

センサデータの周期的更新を反映した推論機構 JUSTO の提案

神田 武*

佐竹 聡†

川島 英之‡

中村 学§

今井倫太¶

慶應義塾大学 理工学部^{||} * ‡ ¶慶應義塾大学大学院 理工学研究科^{**} † §

{takeshi,satake,kawasima,manabu,michita}@ayu.ics.keio.ac.jp

1 序論

我々の研究室では現在、セマンティック・センサネットワーク [1] という新たなシステムを提案している。本研究はセマンティック・センサネットワーク環境における推論機構を提案することを目的とする。

セマンティック・センサネットワーク (以下, SS) とは, センサネットワーク内の各センサデータに物体情報やデータの意味を表すメタデータを付与してデータを論理表現に変換するミドルウェアである。これによりアプリケーションはセンサネットワーク内の環境情報を単なるセンサデータの数値列ではなくメタデータとして得られ, 環境の論理的解釈が可能になる。複数のアプリケーションがメタデータを共有することでアプリケーション間の統一的な環境認識と重複する数値的処理の低減が実現される。例えばユビキタスコンピューティング環境のスマートルームの環境認識やコンテキスト・ウェアなアプリケーションへの利用に有益と考えられる。先行研究として広田らのメタデータ管理機構 MeT[1] がある。

ここで, SS においてメタデータから新たなメタデータを推論によって生成することが有用である。複数のメタデータが成立したとき, その状態を新たなメタデータで表現することによってより高いレベルでの環境の解釈が可能となる。さらに, ユーザが推論を行う際のルールを自由に記述できれば, センサデータや物体情報から得られる基本的なメタデータをもとにユーザが各自好みのメタデータを作成することができる。これによりセンサデータから得られる単純なメタデータだけを用いるよりも, よりユーザの要求に合った環境表現が可能になる。したがって SS 上で推論ルールを読み込み, 汎用的な推論を行う推論機構が必要となる。本研究では推論の方法としてメタデータとセンサ ID の組を述語論理のシンボルとして入力し, 条件のシンボルがすべて成立すれば結果のシンボルを生成することを行う。

一般の推論機構として prolog のインタプリタがある。しかし SS 環境ではセンサデータの変化に伴ってメタデータが動的に変化するため, prolog など従来の推論機構で推論を行うと次の2つの問題が生じる。

問題 1 シンボルの消滅を扱えない。センサデータの変化によりセンサとメタデータの対応関係が変わり, メタデータが消滅することがある。一方, prolog は登録したシンボルを不変として扱う。そのためメタデー

タの消滅を反映するためには全シンボルを最初から再登録しなければならない。

問題 2 推論実行中にシンボルの更新を行えない。メタデータの変化はシンボルの変化を引き起こす。そのため, 推論機構は常に現在のシンボル状態を反映しなければならない。一方, prolog は推論実行中のシンボルの変化を想定していない。そのため prolog は古いシンボルを使った推論を行い, 誤った結果を返す。そこで本研究では推論機構 JUSTO を提案する。JUSTO はシンボルの取り消し機能によりシンボルの消滅に対処する。JUSTO は周期的なシンボルの読み込み機能を持たせて, シンボルの更新に対処する。

2 背景

2.1 メタデータ

SS のメタデータは基本メタデータと推論メタデータがある。基本メタデータはセンサが付加されているオブジェクトの情報やセンサデータの解釈を表す。センサデータの解釈はセンサの種類とデータ値に依存する。例えば1つの温度センサのデータ値から *hot* や *cold*, 2つの位置センサから *near* や *on* というメタデータが付与できる。

推論メタデータは複数のメタデータの成立条件から推論を行うことで得られるメタデータである。例えば, “水の入ったコップが動いている物の上にあるならば, 濡れる危険がある” というルールがあると, “*on(A, B)* かつ *cup(A)* かつ *full(A)* かつ *move(B)* ならば *wetDanger(B)*” という推論ルールより推論メタデータ *wetDanger* が生成される。

2.2 MeT

SS のメタデータ管理機構 MeT ではセンサデータから基本メタデータへの変換を行い, それをデータベースへ登録する。さらに, 各センサのメタデータの問い合わせ及び監視機能を持つ。

2.3 推論機構 JUSTO のシステム構成

本稿では MeT から基本メタデータとセンサ ID の組を読み込み, 生成した推論メタデータとセンサ ID の組を MeT に出力することを想定する。JUSTO はルールベース, 推論制御部, 作業領域, シンボル読み込み待ちバッファから成る。JUSTO のシステム構成を図 1 に示す。

2.4 推論ルール

JUSTO での推論ルールの記述の仕方を述べる。JUSTO は prolog に似た推論ルール構文を持つ。まず推論の結果生成されるシンボルを書き, 次に “:-” を書き, 次に条件となる複数のシンボルを書く。条件のシンボル同士は “&” で接続される。例えば 2.1 節で述べた *wetDanger* の例であれば推論ルールは以下のように記

*Takeshi Kanda

†Satoru Satake

‡Hideyuki Kawasima

§Manabu Nakamura

¶Michita Imai

||Faculty of Science and Technology, Keio University

**Graduate School of Science and Technology, Keio University

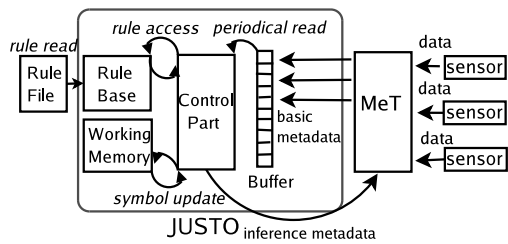


図 1 システム構成

述される。

$wetDanger(B):-on(A,B)\&cup(A)\&full(A)\&move(B)$.
ここで、変数 A,B は任意のセンサ ID を表し、推論時には変数に実際のセンサ ID が置換されて条件の充足判定が行われる。

3 推論機構 JUSTO

3.1 JUSTO の動作

JUSTO の動作は以下ようになる。JUSTO はまずルール記述ファイルからルールを読み込んでルールベースを作成する (図 1 の *rule read*)。次に基本メタデータとセンサ ID の組をバッファから読み込む (図 1 の *periodical read*)。読み込んだ結果は作業領域に入力される (図 1 の *symbol update*)。さらに、作業領域内のシンボルとルールベースの各ルールの条件部のマッチングをとる (図 1 の *rule access*)。条件を満たすシンボルがあれば新たなシンボルを生成し、作業領域を更新する (図 1 の *symbol update*)。また、推論実行中にもシンボル読み込みを周期的に行う (図 1 の *periodical read*)。以上で述べた全体の流れを図 2 に示す。

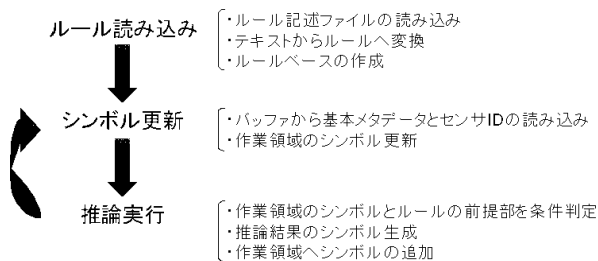


図 2 動作フロー

JUSTO は推論機構として次の特徴的な機能を持つ。ネットワーク構造を用いたシンボル取り消し機能, 周期的なシンボルの読み込み機能である。

3.2 ネットワーク構造によるシンボルの依存関係の表現

推論実行結果として新たにシンボルが生成されると、作業領域内では推論ルールの条件部と結論部のシンボルはネットワーク構造で連結される。例えば、実際に $on(1,2), cup(1), move(2), full(1)$ のシンボルが成立して上記の $wetDanger$ の推論ルールの条件を満たし、結論のシンボル $wetDanger(1)$ が生成されるとする。この場合シンボル間の連結は図 3 のようになる。

3.3 シンボル取り消し機能

JUSTO はシンボルの取り消しができる。あるシンボルが消滅したときは、作業領域内のネットワークを再帰的に辿って関連するシンボルを全て消去することにより、誤りのないシンボルの取り消しを実現する。例えば

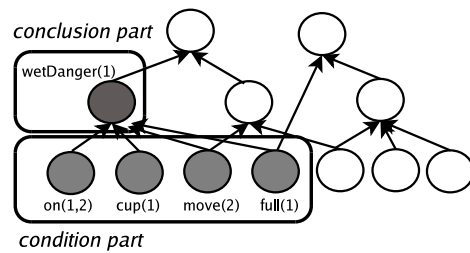


図 3 シンボル間の依存関係のネットワーク

図 3 で $on(1,2)$ のシンボルが消滅すれば、シンボル取り消し機能により $on(1,2)$ を条件として成立している $wetDanger(1)$ などのシンボル群も消去される。

3.4 周期的なシンボル読み込み機能

JUSTO は推論実行中に、バッファにあるシンボルを周期的に読み込み、シンボルの更新を推論結果に随時反映させる。シンボル読み込みの結果、新たに生成されたシンボルの推論を開始し、消滅したシンボルに関しては上記のシンボル取り消し機能によって関連するシンボルを全て消去する。こうして動的に基本メタデータの更新を反映する推論機構を実現する。

3.5 問題の解決

JUSTO はネットワーク構造を用いたシンボル取り消し機能を用いて、シンボルが消滅したときに、取り消したいシンボルに関連するシンボル群だけを消去することが可能である。こうして、シンボル消滅の度に推論機構側でシンボルを全て登録し直さなくてもよくなる。したがって JUSTO は問題 1 を解決する。

JUSTO は実行中のシンボルの周期的読み込み機能を用いて、消滅したシンボルに関する推論を途中で打ち切り、新たに生成されたシンボルに関する推論を動的に組み込む。こうして、時間のかかる推論でも実行中のシンボル更新を反映することができる。したがって JUSTO は問題 2 を解決する。

4 今後の課題

JUSTO の実行コストはシンボルの数、ルールの数に伴って爆発的に大きくなる可能性がある。それゆえシンボル数やルール数、実行時のシンボル比較回数等がどのように実行コストに影響を及ぼすか評価する必要がある。また、推論プログラムの最適化も求められる。具体的には条件の充足判定部分のアルゴリズムの改良などが有効と考えられる。さらに、実際のアプリケーションへの利用も実現したい。

5 結論

JUSTO はシンボル取り消し機能でシンボルの消滅を扱った。JUSTO は周期的なシンボルの読み込み機能でシンボルの更新を扱った。JUSTO はこれら 2 つの機能により、SS 環境で推論を行う際の問題点を解決できたと結論する。

参考文献

[1] 広田 裕, 川島 英之, 佐竹 聡, 梅澤 猛, 今井 倫太: セマンティックセンサ・ネットワークの実現に向けた実世界指向メタデータ管理システム MeT の設計, 情報処理学会研究報告. 2005-ICS-141, pp.29-36, 2005