

クリックابلWeb KB

杉本 正勝

東京情報大学 総合情報学部 情報システム学科

〒265-8501 千葉県若葉区谷当町1200-2

E-mail: sugimoto@rsch.tuis.ac.jp

あらまし クリックابلWeb KBのイメージとそのプロトタイプを紹介する。Webアプリ構築法としてAjax-JSPの組合せを採用した。Web KBのプロトタイプはJavaで記述されたProlog処理系で実現されている。PrologプログラミングにはCSPP: Case Structured Prolog Programmingと名付けた技術を用いた。クリックابلWeb KBのイメージを説明した後、マンマシンインタフェース部の検討、Web KB部の検討、両者をつなぐAjax-JSP連携部について述べる。クリックابلWeb KBは、eラーニング、チャットなど応用が考えられる。Web KBはWeb上に置かれた知識ベースであり、各種のWebアプリを実現する基盤として多くの可能性がある。例えば、ソフトウェア工学の立場からはWebアプリ設計・構築用レポジトリとしての用途が考えられる。

キーワード 知識ベース、ヒューマンインタフェース、Web 2.0、Ajax、eラーニング

Clickable Web KB

Masakatsu SUGIMOTO

Department of Information Systems

Tokyo University of Information Sciences

Abstract An image of Clickable Web KB (Knowledge Base) is discussed and a prototype of it is presented. The prototype has been implemented as an Ajax-JSP system. The prototype is realized through Prolog written in Java. For the Prolog programming, a programming technology named CSPP: Case Structured Prolog Programming has been utilized. The authors are interested in such application as e-learning and chat through the Clickable Web KB. The Web KB is a literally a Knowledge Base on Web. So, there are potentially various applications of it. From the view of Software Engineering, for example, the Web KB would be a good candidate for the repository for designing and implementing Web application systems.

Key words Knowledge Base, Human Interface, Web 2.0, Ajax, e-learning

1. はじめに

本研究の目的は、知的なコンピュータシステムの実現性と高度なヒューマン・コンピュータ・インタフェース(HCI)の実現性の検討である。ムーアの法則に代表されるようなコンピュータの処理能力向上を知的処理にどのように活用すべきか？また、どんな新しいHCIが実現可能か？の研究である。

この目的のために、関連の技術進歩を取り込む形で、近未来的に役立つ知的でHCIのよいシステムのイメージ作りとプロトタイプ作りを行った。

知的なシステム作りについては、第五世代プロジェクト以来の研究テーマであり、現在でも各所で長期的展望の下で研究が進められている[3], [4]。ここでは、「近い時期に使える」を重視し、まずはRDBを超えるデータ格納場所として、PKB: Predicate KBに注目しこれをCase Structured Prolog Programming手法で実現した。

特にPKBをネットで利用できる枠組みを検討し、Web上で動く知識ベース（KB）であるので、Web KBと名付けた。

よりよいHCIの実現では、WebアプリのHCI向上に貢献されているAjax技術に着目した。AjaxでどれだけのHCIが実現できるか、Web KBを含めて総合的にどんな新しいアプリのイメージができるか検討した [5], [6], [7], [8], [9]。

検討を進める中で、Web KBを優れたHCIでアクセスするアプリのイメージを固め、このアプリにクリックابلWeb KB (Clickable Web KB) と名付けた。

本報告では、クリックابلWeb KBのイメージとそのプロトタイプを紹介する。Webアプリ構築法として、Ajax - JSP組合せを採用した。Web KBはJavaで記述されたProlog処理系で実現されている。

Web KBはWeb上に置かれた知識ベースであり、各種のWebアプリを実現する基盤として、多くの可能性が考えられる。

2. クリックابلWeb KB

新しい概念であるクリックابلWeb KBを利用者の観点で説明する。利用者がクリックابلWeb KBを使うとき、クライアントPC画面には多くの隠れた（見えない）「オブジェクト」（関心事の対象）が置かれている。隠れているので通常、画面は無地でもよいが、マウス操作の便利のためにいくつかの線が引かれている。この画面のことを「表示なしのディスプレイ」と呼んでいる。一例を図1示す。この画面はスクリーンセーバ画面のような位置づけである。

表示なしのディスプレイ

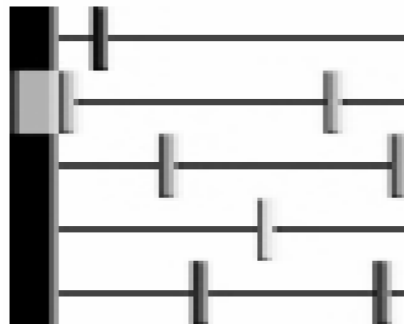


図1 表示なしディスプレイ画面

次の2つの利用形態を検討した。

クリックابلWeb KBインタフェース I :

- (1) この画面をマウスでブラウズすると、ツールチップ (Tooltip、ツールチップと表記することもある) 機能で次々にオブジェクト名を見ることができる。

ツールチップはマウスが置かれた位置に対応した情報を即時に画面に表示する機能。

図2はマウス動作でオブジェクト名（この例では、分散処理方式）が現れたときのスナップショット。

- (2) あるオブジェクト名が表示されているときに、ダブルクリックすると、利用者が『そのオブジェクトの中身をもっと知りたい』と意思表示したことになる。

①そのオブジェクトの属性（プロパティ）に対応するボタン（属性ボタン）がその属性の数だけ表示される。図3は属性ボタンが表示されたところを示す。

②ユーザがその属性ボタンを押すと、値（属性値）が表示される

JavaScriptでは、画面上に画像が表示されているときに、マウス位置を指定しダブルクリックしたとき、位置に応じた処理につなげる機能をクリックابلマップ (Clickable map) と呼んでいる。この言葉からクリックابلWeb KBと名付けた。

表示なしのディスプレイ

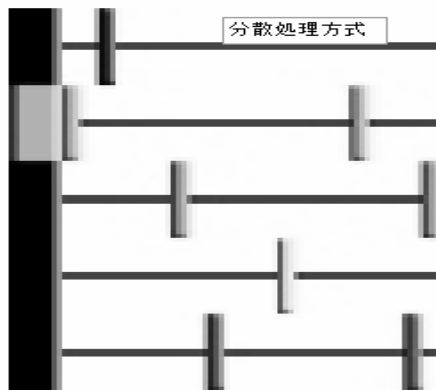


図2 マウス動作で（ツールチップ機能で）オブジェクト名が現れる

表示なしのディスプレイ

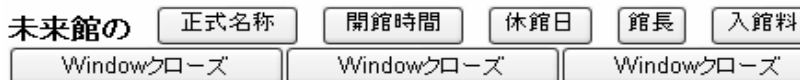
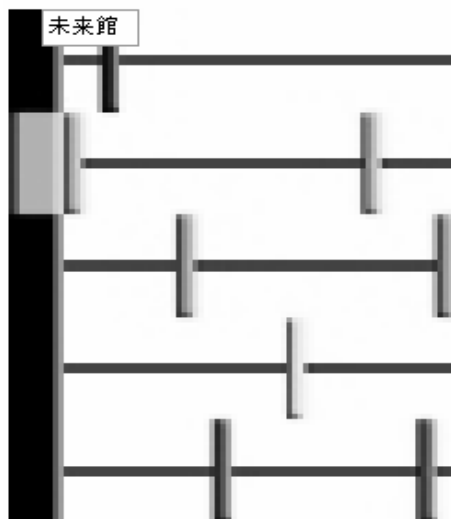


図3 オブジェクトの属性ボタン（複数）の表示

クリックابلWeb KBインタフェースⅡ：

クリックابلWeb KBインタフェースⅠとの差は、(2)のステップでユーザが『そのオブジェクトの中身を知りたい』と意思表示しても、対応の属性ボタンは表示されない。どんな属性があるかを知るには、ツールチップ機能を使う。利用者は属性名スキャン位置でマウスを移動することにより、そのオブジェクトの属性名を知り、さらに値の表示のためにはダブルクリック動作をする。

表示なしのディスプレイ

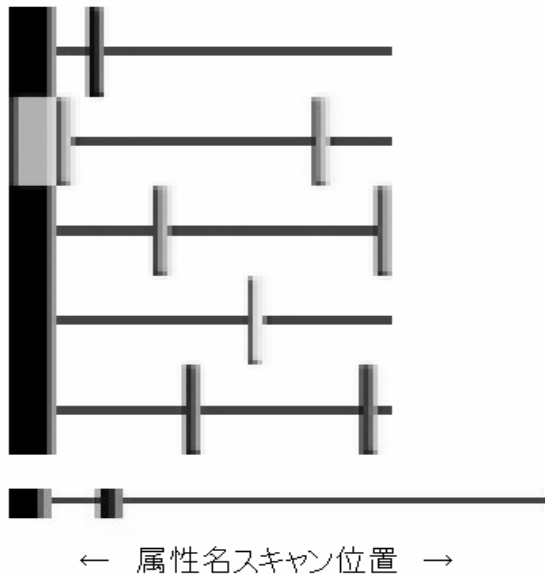


図4 属性名表示もツールティップ型（インタフェースⅡ）

ここで、対象とするオブジェクト集合（KBの中身）はクライアントPCにあるのではなく、Webサーバ上にある。

この利用者インタフェースに注目したきっかけは、Web 2.0のさきがけとしてのGoogle Earthである。『サーバ側におかれた地球の地理情報を、マウス移動の速度で検索できるのであれば、サーバ側に置かれたデータまたは知識を2次元マップとしてマウスで検索することも可能ではないか？ 実地に試して見よう』がクリックابلWeb KB研究の原点である。

また、「知識のマップ」を検索するときに、『余分な情報に目を奪われない』、『画面表示のセキュリティの面や使い易さの点から、マウスが置かれ利用者が見たいと意思表示したときのみ表示する』、『通常は画面上に情報は表示されていない』を実現すべく「表示なしのディスプレイ画面」のインタフェースを検討した。

3. HCIの検討

Web 2.0を実現する基本技術としてのAjaxを用いて、興味深いHCIを実現する試みが各所で行われている。AjaxでどれだけのHCIが実現できるかについての検討した結果、まずはマウスの動きに追従して、画面が変化するHCIに注目した。JavaScriptで実現するインタフェースとしてツールティップとクリックابلマップに注目した。

ツールティップはマウスが置かれた位置に対応した情報を即時に画面に表示する。クリックابلマップはマウス位置でダブルクリックすることで関連動作につなげる機能である。この2つの機能は、通常のWebブラウザに備わっている。

Web上にある「知識のマップ」をクライアント画面に表示し、ツールティップとクリックابلマップの技術で知識を検索および処理する。これがクリックابلWeb KBである。

従来のWebアプリ作成技術、例えばJ2EE体系では、サーバ側の技術に注目が集まりすぎ、クライアント側のHCI向上は手薄であったと考えられる。また、Webアプリケーション実現の必須要素であるデータベースは利用者から見てアプリの奥（バックエンド）に配置されたので、ツールティップやクリックابلマップ形式のデータベース検索は試みられていなかったと推測する。

4. Web KB

Web上の知識ベースについて、まずは、現在のRDBよりも柔軟なデータの置き場所を利用できるようにする。この目的にかねてより試作を進めていた述語型知識ベース (Predicate Knowledge-base、PKB) を利用した。PKBへのQueryインタフェースを表3に示す。このインタフェースをWeb KBにも採用した。

知識登録は、『OのAはV。』の形式で行う。Oはオブジェクト名、Aは属性 (プロパティ)、Vは属性値である。

概念的にはRDBの各データ項目にデータベース名 (オブジェクト名) とそのフィールド名 (属性名) を付加したと考えてよい。RDBでは属性名をつけないのでフィールド位置とフィールド数を固定する。述語型知識ベースでは、属性名をつけ、フィールド数は任意とできる。PKBでRDBのデータを表現できることは自明である [1]。

O, A, Vの三つ組みのみをインタフェースとせず、Oの、Aは、V。のように助詞や句点がある理由は、ユーザにとっての分かり易さである。ユーザが直接このインタフェースを利用することも想定した。Queryに○、△、□という記号を変数の用途に使った理由も同一である。携帯機からユーザがQuery文を直接入力することも想定してこのインタフェースを定めた。

RDBとPKBを比較すると表1のようになる。

Web 2. 0という言葉が使われるときにロングテール (Long tail) という言葉が使われることが多い。ロングテールを実現する鍵は、ニッチな項目のデータを容易に扱えることだと、筆者は考えている。RDBはその「固さ」からどうしても「マイナー項目切捨て型」とならざるをえなかったが、Web 2. 0の時代にはPKBのような柔軟なデータ格納場所が必要であると考えられる。

RDBとPKB比較(1)

| RDB | | | | | | PKB |
|-----|--------|----|------|------|---------|--|
| 顧客名 | 顧客名の読み | 住所 | 郵便番号 | 電話番号 | メールアドレス | (例) オブジェクト: 日本科学未来館: 開館時間: 10:00~17:00 休館日: 毎週火曜日 館長: 毛利 衛 入館料: 大人500円 18歳以下200円 住所: 東京都江東区青柳2丁目41番地 郵便番号: 135-0064 最寄駅: テレコムセンター(新交通ゆりかもめ) 船の科学館駅(新交通ゆりかもめ) 東京テレポート駅(りんかい線、徒歩15分) 展示品: 4004マイクロプロセッサ・チップ(嶋 正利) アシモ・ロボット レスキューロボット・フィールド インターネット物理モデル ナノテクノロジー マイクロマシン ... |
| | | | | | | 属性: 開館時間、休館日、館長、入場料、... |

RDB: Relational DB
PKB: Predicate Knowledge Base

RDBとPKB比較(2)

| | RDB | PKB |
|---------------|--------------------|--------------------|
| フィールド数 | 固定 | 任意 |
| フィールドの追加 | DB管理 | 自由 |
| フィールドのデータタイプ | 設計時に定める | 設定必要なし |
| 文字列データの長さ | 設計時に最大長を定める | 制限なし |
| フィールドデータの階層構造 | 不可 | 可能 |
| 特徴 | マイナー項目切捨て型 固いDB | 「ロングテール型」 柔軟なDB |
| 原理的機能 | — | RDB < PKB |
| 適用アプリ | 従来型 | Web 2. 0等 |

表1 RDBとPKB比較

5. クリックブルWeb KBのプロトタイプ

クライアントPCとWeb KBを連携させてクリックブル Web KBを試作した。プロトタイプ作りでは次のように考えた。Web KBはネット上に置かれる。ネット上でのマシン独立性・OS独立性、ソフトの作り易さの点からJava言語を選択した。また、KBの論理は、筆者が長年利用してきたPrologを利用したいと考えた [11]。Web上で動くPrologを実現するには、Javaで記述されたProlog処理系を利用する必要があった [13]。さらに、Web KBプロトタイプ実現では、CSPP: Case Structured Prolog Programming (Case構造プログラミング) 技術を採用した。CSPPとは、Case構造とSnip機能を多用してプログラムを書く技術である [12, 13]。これにより、Prolog言語でプログラムを書くときの「制御論理の厄介さ」の問題を克服し、保守し易いプログラムを作成できると実感している。

| ／ --- ＼ | 分 類 | 入 力 文 | 応 答 | 備 考 |
|---------------|--------|-----------|-----------------------------------|------------------------|
| 登録文 | | ～1の～2は～3。 | --- is asserted. --- is there. | 正常に登録された。 すでに登録済み。 |
| 問合せ文 | ?付 | ～1の～2は? | ～1の～2は～3。 | |
| | ?付 | ○は? | オブジェクト一覧 | オブジェクト一覧 |
| | ?付 | △は? | 属性一覧 | 属性一覧 |
| | ?付 | □は? | 値一覧 | 値一覧 |
| | Q1 | ○の～1は～2。 | ～3の～1は～2。 | オブジェクト問合せ |
| | Q1 | ～1の△は～2。 | ～1の～3は～2。 | 属性問合せ |
| | Q1 | ～1の～2は□。 | ～3は～1の～2。 | 値問合せ |
| | Q2 | ○の△は～。 | ～1の～2は～。 | オブジェクト・属性問合せ ～は○の△? |
| | Q2 | ○の～は□。 | ～1の～は～2。 | オブジェクト・値問合せ |
| | Q2 | ～の△は□。 | ～の～1は～2。 | 属性・値問合せ ～は何? |
| Q3 | ○の△は□。 | ～1の～2は～3。 | 全知識一覧 | |

表2 Web KBのQueryインタフェース

注1) ○はオブジェクト（主語）変数、△は属性変数、□は値変数

2) ～1、～2、～3は、主語、属性、値の具体値を表現する

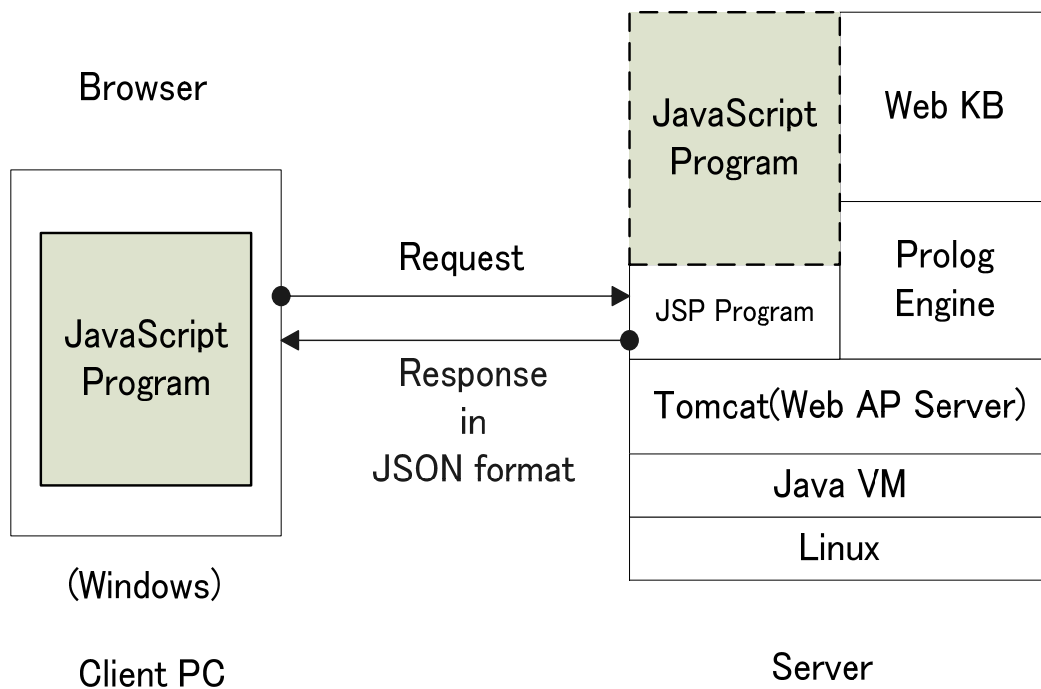


図5 クリックブルWeb KBプロトタイプの構成図

クリックブルWeb KBのプロトタイプ構成図を図5に示す。

クライアントPCとWeb KB連携システムとして、本プロトタイプの特徴は次の通りである。

- ① クライアントのHCI実現にAjax技術を使用した
- ② サーバ側はJSPを用いた [11]
 - 注) 実際のJSPソースプログラムはProlog Engineの初期化などごく小規模になるので、JSPという用語をServletと置き換えてもよい
- ③ Web KBはJavaで書かれたProlog処理系を用いた
- ④ サーバ⇒クライアントのデータ形式にJSON型データを用いた
- ⑤ クライアント側ではサーバ側からのデータをもとに(動的に)属性対応でボタンを表示する。属性ボタンをユーザがクリックすると、属性値が得られる
- ⑥ クライアントとしてはPCのほか、携帯機も利用できる (この場合はテキストでの応答となる)

プロトタイプを実現してクリックブル KBを試用して判明したことは次の通り。

- (1) Ajax関係: JavaScriptを使うことで、HCIをうまく実現できた
- (2) JSPを使うメリット: Javaのマシン独立性とOS独立性を活かせる
- (3) Prologを利用するメリット: Web KBのプロトタイプ作りに適している
 - 特に、CSPPを利用することで、保守性、拡張性のあるPrologプログラムを実現した
- (4) JSONを使うメリット: Prologの関数子 (functor) とJSONとの対応がよい。またJavaScriptでの処理が簡単化できる

6. クリックابلWeb KBの応用途

Web上の知識ベースをユーザがマウスクリック動作で容易に検索や登録・変更が可能となれば、従来実現できていなかった多くの利用途の可能性がある。筆者は次の項目に注目している [14], [15]。

- e-ラーニングへの適用：** 知識項目をWeb KBに登録しておき、「表示なしインタフェース」で学習する。
- 掲示板の利用：** 例えば一つのオブジェクトのある属性を複数人の情報交換用に利用できる。これはチャットにも利用できることを意味する。
- Web KBの応用途：** 順序は逆になるが、単独のWeb KBには多くの可能性があると思う。例えば、Webアプリケーション開発する際の、設計・構築用データを格納するレポジトリとしての利用が考えられる。

7. おわりに

Web上の知識を「マウスクリック」で容易に扱うことを、クリックابلWeb KBと名付けた。クリックابلWeb KBのイメージはGoogle Earthを利用した人には容易に描いてもらえると思う。ここではクリックابلWeb KBのプロトタイプを作成し、使用感を確かめた。また、その先にはどんな課題があるのかを検討した。

今回のプロトタイプのポイントは、「表示なしのディスプレイ画面」とWeb KBであった。プロトタイプ作成の要素は、Ajax, JSP, Javaで書いたサーバ上で動くProlog処理系、Web KB, JSON等であった。

Web上に「整理された知識」を蓄えることの重要性とそれを優れたHCIで扱えることの重要性を再確認できた。

ムーアの法則とギルダーの法則の二大経験則のもとで、クライアントPCやサーバマシンの高速化、ネットワークのブロードバンド化は今後も進む。それに加えて、OSやミドルウェアの進歩、Java処理系の充実に加え、Ajax技術等を融合させるとコンピュータとネットワークの新しい使い方が出てくると予想されている。知的なコンピュータと優れたHCIの組合せの可能性の検討をより深めたい。

参考情報および文献

- [1] 花川賢治、天笠俊之、波多野賢治、宮崎純、植村俊亮：Prologを使ったRDFデータからのユーザ指定による文書構築、情報処理学会研究報告、2005-DD-49 (11)、2005
- [2] 中山幹敏、奥井康弘：標準XML完全解説（上・下）、技術評論社
- [3] 斉藤信男、萩野達也（監修）：セマンティック入門、Ohmsha
- [4] CDL：<http://www.instsec.org/>
- [5] 深津貴之、増井雄一郎、川崎有亮、台場圭一：Ajax、技術評論社
- [6] N. Zakas, J. McPeak, J. Fawcett：Professional Ajax, wrox
- [7] N. Zakas：Professional JavaScript for Web Developers, wrox
- [8] 佐久嶋ひろみ：JavaScript ハッカーズ・プログラミング、メディア・テック出版
- [9] K. Smith：Simplifying Ajax-Style Web Development, pp.98-101, Computer, May 2006
- [10] 近野 睦、高安厚思、戸田和弘、西川 麗、藤村浩士：サーブレット/JSP、Soft Bank
- [11] W. Clocksin, C. Mellish：Programming in Prolog, Fifth Edition, Springer
- [12] The Arity/Prolog Compiler and Interpreter, Arity Corporation
- [13] CKI Prolog, <http://www.csc.liv.ac.uk/~siewwert/programs.html>
- [14] 黒瀬邦夫：ナレッジマネジメントのSE業務への適用、pp.656-661、情報処理、2006.6
- [15] eラーニング白書 2006/2007年度、日本イーラーニングコンソシアム編、東京電機大学出版局