

会 議 報 告

知識の共有と再利用のワークショップ報告

KBS 研究会 (主査: 小山照夫, WS 担当幹事: 山口高平) の主催のもとに表記のワークショップが, 1992 年 12 月 18 日と 19 日の 2 日間にわたって浜松において開催されたのでその概要を報告する。なお, 筆者らの怠慢で執筆が遅れたこと, また, 原稿をまとめる過程において E-mail を通して議論が進められたこともあり, ここに述べる内容は浜松での議論を正確に反映しているとは限らず, 当日の議論と筆者ら 3 人の意見とが混在していることをあらかじめお断りしておく。

ワークショップの目的

近年, 知識の共有と再利用技術が重要な課題として認識されている。エキスパートシステムにおける知識ベース構築方法論確立のため, また情報化時代における知識情報の蓄積, 継承, 流通, 利用を考えるうえで知識の共有と再利用技術は本質的な役割を担うものと期待されている。本ワークショップは, 知識の共有と再利用について参加者による広い立場からの問題提起を受け, 技術の本質, 適用可能性, 将来性, 問題点について討論することを目的として開催された。

以下の四つの特別講演

1. 知識の再利用とオントロジー: 溝口理一郎 (阪大)
2. 協調型ソフトウェアアーキテクチャ: 本位田真一 (東芝)
3. 知識処理と言語処理の融合としての大規模知識ベース: 横井俊夫 (電子化辞書)
4. 知識の共有と再利用への分散人工知能アプローチ: 西田豊明 (京大)

と参加者全員のポジションペーパーの発表を中心に, オントロジー, ソフトウェア工学, 大規模知識ベース, マルチエージェント, CSCW, 知識ベースの評価などについて議論した。

以下に, 主な議論の内容を紹介する。

1. オントロジー

問題解決には知識が必要である。知識工学という言葉が世に出た当初は, エキスパートシステム=推論エンジン+知識で, 推論エンジニアのほうは汎用なもの

が準備できるので, エキスパートは知識のほうのみを準備すればよいと提唱された。一見単純明快で, これぞまさに救世主とも思われたが, 結局このアプローチもソフトウェア工学が抱えている問題と同じ問題を内包していることがすぐに明らかとなった。人工知能が知能の科学であるならば, 知識工学は文字どおり知識の工学であるべきである。人工知能の研究は知識表現, 推論方式, 機械学習の理論など, 知的なエージェントを人工的に実現するために有用な方法論を提供してくれた。しかし, 肝心の知識そのものについては, 入れ物を提供したのだから, 中身のほうは自分で考えよという態度であり, さまざまな形態で存在する知識を抽出し, 入れ物に適合するように加工し, 組織化する方法論を与えてくれている。このことが具体的な問題解決に人工知能を応用するうえでの最大の障害になっている。

我々が問題解決に使用している知識は千差万別である。数学, 物理, 化学など理学の各分野の知識は長い歴史を経て体系化され現在に至っており, すべての問題解決の基礎となっている。設計, 建築, 機械工学などの工学の各分野においても, 問題に応じた各種方法論が体系化されており, 問題解決の枠組みを提供してくれている。世の中の個別の問題に対してもヒューリスティクスといわれるエキスパートの知識が蓄積されている。このほかにも, 哲学, 倫理学, 社会学, 文学などの人文系の知識も間接的には文化という形で問題解決に影響を与えている。さらには, 常識といわれる知識がある。我々は常識の海の中に生活しており, 具体的に数え上げることが困難な知識である。このような知識を個々の問題解決ごとに個別に記述していたのでは, いつまでたっても知識工学は進展しない。

知識を整理して, 複数のプログラムが共通に使えるようにすることは, これからの必然的な方向である。しかし, 何に使おうとするのかという明確な目的なしに知識を事前に整理して, 記述できるかということは自明ではない。CYC の成否を議論するときにも繰り返し問われる問題である。対象とする問題と関連する知識をかなり限定したうえで, このような議論を進めるべきである。

〔1〕 オントロジーとは

同じものを記述する方法は幾通りもある。対象をど

のように眺めるかを明確にすることは極めて重要なことである。そうしなければ、その記述体系は別の観点から眺めた場合には有効ではなくなる。オントロジーとは本来、我々の認識の仕方とは独立したものの存在形態を意味する言葉であり、この世の成り立ちを記述する根源的な視点のことである。しかし、我々が問題解決のために知識を整理する際に使用する場合には、我々がどのように対象を把握しているかの立場を明確にする意味で使われている。したがって、オントロジーの対立概念である認識論に近い、との見方もできる。確かに、AI システム自身は実世界を人間が認識した結果に基づいて構築されていることを考えれば、オントロジーはむしろ認識論レベルの概念であるという意見は正当である。しかし、AI システムを一つの仮想世界と考えれば、その世界のオントロジーは哲学でいう本来の意味のオントロジーと一致する。我々AI 研究者は仮想世界を構成する基本プリミティブを扱うわけであるので、オントロジーという概念はそれほど違和感がない。しかし、本来の意味と異なる使い方をしていくことの自覚は必要である。

〔2〕 オントロジーの分類

問題解決の立場から、対象を記述することは対象をモデル化することであり、そのためには大きく、問題解決のプロセスを記述する知識と具体的な対象に関する知識が必要になる。通常、前者をタスク知識、後者をドメイン知識と呼んでいる。この分類に対応してタスクオントロジーとドメインオントロジーがある。すべての知識をこの2種類のオントロジーに分類すると、数学などの知識はどこに入るのかという疑問が生ずる。これはタスク、ドメインが何を意味するのか曖昧であるからである。通常、設計、診断、レイアウトなどは問題のタイプを表し、タスクの代表的なものであると考えられる。ドメインというと、タスクを例えば診断に固定した場合に、診断の対象となるもの（例えば、病気の診断であれば患者や医学、プラントの診断であれば対象プラントの特性など）を想像する。このとき、数学や物理の知識はドメインを記述するときの基盤として使われているのでドメイン知識だといわれても、対象固有の知識であるわけではない。多くのドメインで共通に使われる知識である。かつ、数学の知識といっても解き方まで含めて知識といってしまうと、数学の問題として定式化したものの解法はタスク知識ではないか、という疑問も生ずる。この議論からもわかるように、問題解決にとって必要な知識の適当な粒度はどうあるべきかを考えないと、実りのない議論に陥ってしまう。このことから、まずタスク

とドメインという二つの概念の定義に関して議論し、その結果次のような意見が出た。

知識処理が対象とするタスクを分解してその粒度を小さくしていくと、どこかでそれ以上分解できない、あるいは分解する必要がないレベルに到達する。例えば、LP(線形計画法)、微分方程式の解法、ソートなどはそれ以上分解したいとは思わない、いわゆる「わかった」解法である。この最小の粒度のタスクに行きついたところで分解を停止することによって、タスクとアルゴリズムを区別することができる。

したがって、解法が定まった問題とそうでない問題との区別は重要である。タスクという言葉は、これまでエキスパートシステムが対象としてきた、解法が明確ではない問題の解決に対して用いるべきであろう。ここで、ドメイン、領域、対象、問題解決、問題、解法、などの概念を以下のように定義してみたい。

- ・**タスク**：ES が対象としてきた問題解決。アルゴリズムではなく、知識に基づく探索が必要となる。例：診断、設計、計画、分析、選択、分類、制御、etc.
- ・**解法**：アルゴリズムが明確な問題解決法。例：ソート、LP、微分方程式を解く、etc.
- ・**領域**：問題解決が行われる場。例：医療、機械、建築、鉄鋼、電力、etc.
- ・**領域知識**：領域で観測される現象を説明する理論、原理知識。例：マクスウェルの方程式、オームの法則、etc.
- ・**対象**：領域を場として作られた(る)もの。例：人体、発電プラント、高炉、車、etc.
- ・**対象知識**：対象に関する知識(領域知識を除く)。対象の構造(トポロジー)、部品の機能。
- ・**ドメイン**：領域と対象の総称。

この分類では、すべての問題はアルゴリズムが存在する問題と知識に基づく探索に基づく問題との2種類の問題に分割されることが仮定されている。これはあくまでも仮説であるが、自然言語理解、同時通訳、アルゴリズムの発見などの問題はタスクの範囲外の高度な問題として除外することにし、それ以外の多様な問題を分析し、アルゴリズムが存在する問題に到達するところで分解を終了するという方法で問題の整理を続けていけば、粒度の決定はできるものと思われる。

〔3〕 数学はドメインか？

〔2〕でも触れたが、数学の知識などはドメイン知識のなかでも共通基盤的な知識であると考えたいという意見と、数学を独立した領域とみなし、数学の対象を表現するのは数学のドメイン知識であり、数学の問題解

決法を表現するのは数学のタスク知識というべきである、との意見がある。これも診断や設計問題の記述や問題解決過程に現れる数学と、数学自身を問題としている場合の相違からくるものであり、粒度の問題ともいえる。これは数学が設計問題から見て小さな問題であるということの意味するものではない。数学はそれ自身さらに細分したほうがよいかもしれない大きな領域であるといえよう。

一般に、数学における問題解決技法、例えば微分方程式では、解法が複数ある場合や明確に解法が定まっていな部分もあるため、実際に適用する際にはさまざまな要因を考慮して適切な解法を選択したり、初期値の設定、現象の近似などを行うことが必要となる。しかし、この作業は上述の「設計タスク」、「選択タスク」などと同様に、アルゴリズムはなく、知識を用いた探索が主となるものであり、「解法」とはいえない性質のものである。この意味で数学における問題解決行為はタスクと解法に分解することができる。

ただし、数学における解法とタスクとの切分けは微妙である。例えば、応用数学者にとっては解法が確立していない問題があり、これはタスクと考えるべきであろう。さらに、数学者という人間の専門家にとっては自明でも、それを実際知識表現言語に表現するのは困難な知識も多いので、数学にもタスクのレベルの問題解決法がかなり含まれている。カオスを例にとれば、その計量法には明確に定式化できない部分が多くあり、どの計量法がどの程度優れているかは議論の対象にもなっている。数学エキスパートシステムを構築するときには、やはり一般のエキスパートシステムと同様の問題が生じるものと思われる。以上の議論の結果、①数学も一つのドメインである、②数学問題解決法もタスクと解法に分かれる、との結論に達した。しかし、数学問題解決固有の我々が名前の付け方を知らないタスクは別として、一般には、知識ベースが対象としているドメインは工学の問題解決の場との暗黙の了解があるのではないかと、との意見もあった。

〔4〕 オントロジーの独立性

異種のオントロジーは直交すべきか、あるいは互いに対応づけられるものであろうか。例えば、安定性解析というタスクを考えると、時間領域で問題を記述するか、ラプラス変換して周波数領域で表現するかは、明らかに等価な二つのオントロジーである。これはともに対象とするプロセスを記述するオントロジーであって、上の分類ではともにドメインオントロジーに属する。よく知られるデバイスオントロジーとプロセスオントロジーも対象を記述する二つの違った視点を表

すもので、これもドメインオントロジーに属する。しかし、この二つは直交しているわけでも等価であるわけでもない。実際、物理システムを記述する場合には、デバイスの中身を記述するのにプロセスオントロジーを使い、デバイスの構造を記述するにはデバイスオントロジーを使っている。つまり、異なる複数の視点からモデル化しないとうまく記述できない対象もあり得る。また、詳細度の違いをもってオントロジーが違っていると主張する意見もある。タスクオントロジーと、ドメインオントロジーは異種のオントロジーであり、表現する対象が異なっている。この意味では、両者は直交すべきものであろう。

異なるドメインオントロジーどうしが互いに変換可能か、対応づけがとれるかどうか(とれなければいけないのか)は現時点ではよくわからない。まず、どのようなオントロジーがあり、それがどのような関係にあるのかを整理することから始めなければならない。また、汎用のオントロジーを考えることは無意味であり、目的を明確に定めたうえで、最も適したオントロジーを選定しなければならない。

〔5〕 デバイスオントロジー vs. プロセスオントロジー

ドメインオントロジーを記述する際の重要な決定事項として、デバイスオントロジーとプロセスオントロジーの選択の問題がある。この選択は対象が集中定数系として記述可能か、あるいは分布定数系で記述しなければならないかに強く依存している。

デバイスオントロジーは集中定数系のモデルの表現に適している。元来、デバイスオントロジーはシステムダイナミクスから生まれてきたものだと考えられる。システムダイナミクスの発想は、現場の技術者が容易に微分方程式を立て、それを解析してシステムのモデル化と解析ができるようにしたものである。その良い例が、集中定数型の電子回路や図式化された機構部品やプラントなどに見ることができる。回路図は回路の部品の同定とその接続関係がわかれば、誰でも作成することができ、回路図から回路方程式も機械的に生成できる。もちろん、各部品をどの程度詳細にモデル化するかなどといった問題が残されているが、それはそのような方式が確立されていない場合に比べるとはるかに少ないことから、デバイスオントロジーの有用性が理解できる。

一方、分布定数系のモデル化には、デバイスの挙動を記述するのに偏微分方程式を用いなければならない。この挙動はデバイスの入出力を直接記述するものではなく、デバイスのなかで起こっている空間的、時

間的な挙動を記述するものである。このような場合には、生起している物理プロセスを中心とした視点を採用していることになり、プロセスオントロロジー主体の記述を採用せざるを得ない。しかし、プロセスオントロロジーではすべての現象がもれなく重複なく記述できることを保証するのは難しい。専門家がその領域において生じ得る現象をいちいち記述しなければならないからである。また、部品といった局所概念もなく、外観(つまり構造)から機械的に合成できないという問題がある。

さらに、次のような問題点もある。

1. すべての領域の現象がデバイスオントロロジーを用いることができるように整理が終わっているわけではなく、デバイスオントロロジーが利用できるのはかなり枯れた領域だけである。
2. どちらの考えを採用したらよいかを、簡単に判断できない場合もある。
3. デバイスオントロロジーという立場をとった場合、すべての変数の値は直接接続されたデバイスを通してのみ伝搬する。したがって、導管のように変換しないでただ素通りさせるデバイスを定義する必要がある。また、放射線や電磁波などの影響がデバイスを介さないで伝わる現象は扱えない(扱いにくい?)、などの問題がある。

以上のように、デバイスオントロロジーとプロセスオントロロジーに関しては一応の理解が得られたものの、不明確な部分も多く、今後継続して議論する必要がある。

[6] オントロロジーとプリミティブ(単語)集合

オントロロジーの決定にはものの見方が大きく関与しており、見方を決めた後、その見方に沿って領域に関する知識を記述するためのプリミティブ(単語)集合を定義する必要がある。ここで大切なことは、適当な粒度(抽象度)を選定し、汎用的なプリミティブを数え上げることである。

粒度の議論はタスクオントロロジーにもドメインオントロロジーにも当てはまる。足し算、引き算をタスクオントロロジーのプリミティブとして捉えるにはあまりにも粒度が細かすぎる。しかし、生成・検査だけでは粗すぎる。タスクオントロロジーのプリミティブを決めるには、ドメインの専門家が理解でき、かつドメインに依存せずタスクに固有のプリミティブを選定する必要がある。タスクごとに選定されたプリミティブのなかには複数のタスクに共通に使用されるものもあろう。ドメインオントロロジーのプリミティブを決めるには、多くの対象を記述するのに必要なドメインを整理して、ドメインごとにプリミティブを選定する必要がある。

る。

大切なことは、入れ物ではなくて、その中身である。オントロロジーの研究グループ(特に、アメリカ)には、知識表現言語や異種表現言語間の変換の自動化の研究に力を入れているものがある。言語や文化を超えた万人に共通のプリミティブの存在を疑問視する気持ちがわからなくはないが、使用目的を限定したうえで中身のプリミティブの選定のほうに注力すべきである。

このように、万人に共通なオントロロジーの存在に関しては一般に疑問視する考えがあるが、知識の共有には共通のオントロロジーの問題を避けては通れない。

Schank の CD 理論が自然言語理解の標準オントロロジーにならなかった一つの理由には、自然言語理解が常識に関わる困難な問題であったことが関係している。それを事例ベース推論で回避できるとする考えがあるが、それは誤りである。事例が万人に共通な知識であるためには、それを検索するインデックスが必要となるが、それは万人に共通でなければならず、詳細なインデックスは事実上、事例を記述するオントロロジーと同質のものとなるからである。結論としては、知識の共有と再利用を目指す限りは、ある程度の範囲の人々が承認する共通のオントロロジーを確立する必要がある。

タスクオントロロジーに関しては、大阪大学の溝口が研究成果を報告した。タスクオントロロジーの具体例とその同定手順の概要を、スケジューリング、および設計タスクを例にして示した。タスクオントロロジーは、処理のプリミティブを表す動詞、処理の対象を表す名詞を中心として構成されており、対象とするタスクの制御構造をドメイン独立な形で表現することができる。すべての動詞はコンピュータコードと対応がつけられているため、タスクの制御構造をコンピュータのコードに変換することができる。したがって、タスクオントロロジーを用いることによって、エキスパートシステムの推論構造を記述して事例ベースに蓄積して再利用することを可能にしている。また、すべてのタスクオントロロジーは対応するドメインの用語に翻訳することができるため、専門家とのインタフェースの改善にも役立つ。スケジューリングタスクに関して得られたオントロロジーは、動詞が 26 個、名詞が 40 個、形容詞が 11 個、制約関連語彙が 67 個、ゴール関連語彙が 21 個である。タスク解析インタビューシステムのプロトタイプはすでにマッキントッシュ上で実現している。有効性の評価はまだ十分ではなく未完成ではあるが、初めての提案として注目に値する。日本 IBM の堀らからも同様の発表もあったが、相互の比較検討は不

十分であり、今後いくつかの提案を待って評価をするべきである。

オントロジーの同定に関して困難な点は、ドメイン依存の概念と、タスク依存の概念の切分けと、粒度、そして抽象度である。実はこの三つは互いに深く関連しており、独立な概念ではない。先に指摘したように、どのような問題も生成・検査法として定式化できるが、そのレベルでオントロジーを探しても抽象度が高すぎて意味がない。逆に、計算機言語の命令のような細かいレベルでの記述もまた無意味であることは自明である。それでは適切なレベルはどこにあるのであろうか？ 非常に曖昧ではあるが、対象とするタスク、例えばスケジューリングタスクに固有の構造を反映する程度に詳細であり、かつそれで記述した事例が再利用できる程度に一般的でなくてはならない。この詳細さと再利用性のトレードオフの問題を明確に解消する指導原理を見つけるのは困難であるが、経験を通してある程度の基準を設定することは必要である。現在の状況ではオントロジーは各研究者の主観に大きく依存しているが、これを解消することが重要な課題であることは参加者全員の一致するところであった。

2. 共有と標準化

知識を共有するためには何らかの標準化が必要であることは明らかである。ここで問題となるのが“標準化”という“問題児”である。一般に“共有”は喜ばれるが、“標準化”は嫌われる傾向があるのは、洋の東西を問わず無視することのできない現実である。しかし、ある程度の標準化をしなければ共有は不可能であることも事実なのである。この矛盾をどのようにすれば超えることができるかも議論の対象となった。

共有の標準化はともに生産性を向上するためのアプローチであることを認識して、現実的な態度で対処すべきであろう。具体的には、固定的なオントロジーを受け入れることには抵抗があることを考慮して、ひな型としてのプリミティブを提供し、使う側が適当に修正することを許せば、多少は標準化が受け入れやすくなるであろう。現段階では具体的にいくつかの標準化案を提案し、比較評価することが重要である。その結果、悪いものは淘汰され、良いものが残るだろう。その意味で、阪大の汎化語彙は標準語彙の候補となると考えてよい。

3. マルチエージェントとエージェント指向

知識の共有と再利用をどのようなソフトウェア機構によって実現するかも重要な話題である。ソフトウェ

アの部品化、部品の知能化という線で可能性を探ると、オブジェクト指向、およびそれを高度化した「マルチエージェントシステム」という考え方が発生するのは自然ななりゆきと考えられる。このような期待感に対して、これがマルチエージェントシステムだという決定版が存在しないので、マルチエージェントシステムとはそもそも何であるのか、既存のパラダイムと実際どこが異なるか、実際のところ知識の共有と再利用にどれくらい有効なのかなど議論が続出した。

主な話題を拾ってみよう。第1に、「オブジェクト指向とどこが違うか？」という疑問が提出された。オブジェクト指向とエージェント指向を比較するとだいたい次のように区別できる。

キーワード	特性
オブジェクト指向	カプセル化, 抽象化, メッセージ伝達, 継承
エージェント指向	協調問題解決, エージェントの自律性(リフレクション, 自己組織化), 高レベルの相互作用

このような違いが果たしてパラダイムの違いと呼べるのであろうか。パラダイムの違いはないと主張する側は、エージェント指向の概念はすべてオブジェクト指向の概念によって記述できるのではないかという考えに基づいている。パラダイムの違いはやはりあると主張する側では、エージェント指向プログラミング言語の持つ、受託(commit)や説得(persuade)という高級概念に基づいて手続きを記述できることがすなわちパラダイムの違いであることを強調した。

第2の話題は、「エージェント指向において、個々のエージェントは自律的な動作をするのに、果たしてエージェント社会全体として所定の目標が達成させることが可能か？ それはなぜか？」という疑問であった。このあたりは、まだこれから研究が進められるところであるが、エージェント間のプロトコル(通信規約)やエージェントの存在する場に工夫をしておくことによって、個々のエージェントがそのプロトコルに従うだけで、エージェント社会全体の一様性が保たれるようにしておくことが考えられる。

この点に関しては後でも触れるが、ここではプロトコルを決めることの、問題の困難さに対する貢献度の大きさも論じられた。すなわち、プロトコルは計算機言語の構文を決めることに相当するが、プログラミングという概念(この場合は通信という概念)がすでにある以上、問題を解く場合に議論すべきことは、どのようなアルゴリズムを考えるかであり、このようなとき

に、そのアルゴリズム実現の言語の構文を議論しているようなものである。エージェントの問題では、どのような内容の通信を行えば自律したエージェントが協調して全体の目的を達成することができるのか最も本質的なことであるにもかかわらず、その通信プロトコルを議論することの有用性に疑問が投げかけられた。この意味で、京大の西田らのようなドメイン概念を含めたプロトコル研究が必要であろう。

第3に、「知識をひとまとめにしておくのではなく、エージェントに分けておくことによるメリットは何か？」という疑問が提出された。これに対する回答はエージェントプログラムにおけるエージェント開発・保守の手間を考えると、自明であろう。さらに、多様な問題解決法や知識表現を容認することが可能になり、またこうすることによってエージェント内部を記述するための内部表現言語の標準化ではなく、エージェント間の外部言語のレベルで、ある程度の標準化を行うことが可能になり、知識の共有と再利用が一步前進するであろう。

また、これに関連して「エージェントに持たせる知識の粒度はどれくらいが適当か」という疑問もある。この疑問はタスクオンロジーの話と直接関連する。

第4の話題として、マルチエージェントシステムと知識の共有と再利用の視点の違いが議論された。それによると、マルチエージェントの研究は計算指向である。つまり、システムの良さは、それによって計算資源消費がどれだけ削減されるかによって評価される。これに対して、知識の共有・再利用は情報指向である。つまり、進歩はエージェントに分散された情報資源がどれだけ有効に利用されるかで評価される。

第5の話題として、「枠組みより内容が重要である」という意見が出た。これは、知識の共有と再利用においてソフトウェア機構の話題になると、共有・再利用すべき知識の中身ではなく、知識を収納する「入れ物」の話ばかりになりがちなことへの批判である。

第6の話題として、CSCW (computer supported cooperative work) と何が共通で何が違うのかについて議論した。CSCW は人間の創造的活動の支援とグループの意思決定を第1に考えたものであり、必ずしも高度な処理を全自動で行う必要はなくなる。これに対して、マルチエージェントによる知識の共有・再利用では、創造的活動はまだスコープに入っていない。もっと月並みなことをきちんとできるようにするところに興味があるといっているだろう。グループの意思決

定は両者において重要である。

4. システム開発者の視点とユーザの視点

システム開発者からは「間違いなくシステムを作る (building the system right) こと」が重要である。それに対してユーザからは「所望のシステムを作る (building the right system) こと」が重要である。前者は verification の問題で、後者は validation の問題である。知識の共有と再利用は、間違いなく動くシステムを作るのには適しているが、果たして出来上がったものが、本当にユーザが望んでいるものであるかの保証はあるのか、このような観点から見た場合に知識の共有と再利用は何かネガティブな要因を持っていないかどうか、という疑問が提示された。

その背景には、最初から仕様が明確に記述できるような問題に対しては知識ベースシステムのアプローチを採用する必要はなく、知識ベースシステムを開発するからには、もっともやもやとした問題を対象とすべきである、との立場がある。このような場合でも、validation の段階で不具合を検出でき、その結果がフィードバックできる仕組みがあることが大事なことであって、これはあらゆる問題解決の手法が直面する問題である。オンロジーを明確にして、共通のベースとなるプリミティブを準備しておくことは、ポジティブな要因となってもネガティブな要因とはならないのではないか、ということで合意が得られた。

5. ま と め

初めにお断りしたように、ワークショップでの議論のすべてをカバーすることはできなかったばかりか、オンロジーにずいぶん重点を置いた報告になってしまった。CSCW に関する議論も活発にあったのであるが、まとめ役の筆者らの専門から遠いこともあって、うまく吸収できなかったことがその大きな原因である。筆者らの非力をお詫びしたい。しかし、2日間の議論は極めて活発で、参加者全員がワークショップの意義を体感したものと思われる。なお、このワークショップは毎年定期的開催される予定であるので、興味をお持ちの方は、ぜひ参加下さるようお願いする次第である。

(元田 浩 ((株)日立製作所基礎研究所)
溝口理一郎 (大阪大学産業科学研究所)
西田 豊明 (奈良先端科学技術大学院大学))