたことになる．

新たなエピソードの獲得を行った結果，ある語がプロパティであるという記述を含むエピソードと，同じ語がクラスであるという別のエピソードが同時に得られることがある．このように，既知のエピソードや新たに得られたエピソードの間に互いに矛盾する記述がある場合にはそれらを除く．さらにこのような結果が得られた原因は一般化表構造を得るときに用いたエピソードに問題があった可能性があるので，一般化表構造を得るのに用い

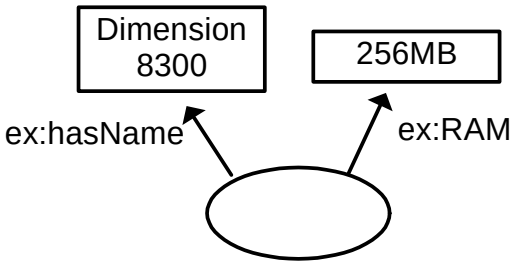

図 11: 表 3 中のデータの関係

たエピソードも除く.

以上の処理の結果，新たに得られたエピソード中のリソースやプロパティが一般化表構造のマッチによって獲得したのとは別の構造の中で出現していることがある．そのような場合には，新たに得られたエピソードを用いて，そのエピソードの獲得に用いられたのとは違う一般化表構造が得られることがある．そこで，表からより多くの情報を得るために，エピソードを与えて一般化表構造を得て，新たなエピソードを獲得するというサイクルを，新たなエピソードが得られなくなるまで繰り返す.

処理の繰り返しが必要となる場合について，表 3 を例に説明する．表 3 に “Dimension8300”, “RAM”, “256MB” の間の関係を記述する図 11 に示すエピソードを与え，処理を適用すると，図 6(a)(b) の 2 通りの一般化表構造が得られる.

しかしこの 2 つの一般化表構造に基づいて新たな情報の獲得を行っても，表 3 に記述されたすべての個体のプロパティとプロパティ値の関係を得ることができない．なぜなら図 6(a) の一般化表構造では，ある一つの個体に関する複数のプロパティとプロパティ値の組み合わせを得る．また図 6(b) の一般化表構造では，ある一つのプロパティに関して，表に記述された個体の名前とそのプロパティのプロパティ値の組み合わせを得る．したがって，最初にエピソードを与えた時に記述したプロパティ “hasName” の値に “Dimension8300” を持つ個体の以外の表に記載された個体は，プロパティ “RAM” 以外のプロパティについてプロパティとプロパティ値の組み合わせを得られない．しかし，図 6(b) に示した一般化表構造から得られる，“OptiPlex240” や “Pavilion505” を “hasName” の値として持つ個体の，プロパティ “RAM” の値について記述したエピソードを解釈として用いて処理を繰り返すと，表 3 に記載されたすべての個体について， “CPU”, “RAM”, “HDD” という 3 つのプロパティとプロパティ値の記述

CPU		
AMD		
AMD2G4XPB	AMD XP 2400 2GHz	$145.00
AMD2G5XPB	AMD XP 2500 2.04GHz	$165.00
AMD2G6XPB	ATHLON XP 2600+	$199.00
AMD2G7XPB	ATHLON XP 2700+	$219.00
INTEL		
INT2G4C1A	Celeron 2400Mhz	$119.00
INT2G6C1A	Celeron 2600Mhz	$155.00
INT3G0P1B	Pentium 4 3Ghz	$479.00
INT32G0P1B	Pentium 4 3.2Ghz	$699.00
COOLING		
CASE COOLING		
FAN60BB	Fan 60mm Ball Bearing	$8.00
FAN80BB	Fan 80mm Ball Bearing	$8.00
FAN-80LVG	"Antec" (USA) 80mm LV Fan	$14.00

図 12: 階層的な分類を持つ表

が得られることになる．

4.2 処理例

より複雑な構造を持つ表に対する処理の例を挙げ，どのような関係が得られるかや，誤った関係が得られる場合について説明する．

図 12 は，表 1 同様に PC 部品の価格表であり，Web 上に Excel ファイルとして公開されていたものである．ファイル全体には多数のデータが記述されていたので，図 12 にはその一部を示している．

図 12 では一つの個体の情報が 3 つの列からなる行に記述され，各列には個体のプロパティ値が記述されている．一つの幅の広いセルから構成される行は個体の分類を表す．これらは個体の属するクラスに相当すると解釈できる．またこの表では，“CPU” や “COOLING” のような上位の分類と “AMD” や “INTEL” のような下位の分類の二階層の分類が用いられており，クラスの階層関係を表していると解釈できる．

この表に対して，図 13 のグラフで表されるエピソードを与えてアルゴリズムを適用すると，図 14 のようなネスト構造を持つ一般化表構造が得られる．カッコの中のボックス “ProductID”, “Name”, “Price” による繰り返しでは，いずれのボックスも縦向き双方向に接続され，“ProductID”, “Name”, “Price” の縦方

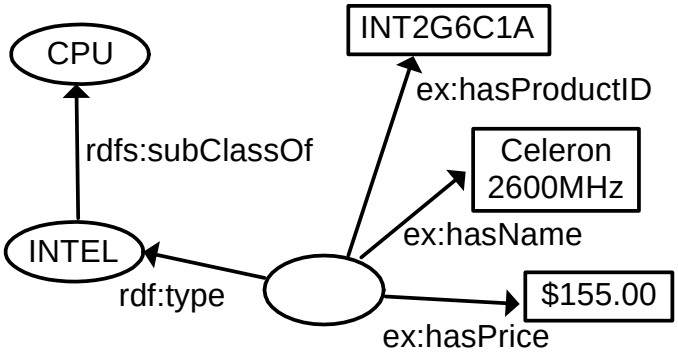

図 13: 表 12 のデータの関係を記述したエピソード

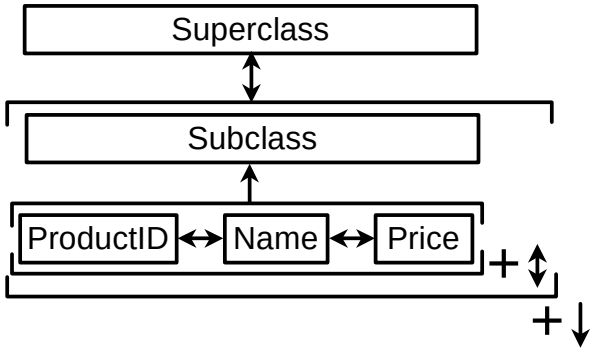

図 14: 表 12 から得られる一般化表構造

向の繰り返し構造で構成されるブロックは，上端では双方向に接続され，下端では下向きの片方向で接続される．

図 14 に対応する文法 G_{Fig14} を図 15 に示す．非終端記号 E_1 は，図 14 の外側の括弧で囲まれた部分を表す．非終端記号 E_2 は，図 14 の内側の括弧で囲まれた部分を表す．図 3 や図 6 に示した一般化表構造と異なり，図 14 では繰り返し構造がネストしているため，対応する文法 G_{Fig14} は文脈自由言語となっている．

図 14 に示した一般化表構造を用いてエピソードの獲得を行うと，表 5 から 1304 個のエピソードが得られ，そのうち 1115 個のエピソードが表の構造を正しく解釈して得られたエピソードであった．新たに獲得されたエピソードでは，表に記述された商品について，プロパティ “hasProductID”, “hasName”, “hasPrice” の値が得られた．さらに “AMD” が “CPU” のサブクラスであること，“CASE COOLING” が “COOLING” のサブクラスであるなど，クラス階層の記述が得られた．

残りの 189 個のエピソードでは，誤ったクラス階層の記述が得られた．図 14

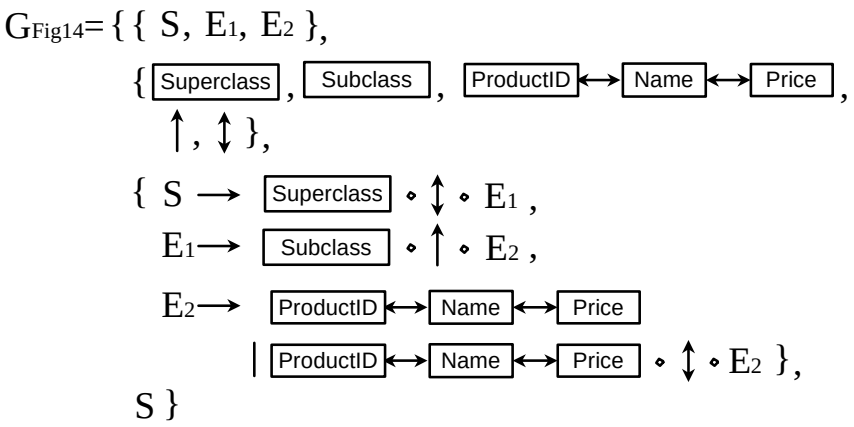

図 15: 図 14 に対応する文法

HARD DRIVES		
IDE		
HDDSEA40G	ST340014A Seagate 40Gb	$103.00
HDDSEA80G	ST380011A Seagate 80Gb	$122.00
HDDSEA120	ST3120023A Seagate 120Gb	$165.00
SATA		
HDD-WD1200	WESTERN DIGITAL 120GB	$209.00
HDD-WD2000	WESTERN DIGITAL 200GB	$359.00
FLOPPY DRIVES		
FDD1.44PA	1.44 MB PANASONIC FDD	$18.00
FDD1.44PB	1.44 MB PANASONIC FDD BLACK	$18.00
MEMORY		
RAM PC 100 - 150		
MEM128S133	OR "SPECTEK" Brand	$50.00
MEM256S133	"AM1" OR "SPECTEK"	$89.00

図 16: 誤ったクラス階層が得られる部分

の一般化表構造では，幅の広いセルが二つ連続している構造の上側のセルをスーパークラスとし，下に 3 つのセルが隣接する幅の広いセルをサブクラスとして解釈する．しかし実際には図 16 に示した部分のように図 “CPU” や “COOLING” と同列であるスーパークラスをあらわすセルの下に，そのサブクラスをあらわすセルなしに直接商品の情報が記述されている部分が存在した．そのような部分では，本来 “CPU” や “COOLING” と同列であるクラスが，その上部に現れるスーパークラスのサブクラスであると解釈される．

図 16 に示した部分では，“HARD DRIVES” という分類には，“IDE” や “SATA” などの “HARD DRIVES” のサブクラスに相当するより細かい分類が示されて

Drives		
Maxtor 3.5" IDE		
40GB Maxtor Diamondmax	£42.00	£40.25
80GB Maxtor Diamondmax	£57.00	£54.63
80GB Maxtor Diamondmax	£68.40	£65.55
Maxtor Serial ATA150		
80Gb Maxtor Serial-ATA	£78.00	£74.75
120Gb Maxtor Serial-ATA	£94.80	£90.85
Maxtor SCSI		
Maxtor Atlas 10K III 18.4GB	£78.00	£74.75
Maxtor Atlas 10K III 36.7GB	£126.00	£120.75

図 17: クラスの記述と個体の記述で構造に差のない表

いる．しかし “FLOPPY DRIVES” という分類の下には，サブクラスに相当する分類がない．そのため，図 14 に示した一般化表構造を用いると，“FLOPPY DRIVES” は “HARD DRIVES” のサブクラスとして獲得される．これは元の表では，スーパークラスとサブクラスで表構造に差異がなく，異なるセルの背景色を用いることで区別している場合があることが原因である．

図 12 に示した表のような，クラスを表す行と個体の情報が記述された行との構造的な差のない表は他にもあり，図 17 にそのような表の一部を示す．図 17 では，上端に “Drives” という個体のクラスに相当する語が記述されている．さらに，2, 6, 9 行目には，“Maxtor 3.5” IDE”, “Maxtor Serial ATA150” など，“Drives” のサブクラスに相当する語が記述されている．しかし，図 12 に示した表と異なり，“Maxtor 3.5” IDE” などのサブクラスに相当する語が記述された行は，個体の情報が記述された行と同様に 3 つの行に分割されている．そのため，構造だけに注目すると，クラスの記述された行と個体の情報が記述された行を区別することができない．そのため，このような表では正しい一般化表構造を得ることができない．

4.3 複数の表の利用

提案手法では，初めに人手によりエピソードを与える処理が必要となるため，自動的に多くの表を処理することができない．そこで，ある表で得られたエピソードを，他の表での解釈を与えるのに用いることを考える．

一般には同じ表構造でも，異なる表ではそのセマンティクスが異なるため，ある表で得られた一般化表構造は別の表で利用することはできない．しかしエピソードについてはある程度一般性があるため，ある表で得られたエピソードを使って他の表で一般化表構造の獲得を行うことができると考えられる．

ここで，以下のことをあらわすエピソードを与え，ある PC 部品の価格表に新たなエピソード獲得のアルゴリズムを適用した．

- ある個体はプロパティ hasName の値として “Pentium4 2.80GHz” を持ち，クラス “CPU” に属する

その結果，クラス “CPU” とのクラス-インスタンス関係を記述する 32 のエピソードが得られた．ここで得られたエピソードは，いずれも正しい関係を記述したものであった．次に，得られたエピソードを用いて，Web から収集された約 2000 の PC 部品に関連する表を対象に，パターンを得てエピソードの獲得を行った．その結果 7 つの表から，新たにクラス CPU のインスタンスを記述する 317 のエピソードが得られた．しかし新たに得られたエピソードの中には，実際にはプロパティとプロパティ値の組であるものが，クラスとインスタンスの組として記述されたエピソードも多くあった．これは，同じ語でもその意味や他の語との関係が表によって異なる解釈がされることがあるためである．例えば PC 部品の価格表では，“CPU” と “Pentium4 2.80GHz” いう語は，表によってクラスとそのインスタンスして扱われることも，プロパティとそのプロパティ値として扱われることもある．この場合，“CPU’ と “Pentium4 2.80GHz” の 2 つの語の組み合わせがプロパティとプロパティ値として扱われている表があり，そこから得られた一般化表構造がその表の中で一致する箇所を探すと，同様にプロパティとプロパティ値の組み合わせを記述した箇所に一致した．

もう一つの問題として，同じものを表す場合にも異なった表記がされることが多く，語によっては他の表で現れる可能性が非常に低いということが挙げられる．上に述べた例では，“CPU” という語と “Pentium4 2.80GHz” という語の組み合わせは，約 2000 の表のうち 7 つの表にしか出現していない．

そのため，ある表から得られたエピソードを用いて別の表で一般化表構造を得て，自動的に多くの表を処理するという方法を用いるには，対象とする関係を限定する必要がある．

第5章　評価

Web 上の表データに対して本研究の提案手法を適用し，実際の表からどのようなデータ間の関係を得られるかを調べる．さらに，従来の研究で提案された手法と比較し，提案手法の有効性を明らかにする．

5.1　獲得できる関係

本研究の手法を適用することで，主にどのような構造からどのような関係を獲得できるのかを調べる．

5.1.1　評価方法

実際に Web 上のいくつかの表に対して提案手法を適用した結果，以下に示す関係が得られた．

プロパティ　表 1 や表 3 のように，個体の持つプロパティとプロパティ値が対応する属性と属性値などの形で表中に記述される場合と，図 12 に示した表のようにプロパティ値のみが記述され，プロパティの記述がない場合がある．プロパティ値のみが記述される場合には，エピソードを与える際に人手でプロパティ名を記述する必要がある．

クラス-インスタンス　一行に一つの個体の情報が記載されている場合，いくつかの行のまとまりにラベルがつけられて，個体のクラスを表すことがある．例えば図 12 に示した表では幅の広いセルが挿入され，個体の属するクラスを表している．また表 1 のように，表の上端にキャプションに相当するセルを持ち，それが表に記載された個体の属するクラスを表す表もある．

クラス階層　図 12 に示した表のように，個体のクラスを表すラベルが階層的になっており，それらがクラス階層を表す場合がある．

プロパティ階層　属性が複数の行や列にまたがって，階層的に配置されている表がある．階層的に配置された属性間の関係は場合によってさまざまであるが，属性として記述された語の意味から，プロパティ階層として解釈するべき場合がある．

どのような関係が得られるかは，表の内容や構造によって異なる．そこで，これらの関係がどのような表からどの程度得られるのかを調べる．

本研究の手法では，同じ種類のデータが繰り返し記述される構造を一般化することで関係を獲得する．例えば，ある特定のプロパティに対する，複数の個

Rate(%)	Regular	Float
Regular Fixed Deposit		
1 Year	5.25	5.25
2 Years	5.35	5.35
3 Years	5.35	5.35
Fixed Deposit		
3 Months	4.4	4.4
6 Months	4.95	4.95
9 Months	5.05	5.05
1 Year	5.15	5.15
2 Years	5.25	5.25
3 Years	5.25	5.25

(a)

Items & Period		Regular	Float
Fixed Deposit	3 Months	4.4	4.4
	6 Months	4.95	4.95
	9 Months	5.05	5.05
	1 Year	5.15	5.15
	2 Years	5.25	5.25
	3 Years	5.25	5.25
Regular Fixed Deposit	1 Year	5.25	5.25
	2 Years	5.35	5.35
	3 Years	5.35	5.35

(b)

Tour Code			DP9LAX01AB	
Valid			01.05.-.30.09.04	
Class/Extension			Economic	Extended
Adult	PRICE	Single Room	35,450	2,510
		Double Room	32,500	1,430
		Extra Bed	30,500	720
Child		Occupation	25,800	1,430
		No Occupation	23,850	720
		Extra Bed	22,990	360

(c)

図 18: Complex Table クラスの表

体のプロパティ値や，ある特定のクラスに属する複数のインスタンスの記述は，繰り返し構造に基づいて獲得できる．そこで，このような本研究の提案手法での一般化が，関係の獲得にどの程度有効であるかに注目して，提案手法の適用結果を評価する．ここでは，プロパティ-プロパティ値の関係については，得られたプロパティの数，同様にクラス-インスタンスの関係については得られたクラスの数，クラス階層・プロパティ階層については得られた上位クラス・上位プロパティの数を調べることで評価を行う．

5.1.2 データセット

表の構造によって得られる結果が異なると考えられる．[12, 10] では，レイアウトに基づいて以下のような表のクラスが提案されている．

1-dimensinal table このクラスに属する表は，上端に一行かそれ以上の属性

を表す行を持つ．属性は複数行にまたがって階層的に表現されることがある．属性値は同じ列の属性に対応する．表 1 はこのクラスに属する．

2-dimensinal table　表の上端と左端に属性が記述される．1-dimensional table 同様，属性は複数の行や列にまたがって階層的に表現されることがある．表 3 はこのクラスに属する．

Complex table　このクラスの表は，さまざまな特徴を持つ．[12] では，いくつかの特徴に基づいてさらに次のように 3 つのクラスを示している．

Partition label　このクラスの表では，表をいくつかに分割するラベルが含まれる．分割された各部分は共通の属性に対応する属性値を持つ．属性は表の上端にだけ記述され，各部分には記述されないことがある．図 18(a) にこのクラスに属する表の例を示す．

Over-expanded label　このクラスの表は，複数の行や列にまたがるセルを含む．そのようなセルは，他のより幅の小さいセルと隣接して属性の階層関係を表す場合や，連続するセルが同じデータを持っていることを表す場合がある．図 18(b) にこのクラスに属する表の例を示す．

Combination　属性と属性値の対応などを持たない，構造的に相互に独立した複数の表によって構成される．図 18(c) にこのクラスに属する表の例を示す．図 18(c) では，1–2 行目と 3–9 行目が互いに独立した表となっている．

ここでは，これら 5 つのクラスから表を集め，提案手法の適用の結果を比較する．またドメインによっても現れる表の構造やデータの種類が異なる．ここでは価格表, タイムテーブル, 統計データの 3 つのドメインから表を集めた．本研究の手法で獲得されるオントロジーは，セマンティック Web においてエージェントが高度な検索を行う場合などに利用することが想定される．価格表は電気製品など一部のデータに関しては，すでに複数のメーカーや店舗の情報をまとめて整理したサイトが存在している．しかしそれ以外の製品については，値段や特性などを指定して複数のメーカーや店舗を横断的に調べるには手間がかかるため，本研究のような手法でさまざまな価格表のデータを RDF/OWL によるオントロジーの記述として得ることは有用である．同様に，タイムテーブルも，電車の乗り継ぎの時刻を調べるサイトなどは充実しているが，飛行機や船，バスやイベントの時間などをまとめて調べてプランニングするなどの目的には，本研究のようにさまざまなタイムテーブルからオントロジーを獲得することが

必要となる．また，統計データはその他の種類の表に比べて特に多くのデータを含むことが多く，Web での知識の知識の蓄積という点で重要であると考えられる．

今回は 5 つの構造のクラスごとに 5 つの表を収集し，あわせて 25 の表をさらに 3 つのドメインで収集した．以降ではこの合計 75 の表を対象として提案手法を適用した結果を示す．

5.1.3 評価結果

集めた表に対して，本研究の提案手法を適用した結果を表 4 に示す．数字は表のクラスごとの得られた関係の数の合計を示す．1-dim, 2-dim はそれぞれ 1-dimensional table，2-dimensional table を表す．

表 4: 得られた関係の数

関係	1-dim	2-dim	Partition labeled	Over- expanded label	Combination
プロパティ	75	47	36	33	36
プロパティ (プロパティ値のみ)	5	30	21	20	10
クラス -インスタンス	0	0	33	2	4
クラス階層	0	0	2	0	0
プロパティ階層	0	0	2	3	0

表 4 から，提案手法を用いると，主にプロパティが得られることがわかる．これは表 3 のように，表の上端や左端にある属性とその属性に対応する属性値の組み合わせが，プロパティとプロパティ値となっている表が多くあるためである．

プロパティ値だけが記述され，プロパティが表中に記述されない表としては，図 12 に示した表のように属性値に対応する属性が記述されていないものや，表 5 のように，属性がプロパティ値となっているものがある．

属性がプロパティ値となっている表の例を表 5 に示す．表 5 は曜日と時間ごとの当番表である．一行目に記述された “Monday”, “Tuesday, “Wednesday” は，「当番の曜日」というプロパティのプロパティ値であると考えられる．同様に，

一列目に記述された “午前”, “午後” は,「当番の時間」というプロパティのプロパティ値であると考えられる．また人名は「当番の人」というプロパティのプロパティ値であると考えられる．表 5 のような，属性がプロパティ値として解釈できる表は，2-dimensional table クラスや Over-expanded label クラスに多く見られる．

表 5: 属性がプロパティ値となる表

	Monday	Tuesday	Wednesday
Morning	Clemens	Aaron	Celine
Afternoon	Aaron	Celine	Clemens

クラス-インスタンスの関係の多くは，Partition labeled クラスの表から得られている．これは，このクラスの表では，図 12 に示した表のように表中に挿入されたラベルが個体のクラスを表していることが多いためである．同様にクラス階層も，図 12 のように階層的にラベルの付けられている表から得られたが，そもそも階層的なラベルを持つ表が少なく，得られたクラス階層の数は非常に少なかった．

Over-expanded label クラスや Combination クラスに属する表では，属性が記述された部分が複雑な構造を持つことが多い．階層的に配置された属性を持つ表には，プロパティ階層を得られるものがあった．例えば，表 6 では，1 行目に記述された “Storage”, “Software” はプロパティを表し，2 行目に記述された，“Hard Drive”, “DVD ROM/RW”, “Zip”, “OS”, “Utility” はそれらのプロパティのサブプロパティを表す．

表 6: 属性がプロパティ階層を表す表

	Storage			Software	
	Hard Drive	DVD ROM/RW	Zip	OS	Utility
Dimension8300	80GB	DVD ROM	250MB	Win XP	Premiere
G310 Series	40GB	DVD ROM	250MB	Win XP	Acrobat
PCV RS220	60GB	12x DVD-RW	None	Win 2000	Acrobat

しかし，階層的に配置された属性のうちでプロパティ階層を表すものは一部であり，得られたプロパティ階層の数は少なかった．階層的な属性が表すその他の関係には，表7のように，クラスとそのクラスに属する個体が持つプロパティなどがあった．表7では，1行目に記述された “Processor”, “Memory” はクラスを表し，2行目に記述された，“Type”, “Frequency”, “Size”, “Characteristics” はそれらのクラスに属するインスタンスのプロパティを表すと考えられる．

表 7: 属性がクラスとプロパティを表す表

	Processor		Memory	
	Type	Freqency	Size	Characteristics
Dimension8300	Pentium4	2.6GHz	256MB	DDR SDRAM
G310 Series	Celeron	2.2GHz	128MB	184pin DDR
PCV RS220	Pentium4	2.4GHz	512MB	DDR SDRAM

5.2 従来研究との比較

第1章で述べたように，同じ表構造でも表によって表す関係は異なる．ある構造をどう解釈するのが適当かは，表中に記述されたデータの意味によって決まるため，知識ベースを用いない，データの字句的な特徴や表の認知モデルなどに基づくアプローチでは，詳細な構造の解釈が難しい．本節では，本研究での提案手法が表ごとに構造を適切に解釈できているかを，[12] で提案された手法と比較することで評価する．

5.2.1 手法の特徴

[12] で提案された手法は，表の認知モデル [11] に基づいて表を解釈し，表からフレームを獲得する．この手法は以下のような特徴を持つ．

領域への分割 表を属性が記述された領域と，属性値が記述された領域に分割する．各領域が属性を表すか属性値を表すかの決定は，セル中のデータの種類 (文字列・数値・日付など) と，その配置に基づいて行う．

連続する属性と属性値の対応付け 属性を記述された領域と属性値が記述された領域の位置関係から，それらの関連を判断する．基本的に属性を記述された領域と属性値が記述された領域が連続しているとき，それらが対応し

ていると見なす．属性を記述した領域が連続しているとき，フレームのメソッド名・パラメータ名はそれらの属性を結合して作られる．

一方本研究の手法は，構造の解釈を人手で与える．以下のような特徴を持つ．

解釈の割り当て　表中のデータ間の関係の記述を与えることにより，そのデータの含まれる構造と，その構造が表すデータ間の関係の対応付けを行う．

表構造の一般化　隣接するセルの辺の大小関係や，同じ特徴を持つ構造の連続する出現に注目して表構造を一般化する．

5.2.2　比較方法

表中の語の関係を RDF ステートメントで記述する場合，表中の語は RDF ステートメントのリソースやプロパティとなる．そこで，リソースやプロパティとなっている語の関連を，[12] の手法と本研究の手法を適用することで獲得できるかという点に注目する．

[12] で提案された手法は表からフレームを得るものであり，表中の語の関係を直接得るものではない．一方本研究の手法では，表中の語の関係を記述する RDF ステートメントを得る．そこで，これらの手法の適用結果を比較するために，[12] の手法の結果を RDF ステートメントに変換することを考える．

[12] の手法で得られるフレームのメソッド名やパラメータ名は，表中の語を元に作られ，また表中に記述されたデータがメソッドやパラメータの値となる．このためメソッド名とメソッドの値や，パラメータ名とパラメータの値はプロパティとプロパティ値などの何らかの関係を持っていると考えられる．同様に，連続して配置された属性が連結されてメソッド名やパラメータ名を構成する場合も，それらの属性の間には何らかの関係があると考えられる．そこで，[12] の手法で得られたフレームのメソッド名とメソッドの値，パラメータ名とパラメータの値，連結してメソッド名やパラメータ名を構成する語の間の関係は，RDF ステートメントで記述される何らかの関係に対応していると見なし，フレームを RDF ステートメントに変換する．実際に調べた結果，メソッド名とメソッドの値や，パラメータ名とパラメータの値の間の関係は，以下の 3 通りの場合があった．

- メソッド名（パラメータ名）が個体の持つプロパティに対応し，メソッド（パラメータ）の値がプロパティ値に対応する．
- メソッド名（パラメータ名）が個体の属するクラスに対応し，メソッド（パラメータ）の値が何らかのプロパティのプロパティ値に対応する．

- メソッド名（パラメータ名）とメソッド（パラメータ）の値の両方が，個体の持つ 2 つの別のプロパティのプロパティ値に対応する．

また，階層的に配置された属性が連結されてメソッド名やパラメータ名を構成する場合，それらの属性の間の関係は，以下の関係を表す場合があった．

- 属性がプロパティを表し，下部に記述された属性が上部に記述された属性の下位プロパティとなっている．
- 上部に記述された属性がプロパティを表し，下部に記述された属性がそのプロパティのプロパティ値となっている
- 上部に記述された属性が個体の属するクラスを表し，下部に記述された属性がそのクラスに属する個体のプロパティを表す（対応するプロパティ値はより下の部分に属性値として記述される）．
- 上部に記述された属性が個体のプロパティを表し，下部に記述された属性はまた別のプロパティのプロパティ値を表す（上部に記述された属性が表すプロパティのプロパティ値はより下の部分に属性値として記述される）．

表中に属性と属性値の組み合わせや階層的に配置された属性があるとき，いずれの関係を表すと解釈するべきかは，記述されているデータの意味によって決まる．そのため，データの字句的な特徴や表の認知モデルなどに基づく [12] の手法は，属性と属性値の対応や，階層的に配置された属性をどのように解釈するべきか判断できない．そこで，ここではメソッドとメソッドの値や，パラメータとパラメータの値は，プロパティとプロパティ値の関係を表すと見なす．さらに，メソッドやパラメータにおいて連結された語は，プロパティ階層関係を表すと見なす．

一方本研究の提案手法では，属性と属性値の組み合わせや階層的に配置された属性がいずれの関係を表すかを，一部のデータ間の関係を人手で与えることで指定する．しかし，表によっては解釈を与えた構造の一般化がうまく行えないことがあり，誤った関係が得られたり，表に現れている関係の一部しか獲得できないことがある．

以上の各手法の特徴を考慮して，5.1.1 節で述べた評価方法と同様に，プロパティ-プロパティ値の関係については得られたプロパティの数，クラス-インスタンスの関係については得られたクラスの数，クラス階層・プロパティ階層については得られた上位クラス・上位プロパティの数に基づいて評価を行う．これにより，[12] の手法では区別しきれない関係がどの程度あるか，また提案手法

表 8: 不足している記述の数の比較

表のクラス	[12] の手法	本研究の手法
1-dimensional table	10	1
2-dimensional table	13	5
Partition Label	58	14
Over-expanded Label	19	16
Combination	26	23

の一般化がどの程度有効であるかを調べる．

評価は以下の手順で行う．

Step 1 表中の語の関係を RDF ステートメントを用いて人手で記述する．個体に相当する匿名リソースやセル中の語として現れていないプロパティは適当に補う．

Step 2 各手法を適用する．対象とする表は 5.1 節で述べた評価で用いたものと同じであり，異なる表のクラス・異なるドメインから収集したものを用いる．

Step 3 Step 1 で人手で記述した関係に対して，Step 2 で得られた関係において不足しているクラス，プロパティ，クラス階層とプロパティ階層の記述の数を比較する．

5.2.3 比較結果

集められた表に対して 5.2.2 節で述べた評価を行う．人手で記述した関係と比較して，不足しているプロパティ，クラス，クラス階層とプロパティ階層の数を表のクラスごとに合計した．結果を表 8 に示す．

1-dimensional table, 2-dimensional table に関しては，[12] の手法，本研究の手法ともに属性と属性値の対応を正しく得ることはできていた．結果に差が出ているのは，属性はプロパティとして解釈するべき場合とプロパティ値として解釈するべき場合があることが原因である．

例えば表 3 において，1 行目に記述された “CPU”, “RAM”, “HDD” などの属性はこの表に記述されている個体のプロパティであると解釈するのが適当である．一方，表 5 では属性をプロパティ値として解釈するのが適当である．

どちらと解釈するべきかは，属性の意味によるため，セル中のデータの種類

や配置の特徴に基づく [12] の手法はこれらを区別することができない．一方本研究の手法は，表ごとに人手による構造の解釈を与え，構造にデータ間の関係を対応付けるため，属性がプロパティかプロパティ値かは正しく解釈できる．

同様に階層的に配置された属性の間の関係も，プロパティ階層を表す場合やクラスとプロパティを表す場合などさまざまである．表 6 では，階層的に配置された属性はプロパティ階層を表す．一方，表 7 では，階層的に配置された属性はクラスとそのクラスのインスタンスが持つプロパティを表している．

Partition Label クラスに属する表では，図 18 のように，表の最上部にだけ属性が記述され，ラベルで分割された各部分には属性の記述がないことが多い．[12] の手法では，属性と対応する属性値は連続的に配置されるとして表を解釈するため，そのような場合には正しく属性と属性値の対応を得られない．本研究の提案手法では，離れて配置された属性と属性値・ラベルについて，それらが関連することを解釈として与えることができる．しかし 4.2 節で述べたように，表を分割するラベルと同じ構造で，注釈などの異なる種類のデータが記述されている場合には，その表における構造と関係の正しい対応は得られない．

Over-expanded label クラスや Combination クラスに属する表では，対応する属性と属性値が離れた表や，階層的な属性の記述を含む表が多いため，[12] の手法は良い結果を得ていない．また同じデータを持つセルを結合して，複数の行や列にまたがるセルで表現されることが多い．表 9 は表 6 において，同じデータをもつ隣接するセルを結合したものである．本研究の手法では隣接する幅の異なるセルには異なる種類のデータが記述されているという考えに基づいて表構造の一般化を行う．そのため，表 9 のように隣接する幅の異なるセルに同じ種類のデータが記述された表で，繰り返し構造で関係を表現できず，本研究の手法は良い結果が得られなかった．

表 9: 同じデータが記述されたセルが結合された表

<table>
<tr><td rowspan="2"></td><td colspan="3">Storage</td><td colspan="2">Software</td></tr>
<tr><td>Hard Drive</td><td>DVD ROM/RW</td><td>Zip</td><td>OS</td><td>Utility</td></tr>
<tr><td>Dimension8300</td><td>80GB</td><td rowspan="2">DVD ROM</td><td rowspan="2">250MB</td><td rowspan="2">Win XP</td><td>Premiere</td></tr>
<tr><td>G310 Series</td><td>40 GB</td><td rowspan="2">Acrobat</td></tr>
<tr><td>PCV RS220</td><td>60 GB</td><td>12x DVD-RW</td><td>None</td><td>Win 2000</td></tr>
</table>

以上の結果から，[12] で提案された，前もって与えた表の認知モデルやデータの種類・配置の特徴に基づく自動的なアプローチでは，属性をプロパティと解釈するかプロパティ値と解釈するかなど，表中の語の意味によって適切な解釈が決まる場合に正しい関係が獲得できないことがわかる．一方，本研究の手法は，人手によってデータ間の関係を記述して与えるため，データの意味によって適切な解釈が決まる場合にも，正しい関係が取得できる．これにより，属性と属性値の対応を得るだけでなく，より詳細に関係を獲得することができる．そのため，本研究の手法はプロパティ-プロパティ値・クラス階層・プロパティ階層など，RDF/OWL でのオントロジー記述に必要な関係を得るのに有効であると考えられる．しかし同じデータを持つ隣接するセルを結合した表など，同じ特徴を持つセルの単純な繰り返し構造で表現できない表では，表構造が一般化できないため，本研究の手法では関係が獲得されない．このような場合には，前処理として，セルの位置やデータの特徴などに基づいて，結合されたセルを分割するという方法が考えられる．

第6章　関連研究

第3章で述べた表構造の形式化・一般化のための表現は，本研究で提案する手法において非常に重要な役割を占めている．このような表構造の形式的表現は，従来の研究においてもさまざまな種類のものが提案されてきた．その目的の一つとして，すでに存在する紙に印刷されたフォームをスキャンしたあと，構造化された情報を獲得して大量に存在する紙のデータを電子化するため，論理的な構造を解析することが挙げられる．また逆に，蓄えられた電子的なデータを，表やフォームの形式で人間にわかりやすいように出力することを目的としたものもある．

以下に従来の研究で提案されている表の形式的表現を挙げる．

[15] では，表の構造を大域的な構造と局所的な構造の2段階に分けて表現する．どちらの構造も，与えられたルールに従って，セルの特定の配置をツリー構造に変換していく．紙の上に印刷されたフォームをスキャンし，表中に記述された情報を得ることを目的としている．

文書レイアウトの表現として用いられる XY-Tree[16] を表の表現に応用する手法も提案されている [17]．XY-Tree を用いた表現では，表を再帰的に分割し，分割の結果を木構造で表現する．木構造のノードは，セルが一つ以上集まったブロックを意味する．また木構造中で親ノードと子ノードは，分割前のブロックと分割されてできたブロックに相当する．根ノードは表全体に相当し，葉ノードはセルに相当する．複数の行や列にまたがる幅の広いセルを基準に分割していくため，さまざまな形状のセルが多く含まれるフォームに適しているとしている．

[18] では，TFML という XML に基づく言語によるフォームの表現が提案されている．フォームをセルの隣接関係に基づいてグラフで表現される．グラフのノードは対応するセルに値が記入されているかどうかや，セルの役割に応じて区別され，グラフに対応する文法を用いて表の構造を解析する．XML やグラフに対応した文法を用いることで，一貫性を保ちながらフォームを改変するのが容易になるとしている．グラフを用いた表の表現は，[19] においても提案されている．セルの隣接関係に関しては，本研究と同じ表現を用いている．

[20] では，これまでの表を扱う研究でのさまざまな表の表現や，目的とするアプリケーションについてまとめられている．

本研究で扱ったような，Web 上の表からの情報獲得を行う際には，HTML から表を発見するための手法も重要になる．なぜなら，HTML 中の TABLE タグは Web ページのレイアウトのために用いられることも多く，必ずしも表形式となったデータを意味しないからである．手法としては HTML タグによる構造に関するヒューリスティクス [9] や，機械学習を用いるもの [21, 22] がある．[9] では，隣接するセル同士でのデータの類似度や，セルに記述されたデータの種類に基づいて，表かどうかを判断する．[21, 22] では特徴として TABLE タグによるレイアウトの特徴や表の内容を用いながら，決定木や SVM によって分類を行う．本研究では基本的にすでに文書から抽出された表を対象にするため，HTML 中の表を扱うためにはここで述べられているような手法を前処理として用いる必要がある．

表からの情報抽出に関する従来の研究では，表を解釈するために，与えられた表を典型的な表構造のいずれかに帰着するもの [9, 23, 10] や，表の認知モデル [11] を利用するもの [12] がある．またラッパーの学習を用いる手法も提案されている [24]．より詳細な関係を獲得するために，対象ドメインの知識ベースを用いて表中のデータの種類を判定するもの [13, 14] もある．

[9] では，HTML から表を検出し，属性と属性値の対応を得る手法を提案している．連続するセル中に記述されたデータの類似度によって，表が縦向きか横向きかを判断する．属性と属性値は隣接しているものと考え，対応関係は位置関係から判断する．複数の行や列にまたがるセルに階層的に属性が記述されている場合にも，隣り合うセルの関係を順に獲得していくことで，複数の属性の関連も獲得できる．しかし表の解析のアルゴリズムは単純なものであり，対応できる構造は限られたものである．また表からの属性と属性値の獲得に関する評価結果は示されていない．

[23] では，表から階層構造を持つデータを獲得する手法を提案している．しかし，表のどの部分にどのようなデータが記述されているかの解釈については，表の内容によって判断するのではなく，前もって想定した表構造に従って行う．例えばもっとも左の列にキーが記述されるなどの仮定を置いているため，有効な結果を得られる表構造が限られる．表から階層構造を持つデータを得ることで，より高度なクエリーや，HTML から XML への変換などの利用が可能になるとしている．

[10] では，表から属性と属性値の対応を得ることにより，表に記述された情

報をデータベースの形式に変換する手法が提案されている．表に記述された情報のセマンティクスを得るための知識ベースが利用される．知識ベースを用いて表の各部分に記述された情報の種類を判断した上で，想定した複数の表構造のいずれに一致するかを決定する．また表のレイアウトのクラスについて，[12] でも用いられている分類を提案している．

[24] では，ラッパの学習を用いて，表からの情報抽出を行う手法が提案されている．学習のための例が与えられ，HTML のタグやタグのコンテンツを利用してルールを学習する．実験結果として数少ないサンプルで学習が行えていることが示されているが，異なるサイトでタグの利用の特徴が異なる場合には有効ではない．

表からのオントロジー獲得を扱うものとしては，[12, 13, 14] がある．

第 5 章で述べたように，[12] では，表の認知モデル [11] に基づいて表を解釈する．表を属性が記述された領域と，属性に対応する値が記述された領域に分割し，位置関係からそれらの対応を得ることで表を解釈する．属性は対応する値が記述された領域の上や左に連続する領域に配置されると仮定している．表の領域の分割と各領域が属性を表すか値を表すかの決定は，セル中のデータの種類 (文字列・数値・日付など) と，その配置によって決定される．結果はフレームとして得られる．

[25] では，同じ種類の情報を記述した表を，一つの大きな表として統合する手法を提案している．表の構造の解析には，想定する構造のタイプをいくつか与えておき，EM アルゴリズムによってどのタイプに属するか判断する．複数の表の統合のために，語の共起に基づく表のクラスタリングや，語の類似度に基づく属性のクラスタリングを用いる．

[13] では，表のスキーマのマッピングを獲得することを目的とする．属性と属性値の組み合わせの獲得には，[23] で提案されている方法を用いる．また，表に記述されたデータを与えたスキーマにあわせて抽出するために，対象となるドメインに特化したラッパを用いる．

[14] では，人が最初に小さな与えたオントロジーに対して，表から得られた情報を加えて拡張を行っていく．また異なる表から得られた関係を，すでに得られている記述を利用して，一つのオントロジーに対して付け加えていく．表からの関係の獲得には，属性間の関数従属性やドメインの知識ベース・表中の値の類似度などが用いられる．例として地理情報のオントロジーを，2 つの異

なる表から得られた情報を用いて拡張していく処理が説明されている．しかし，オントロジー獲得の大まかな枠組みを提案したものであり，自動的にオントロジーを獲得するための処理を説明したものではない．

図 19: Excel 上に実装したインターフェース

第 7 章　システムの実装

本研究で提案する手法では，表中のデータ間の関係を RDF ステートメントにより人手で記述する必要がある．しかし，RDF ステートメントの記述はやや煩雑である．そこで容易に提案手法を実行できるようにするため，提案手法を Excel 上で実行できるシステムを構築した．

システムは Excel のアドインとして実行され，Excel のメニューに登録される．システムを利用するには，Excel のメニューから図 19 のダイアログを呼び出す．

図 19 のダイアログはエピソードの入力と，処理の実行のためのインターフェースとなっている．上部の 3 つのコンボボックスから RDF ステートメントを入力し，追加ボタンで下のリストボックスに加える．3 つのコンボボックスには，それぞれ RDF ステートメントの主語・述語・目的語を入力する．主語・述語・目的語には，Excel のシート上の語を指定して入力することができる．また，表中に現れない暗黙的に表現されるプロパティや，匿名リソースなどはコンボボックスに直接タイプすることで入力できる．

リストボックスに表示されている RDF ステートメントの集合が処理に用いるエピソードとなる．RDF ステートメントは，簡単のために 3 つの語の組み合わせとして表示される．

指定するエピソードによっては，個体を特定するキーとなるプロパティを指定する必要がある．これは一般化表構造で表現される繰り返し構造が一つの個

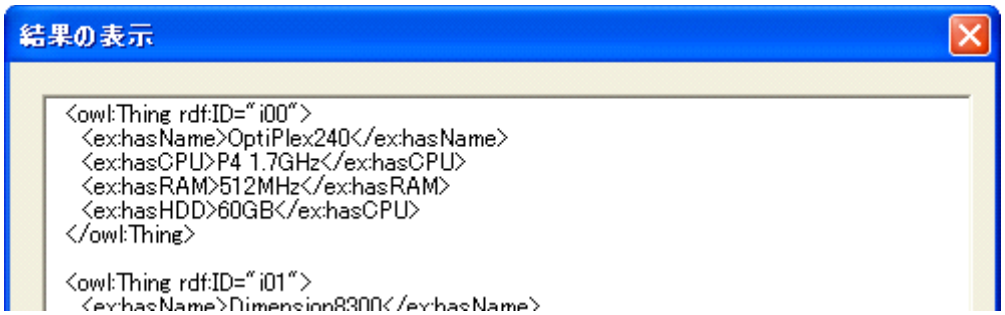

図 20: 実行結果の例

体に関する複数のプロパティに相当するのか，多数の個体に関するプロパティに相当するのかを判断する必要があるためである．一般化表構造の繰り返し構造の中にキーとなるプロパティのプロパティ値が含まれている場合には，その繰り返し構造は複数の個体に関する記述を表していると考えられる．キーを指定せずに実行した場合には，得られた記述はすべて同じ個体に関するものであるとして扱われる．

「出力」ボタンは設定したエピソードをファイルに出力する.「データ変換」ボタンを押すと，開いている Excel ファイルの形式を変換する.「実行」を押すと，設定したエピソードを用いて新たなエピソードの獲得を行う．

「実行」を押して処理が終了した後,「結果の表示」ボタンを押すと図 20 のように，新たに得られたエピソードを見ることができる．結果は RDF/OWL で出力される．これにより，RDF/OWL による記述の煩雑さを軽減できる．

表 3 に対して，図 11 の RDF グラフで表されるエピソードを与えた結果を図 21 に示す．キーとなるプロパティとして，“ex:hasName” を指定した．

ここでは 3 つの個体の記述が得られている．図 6(a) の一般化表構造では，繰り返し構造の中に含まれるのはプロパティとプロパティ値である．そのため，図 6(a) のから得られる記述は全て同じ個体に関する記述であると見なす．一方図 6(b) の一般化表構造では，プロパティ“ex:hasName” のプロパティ値が記述されたボックスに相当する “Name” とラベルの付けられたボックスが，繰り返し構造の中にある．そのため，この一般化表構造から得られる記述は複数の異なる個体に関するものであるとみなす．以上を表現するため，個体の識別に番号を付けた rdf:ID を適当に与えている．

図 21 では，各個体について 4 つのプロパティの値が記述されている．このうち “ex:hasName” のみが人手で指定されたプロパティである．それ以外の

```
<rdf:Description rdf:ID="i00">
  <ex:hasName>OptiPlex240</ex:hasName>
  <ex:hasCPU>P4 1.7GHz</ex:hasCPU>
  <ex:hasRAM>512MHz</ex:hasRAM>
  <ex:hasHDD>60GB</ex:hasCPU>
</rdf:Description>

<rdf:Description rdf:ID="i01">
  <ex:hasName>Dimension8300</ex:hasName>
  <ex:hasCPU>P4 2.4GHz</ex:hasCPU>
  <ex:hasRAM>256MHz</ex:hasRAM>
  <ex:hasHDD>80GB</ex:hasCPU>
</rdf:Description>

<rdf:Description rdf:ID="i02">
  <ex:hasName>Pavilion505</ex:hasName>
  <ex:hasCPU>Cel 2.4GHz</ex:hasCPU>
  <ex:hasRAM>256MHz</ex:hasRAM>
  <ex:hasHDD>40GB</ex:hasCPU>
</rdf:Description>
```

図 21: 出力された関係の記述

“ex:hasCPU”, “ex:hasRAM”, “ex:hasHDD” は，表中からプロパティとして得られた “CPU”, “RAM”, “HDD” という 3 つのプロパティに，名前空間 “ex:” とプロパティであることをわかりやすくするための “has” をつけて作成している．

第8章　おわりに

本研究では，表形式のデータから，オントロジーを獲得する手法を提案した．オントロジーの構築は，関係の定義の難しさや記述の煩雑さなどの理由により非常にコストがかかる．そこで人間にとって直感的に作成しやすく，またすでに大量に蓄積されている表形式のデータの構造を利用してオントロジーを獲得することはセマンティック Web 実現のために有益である．

表中のデータ間の関係は，表構造によって表現されていると考えられる．そこで表ごとに特定の表構造がどのような関係を表現しているかということを人間が解釈して与えることにより，表中のデータ間の関係を得ることができる．ここで解釈が与えられた構造を，その構造が表す意味が変わらないように一般化することにより，より多くのデータ間の関係が獲得できる．本研究では，セルの隣接関係や繰り返し構造に注目して表構造の形式化・一般化を行う．一般化された表構造が表中で一致する箇所を探すことで，データ間の関係を獲得できる．

本研究の貢献は以下の通りである．

表に応じた構造の解釈　表ごとに表中の一部のデータ間の関係を人手で記述して与えることにより，データの意味まで考慮した，それぞれの表に応じた構造の解釈が可能である．これにより従来の研究では得られなかった，そのまま RDF/OWL の記述として利用できるより詳細な関係が得られる．

様々なドメインへの適用　本研究で提案する手法では，人手で与えられる構造の解釈に従って表中のデータ間の関係を得る．この構造の解釈を与えるためのコストは小さく，また表中のデータの意味を理解するための知識ベースを必要としないために，様々なドメインに容易に適用することができる．

さらに，提案手法の有用性を確認し，問題点を明らかにするために，[12] で提案されている手法と比較した．その結果，[12] で提案された，前もって与えた表の認知モデルやデータの種類・配置の特徴に基づく自動的なアプローチでは，属性をプロパティと解釈するかプロパティ値と解釈するかなど，表中の語の意味によって適切な解釈が決まる場合に正しい関係が獲得できなかった．一方，本研究の手法は，人手によってデータ間の関係を記述して与えるため，データの意味によって適切な解釈が決まる場合にも，正しい関係が取得できる．これにより，属性と属性値の対応といった簡単な関係を得るだけでなく，RDF/OWL でのオントロジー記述に必要とされるより詳細な関係を獲得することができた．

また多くの同じ種類の情報が同じ構造の繰り返しで記述されている場合には，少ないコストで表中に記述された情報の RDF/OWL によるオントロジーとしての記述を多数得られるため，セマンティック Web 実現のためのコンテンツの蓄積に有益であると考えられる．

一方で一つの表の中でも同じ表構造が異なる関係を表す場合があり，その場合には誤ったデータが得られた．また，同じデータを持つ隣接するセルを結合した表など，同じ特徴を持つセルの単純な繰り返し構造で表現できない表では，表構造が一般化できないため，本研究の手法ではうまく関係が獲得できない．

また，本研究の手法を実装したシステムを Excel のアドオンとして構築した．これにより，解釈を与える際の RDF ステートメントによる関係の記述の煩雑さを解消できる．このシステムを用いると，GUI を用いてセルを選択したりプロパティ名を入力することで，容易に表の解釈を与えることができる．本研究の手法では，RDF/OWL による表中のデータの関係を直接得られるため，出力結果はそのままオントロジーの記述として利用することができる．

謝辞

本研究を行う機会と環境を与えて下さり，熱心にご指導していただいた石田亨教授に深謝いたします．そして日頃より有益なご助言をくださる石田研究室の皆様に心より感謝いたします．

参考文献

[1] Handschuh, S. and Staab, S.: Authoring and Annotation of Web Pages in CREAM, *11th International World Wide Web Conference*, pp. 462–473 (2002).

[2] Handschuh, S., Staab, S. and Volz, R.: On Deep Annotation, *12th International World Wide Web Conference*, pp. 431–438 (2003).

[3] Kahan, J., Koivunen, M., Prud'Hommeaux, E. and Swick, R.: Annotea: An Open RDF Infrastructure for Shared Web Annotations, *10th World Wide Web Conference*, pp. 623–632 (2001).

[4] Staab, S., Angele, J., Decker, S., Erdmann, M., Hotho, A., Maedche, A., Schnurr, H.-P., Studerand, R. and Sure, Y.: Semantic community Web portals, *9th World Wide Web Conference*, pp. 473–491 (2000).

[5] Maedche, A.: *Ontology Learning for the Semantic Web*, Kluwer Academic Publishers (2002).

[6] Ashish, N. and Knoblock, C.: Wrapper Generation for Semi-Structured Internet Source, *ACM SIGMOD Records*, Vol. 26–4, pp. 8–15 (1997).

[7] Cohen, W. and Fan, W.: Learning Page-independent Heuristics for Extracting Data from Web Pages, *8th World Wide Web Conference*, pp. 1641–1652 (1999).

[8] Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S. Tomkins, A., Tomlin, J. and Zien, J.: Semtag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation, *12th World Wide Web Conference*, pp. 178–186 (2003).

[9] Chen, H., Tsai, S. and Tsai, J.: Mining Tables from Large Scale HTML Texts, *18th International Conference Computational Linguistics*, pp. 166–172 (2000).

[10] Wang, H., Wu, S., Wang, I., Sung, C., Hsu, W. and Shih, W.: Semantic search on Internet Tabular Information Extraction for Answering Queries, *9th International Conference on Information and Knowledge Management*, pp. 243–249 (2000).

[11] Hurst, M.: Layout and Language: Beyond Simple Text for Information

Interaction - Modelling the Table, *2nd International Conference on Multimodal Interfaces*, pp. 243–249 (1999).

[12] Pivk, A., Cimiano, P. and Sure, Y.: From Tables to Frames, *3rd International Semantic Web Conference*, pp. 166–181 (2004).

[13] Embley, D., Tao, C. and Liddle, S.: Automatically Extracting Ontologically Specified Data from HTML Tables with Unknown Structure, *21th International Conferenceon Conceptual Modeling*, pp. 322–337 (2002).

[14] Tijerino, Y., Embley, D., Lonsdale, D. and Nagy, G.: Ontology Generation from Tables, *4th International Conference on Web Information Systems Engineering*, pp. 242–252 (2003).

[15] Watanabe, T., Luo, Q. and Sugie, N.: Layout Recognition of Multi Kinds of Table-Form Document, *IEEE Transactions Pattern Analysis and Machine IntelligenceComputer Design*, Vol. 17–4, pp. 432–435 (1995).

[16] Nagy, G. and Seth, S.: Hierarchical Representation of Optically Scanned Documents, *7th International Conference on Pattern Recognition*, pp. 347–349 (1984).

[17] Duygulu, P. and Ataly, V.: A Hierarchical Representation of From Documents for Identification and Retrieval, *IS&T/SPIE's 12th International Symposium on Electronic Imaging-Document Recognition and Retrieval VII*, pp. 128–139 (2000).

[18] A.Amano and N.Asada: Complex Table Form Analysis Using Graph Grammar, *5th International Workshop on Document Analysis Systems*, pp. 283–286 (2002).

[19] Rahgozar, A. and Cooperman, R.: A Graph-based Table Recognition System, *IS&T/SPIE's 12th International Symposium on Electronic Imaging-Document Recognition III*, pp. 192–203 (1996).

[20] Lopresti, D. and Nagy, G.: *A Tabular Survey of Automated Table Processing, Lecture Note in Computer Science 1941*, Springer–Verlag, pp. 93–120 (2000).

[21] Wang, Y. and Hu, J.: A Machine Learning Based Approach for Table Detection on The Web, *11th International World Wide Web Conference*, pp. 242–250 (2002).

[22] Wang, Y. and Hu, J.: Detecting Tables in HTML Documents, *5th International Workshop on Document Analysis Systems V*, pp. 249–260 (2002).

[23] Lim, S. and Ng, Y.: An Automated Approach for Retrieving Hierarchical Data from HTML Tables, *8th International Conference on Information and Knowledge Management*, pp. 466–474 (1999).

[24] Cohen, W., Hurst, M. and Jensen, L.: A Flexible Learning System for Wrapping Tables and Lists in HTML Documents, *11th World Wide Web Conference*, pp. 232–241 (2002).

[25] Yoshida, M., Torisawa, K. and Tsujii, J.: Extracting Ontologies from World Wide Web via HTML tables, *Pacific Association for Computational Linguistics*, pp. 332–341 (2001).