B-037

知識の関係性を試すプログラム環境 LKP

A programming environment for the system to try relationship of knowledge LKP (Lightweight Knowledge Platform)

河合 一夫†‡
Kazuo Kawai

1. はじめに

1.1 背景

インターネットが普及した現時点において，我々は容易に大量の情報を得ることができる．しかし，大量の情報を容易に収集することはインターネットの検索サイトで可能となったが，収集した大量の情報を利用目的に応じた構造とすることは難しい．情報を構造化することは知識として利用する際には必須である．例えば，SWEBOK[1]やPMBOK®[2]に代表される知識体系は多大な労力がかけられ改訂作業が続けられている．また，必ずしも全ての利用者の欲する構造とはなっていない．大量の情報を利用者が必要とする構造に変換し知識として利用可能とすることは，ますます重要性が高まってくると考えられる．

1.2 動機

筆者は，リスクの知識体系を階層的な構造により構築する際，さまざまな視点により知識の構築が必要であることを感じた[3]．リスクに代表される多様性を持った概念に関する情報を体系化し知識として利用可能にするためには，その概念が持つ多様性を許容する知識の体系化が必要である．情報の関係をA→Bで表すとした時，AがBを包摂する，一般化と特殊化，原因と結果，目的と手段，など様々な意味として捉えることが可能である．これらは知識を利用する目的や利用者の意図に依存する．情報を知識として利用可能とするための体系化は一度で出来るものではなく，目的や意図を考慮したものとすることが必要であり，繰り返し試行可能な知識の体系化を支援する環境が必要である．

これらのことより知識の体系化を試行可能な環境の必要性を感じた．本論で述べる環境はこれらの動機に基づいている．知識の関係性に関しては，次節にて論じる．

2. 知識の関係性

2.1 知識体系の構築

ものごとの分類は，生物学における分類学として発展してきた[4]．生物の「目に見える」特徴を捉えることから始まり，器官の機能的な特徴を捉える分類へと変化をしていった．我々が，知識体系を構築する目的は，対象（先の例では生物全体）の在り方を認識，共有するためである．特に，大量の情報を扱う際には，個別の情報を扱うのではなく，体系の中における位置づけを考えることで物事の理解が容易になる．こういったことから，知識を体系化するという作業は，今後ますます重要となる．

2.2 表構造における関係性

我々が普段扱う情報のほとんどは表形式として扱っている．複数の見出しを持った表構造のデータは我々のほとんどの業務において利用可能である．その際，見出し間には暗黙の関係性を仮定している．従って，異なった目的に利用しようとした場合，新たな見出しを付与したり，余分な見出しを削除して新たな表を作成する．表形式を扱うツールにはこれらを容易に実施する機能が付与されている．表形式の持つ関係を容易に扱える環境を作ることは知識体系の構造を容易に作成することにおいて意味がある．

本稿で紹介するプログラム環境は，表構造の情報が持つ暗黙的な情報間の関係性を明示的に扱うためのプログラム環境の構築を支援すること，知識の関係性を様々な点より試行する環境の構築を行うことを支援する．

3. LKP

3.1 概要

LKP(Lightweight Knowledge Platform)[5]は，Java で作成された RDF(Resource Description Framework)[6]を処理するプログラム環境であり，知識構築の試行を支援する．想定している利用例としては，大規模な知識体系を構築するためのプロトタイプや小規模なアプリケーションを想定している．RDF を処理するライブラリとして Jena[7]を利用している．

LKP を用いたアプリケーション例の概要を図 1 に示す．

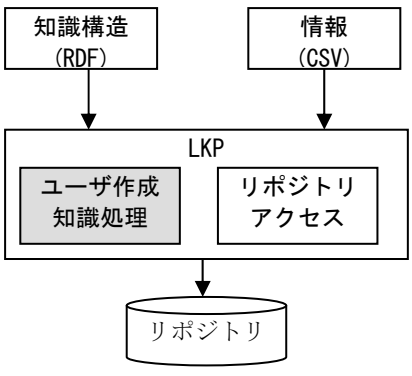

図 1 LKP の概要

2007 年 6 月時点での LKP は基本的な骨格ができあがり，単純なアプリケーション作成に利用可能な状態となっている．

3.2 構造

LKP は，動作を規定するインタフェースと基本的な実装コードを提供し，それらを随時拡張することで必要なアプリケーションを作成する．LKP の全体構造を図 2 に示す．

† （株）ニルソフトウェア

‡ PMI 東京支部 リスクマネジメント研究会

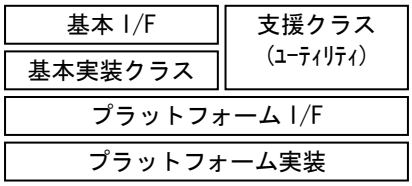

図 2 LKP の全体構造

主要な基本 I/F を以下に示す.

- Action：実行の単位
- Condition：実行条件
- Context：実行結果に伴う状況

情報の処理は，プラットフォームに登録された基本 I/F で定義されている Action インタフェースを実装したクラス群が行う．Action で利用するオブジェクトや Action の実行結果は Context が保持し，Action の実行条件は Contition により決定される．その関係を図 3 に示す.

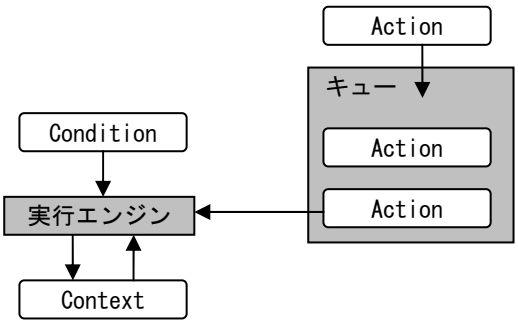

図 3 プラットフォーム内の処理概要

Action インタフェースを実装した情報処理モジュールがプラットフォーム内のキューに登録され，実行エンジンにより処理される．その過程で作成したオブジェクトや利用するオブジェクトは Context を通じて行う．Context には全てのアクションが実行された結果が蓄積される．このように LKP は非常に単純な動作で実行するため，その理解は容易であり，また簡単に拡張することも可能となっている.

4. サンプルアプリケーション

図 1 に示した LKP を利用した簡単なアプリケーションは，CSV 形式で作成された情報を RDF に変換してリポジトリに登録し，検索や更新を可能にするものである．その内容を以下に示す.

(1) 情報は表構造で定義され，データは CSV 形式で表現されている.
(2) 知識の構造は，情報の表構造を定義している見出しの関係を RDF で記述される.
(3) RDF を読み込み，RDF のステートメントとしてリポジトリに登録を行う.
(4) CSV データを RDF で記述されているステートメントの形式にあわせて変換を行いリポジトリに登録する.
(5) 登録されているステートメントを検索，更新する．検索時は，図 1 で示される情報が検索情報として解釈されることとなる.

RDF では，ステートメントと呼ばれる三つ組み（主語，述語，目的語）で情報を有向グラフとして表現する．最初に読み込まれた知識表現のグラフに後から読み込んだ情報を知識表現のグラフに付加したグラフを作成する．この簡単なアプリケーションでは，表形式のデータを RDF のステートメントとして表現することで，表の見出し間にあった暗黙の関係を明示的にすることで，知識として取り出すことが容易となる.

5. おわりに

知識を体系化する必要性から簡単に体系化を試行できるプログラミング環境を作成した．情報間の関係性を明確にし，業務に利用可能としたものが情報システムであり，これまでにも様々な手法が提案され実践されてきている．しかし，今後は情報システムを構築するために必要となる知識の体系化や，組織の形式的な知識を体系化することが必要とされることが予想される．そのためにも知識を利用するための体系化の手法や具体的なシステム構築方法に関する手法が必要とされる．LKP はそのための第 1 歩であり，今後はより複雑で大量の情報を扱うためのプラットフォームの検討が課題であると考えている.

6. 参考文献

[1] IEEE，松本吉弘，ソフトウェアエンジニアリング基礎知識体系 SWEBOK，オーム社，2003
[2] PMI®，プロジェクトマネジメント知識体系ガイド第３版，2004
[3] 河合一夫，“リスクマネジメント知識体系 RiMBOK の構築”，プロジェクトマネジメント学会 2007 年春季大会，2007
[4] 吉田政幸，分類学からの出発，中公新書，1993
[5] LKP，http://sourceforge.jp/projects/lkp/
[6] RDF，http://www.w3.org/RDF/
[7] Jena，http://jena.sourceforge.net/