

# オブジェクト関連を用いたセマンティクスの定義とその利用法

Definition of Semantics with Associated Objects and Its Usage

楓 仁志† 中村 絵美† 島川 博光†  
Satoshi Kaede Emi Nakamura Hiromitsu Shimakawa

## 1. はじめに

現在、我々の生活をサポートするツールとしてさまざまなスケジュール管理ツールが開発され、利用されている。ユーザはそれらを利用して用件の時刻や実行場所、件名、メモなどを入力し、スケジュールを管理する。しかし、Microsoft 社製 Outlook のような既存スケジュール管理ツールでは、入力された項目を利用してサービスを動的に提供する点においては、ユーザが本来求める機能は実現できていない。たとえば、顧客先での仕事が予定より早く終わり「空いた時間」ができた営業マンに対して、顧客先の近くにある市役所で「ついで」に私用を済ますことをツールからユーザへ提案することは不可能である。

上述のような機能を実現するためには「空いた時間」というユーザの状態を考慮する必要がある。そのほかに、顧客先と市役所が地理的に「近い」という事実間に関連が存在することを認識し、利用することが必要となる。

ユーザ状態を考慮したツールとして CybreMinder[1]が開発されているが、そのリマインド機能が利用するのはユーザの存在場所だけである。CybreMinder や Outlook のリマインド設定時には通知時刻を設定する必要があるため、ユーザの状態に柔軟に対応できない。また、ユーザのスケジュールを利用したツールとしては、iCAMS[2]が開発されているが、その機能は登録した時刻と実行場所に応じて通信手段を切りかえることにとどまっている。

これらの研究では対象用件に直接係わる事実だけが利用されている。本稿ではこれら事実だけでなくスケジュールに登録されるタスクのセマンティクスとして、事実に関連するさまざまな情報を用いてタスク間の関連を表現する。

タスク間関連を利用したスケジューリングとして生産工程管理[3]や CPU のタスクスケジューリング[4]が挙げられる。しかし、[3]の場合スケジュール確定後にタスクを随時追加することができず、個人スケジュール管理には適応できない。また、[4]の場合はタスクを実行するうえで必要な事実が同一でなければならないため、きわめて不便である。

セマンティクスの定義としては SCROL[5]をはじめ、多くの研究がされている。我々はタスクとして登録されたオブジェクトの持つ関連情報の定義をおこない、その関連度についても定義する。そして、タスク実行に係る関連情報をセマンティクスとし、セマンティクスの積集合を関連セマンティクスと捉えることでタスク間関連を表現する。このタスク間関連とユーザの状態、スケジュールを利用することで「ついで」を実現するスケジューラを提案する。

以下本稿では、2章でセマンティクスの定義、3章でイベント間関連の利用法、4章で関連を利用したスケジュール管理、5章で既存ツールとの比較をおこない、6章でまとめる。

## 2. セマンティクスの定義

### 2.1 セマンティクスと関連セマンティクス

本稿におけるオブジェクト  $o$  とは、場所、物品、人物とする。タスクを実行するうえで必要な複数のオブジェクトが存在し、またそのタスクの実行により状態が変化するオブジェクトもある。タスクに参照されたオブジェクトには強く関連する複数のオブジェクトが存在し、これらの集合を関連情報  $R(o)$  と呼ぶことにする。オブジェクトの関連情報  $R(o)$  の和集合をとったものをそのタスクのセマンティクスと定義する。また、複数のタスクが存在するとき、それぞれのセマンティクスの積集合をこれらタスクの関連セマンティクスと定義する。

登録済みのタスクに対して、この関連セマンティクスについて特定の条件が成立するタスクを「ついで」に実行可能であると判断し、スケジュールに挿入する候補とする。

### 2.2 関連情報の定義

あるタスクから参照されたオブジェクト  $o$  を考える。 $o$  の関連情報  $R(o)$  を次式に示す。

$$R(o) : \{o_i \mid \alpha_i(o, o_i) \geq c\} \quad (c \text{ は定数})$$

ここで  $\alpha_i$  とはオブジェクト  $o$  に関連するオブジェクト  $o_i$  間の関連度と定義する。しかし、オブジェクト  $o$  に関連するオブジェクト  $o_i$  は多数存在する。そのため、スケジュールに登録されているタスクの「ついで」となりうるタスクを同定するさいに、 $o$  を介して登録されているタスクに関連するオブジェクトすべてを利用することは非現実的である。そこで、関連が強いオブジェクトのみを考えるようにする。図1に示すように  $o$  と関連するオブジェクト  $o_i$  の関連度を  $\alpha_i$  とし、 $\alpha_i$  が一定値  $c$  以上の関連情報  $R(o)$  のみをタスクのセマンティクスとして利用する。  $c$  を  $R(o)$  を決定する閾値と呼ぶことにする。

「ついで」を実現するためには、 $c$  の値をユーザの状態によって適切に変化させる必要がある。そこで、オブジェクトを表現する値として時間的要素、地理的要素、役割的要素に基づいて  $c$  を決定する。時間的要素とは、各オブジェクト間で意味を成す時刻と期間であり、地理的要素

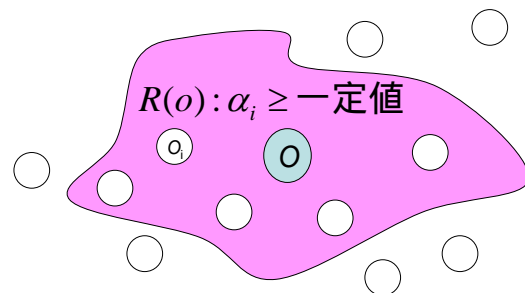


図1 関連情報と 値

† 立命館大学大学院 理工学研究科

素とはオブジェクト間の距離である。役割的要素とはオブジェクト  $o_i$  がなう複数の役割である。たとえば、「A 郵便局」というオブジェクトは郵便物を送る役割、預金の役割、指定金額を引き落とす役割を持つ。これらオブジェクトと役割の種類は既に定義されており、取得可能であるものと仮定する。

### 3. イベント間関連の利用

#### 3.1 イベントを用いたアクションの定義

本稿では、基盤データベースとしてコンテキスト・データベース[6]を利用する。これはユーザのコンテキストをイベントのシーケンスと捉え、イベントを時刻印、関連する情報、関係する外部コンテキストの組と定義するものである。これらを踏まえたうえで、図2に示すモデルを提案する。

- ・ アクト(Act)：開始イベントと終了イベントを統合したもの。特に、実行時刻は未定であるが必ず行う未来のアクトを浮動アクト(To-do Act)と呼ぶ。
- ・ アクション(Action)：一連のアクトの集合によって完結する作業である
- ・ 定型アクション(Rule Action)：頻出するアクション  
筆者が提案する浮動アクトとは、登録時に実行時刻を設定する必要がない。よって、ユーザにとって都合がよいとスケジューラが判断するタイミングで登録される浮動アクトをスケジュールに織り込むことが可能である。スケジューラはスケジュールに登録されているアクトのオブジェクトや順序について、関連する浮動アクトを浮動アクトリストから検索し、「ついで」に実行できると判断した場合にスケジュールに挿入し、ユーザにアドバイスとして提示する。

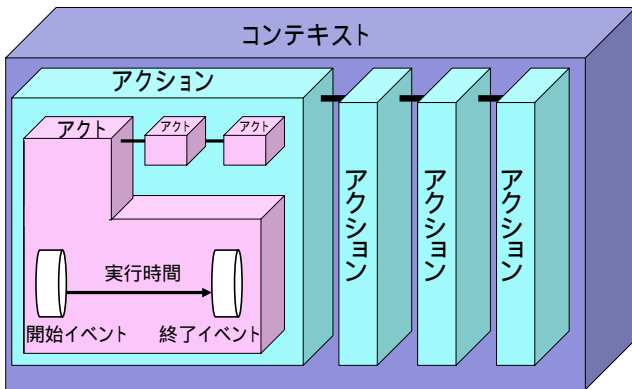


図2 提案モデル

#### 3.2 アクト間関連の種類

「ついで」を実現するためにはアクト間の関連を明確に分類する必要がある。なぜなら、「ついで」を表現するアクト間の関連とは多種多様であり、すべての場合に適応する条件を同定することはきわめて困難だからである。そこで、既存スケジュールに対してコンテキスト・データベース[6]に登録されている項目に注目したアクト間関連を補充関連、実行順序関連、オブジェクト関連に分類する。与えられた関連がどの分類に属するかを判定するための条件を次にしめす。

- ・ 実行順序関連：スケジュールに登録されているアクト  $a$  がアクション  $A$  の一部であり、かつ  $A$  の一部に浮動アクト  $t$  が含まれる。
- ・ オブジェクト関連：アクト  $a$  から参照されるオブジェクト  $o_a$  の関連情報  $R(o_a)$  と浮動アクト  $t$  から参照されるオブジェクト  $o_t$  の関連情報  $R(o_t)$  の積集合が空集合でなく、 $o_a$  と  $o_t$  の双方がこの積集合に含まれる。
- ・ 補充関連：オブジェクト関連のうち、とくに  $o_a$  がユーザであり、彼が実行する時点で浮動アクト  $t$  から参照されるオブジェクト  $o_t$  とユーザ  $o_a$  の存在場所が同一でない。

### 4. 関連を利用したスケジュール管理

#### 4.1 挿入判定に用いる条件

上述の各定義を利用するスケジュール管理ツールについて述べる。図3に示すように、各浮動アクトには一般条件と固有条件が設定されている。

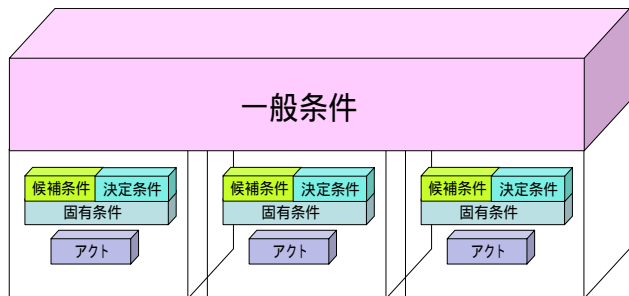


図3 一般条件と固有条件

各条件を判定するさい、上述のように関連するオブジェクトのすべてを利用せずに閾値  $c$  を用いて絞りこんだ関連情報を使用する。一般条件とは、すべての浮動アクトに共通する条件である。本稿では一般条件を時間的制約だけで表現されるものとする。固有条件とは各浮動アクトに設定される条件である。固有条件は上述のアクト間関連を分類する条件のいずれかであり、候補条件と決定条件に区別される。候補条件は、判定対象となる浮動アクトを浮動アクトリストから絞りこむ、処理時間の短い単純な条件であり、この条件が成立したとき浮動アクトはスケジュールに織り込まれる候補となる。最終的に決定条件が成立したとき、その浮動アクトはスケジュールに織り込まれる。

#### 4.2 関連セマンティクスを利用したスケジューラ

「ついで」となりうる浮動アクトを検索するさい、スケジューラは、候補条件判定で対象浮動アクトを絞りこみ、候補リストを作成する。次に、一般条件判定として時間的制約を満たすものを検索し、最後に決定条件判定を行う。ここで、関連セマンティクスを利用したスケジューラはオブジェクトの状態を管理する内部記憶領域を持っているものとする。さらに任意のアクトがスケジュールに織り込まれたとき、そのアクトを実行したユーザとオブジェクトの状態をこの記憶領域に書きこめるものとする。

例を用いてスケジュールへの各条件判定を図4に示す順に述べる。ユーザAは浮動アクト*t*:「指定金額を下ろす」を登録しているとする。以下に設定される固有条件を示す。

- ・候補条件 ユーザの存在場所と金融機関が近い

- ・決定条件 挿入された時刻が金融機関の業務時間内

スケジュールはアクトから参照されるオブジェクトの状態について内部記憶領域を調べ、ユーザ  $o_a$  の状態を取得する。 $o_a$  の関連情報として金融機関( $o_1, o_2$ )が検出されたとする。このとき候補条件の「近い」という地理的要素に注目する。ユーザ  $o_a$  の近くにあるオブジェクトの集合が  $R(o_a)$  であり、 $o_1, o_2$  の近くにあるオブジェクトの集合が  $R(o_1), R(o_2)$  である。地理的要素として「近さ」に着目し、関連度の閾値を適当にとれば  $R(o_a), R(o_1), R(o_2)$  が絞られる。いま、 $R(o_a)$  と  $R(o_2)$  の積集合が空集合でなく、かつ、 $o_a$  と  $o_2$  がこの積集合の要素なら、オブジェクト関連により候補条件が成立する。よって、この浮動アクトを候補リストに追加する。

次に、一般条件判定を行う。アクト  $a_j, a_{j+1}$  間に浮動アクトを挿入するとき  $a_j, a_{j+1}$  間の移動時間を  $m_1$ 、その前後の余裕時間を  $f_1, f_2$  とする。また、対象浮動アクトを挿入したときに  $a_j$  から  $t$ 、 $t$  から  $a_{j+1}$  への移動時間を  $m_2, m_3$  とし、浮動アクト  $t$  の実行時間を  $(t)$  で表すとき

$$f_1 + m_1 + f_2 \geq m_2 + \delta(t) + m_3$$

が成立すると一般条件が成り立つ。金融機関  $o_2$  への移動時間と金融機関での作業時間を上記の式に代入した結果、一般条件が成立するとする。なお、定型アクションに登録されているアクトが浮動アクトに登録されている場合は、それを優先してスケジュールに織り込むようにする。なぜなら、一般に定型アクション内のアクトを連続して実行することがよく要求されるからである。

次に決定条件判定の結果、スケジュールにこの浮動アクトが挿入される時刻が金融機関の業務時間内であると、同じくオブジェクト関連により決定条件も成立したため浮動アクトを挿入したスケジュールをユーザにアドバイスとして提示する。

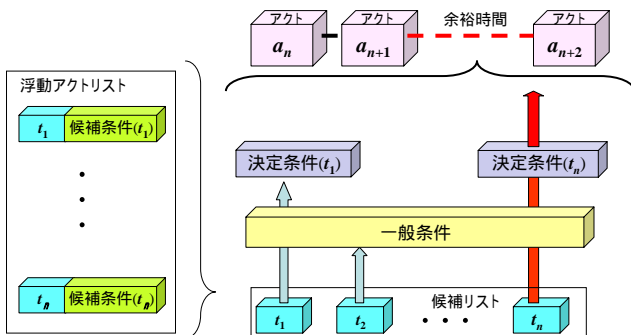


図4 浮動アクト織り込み条件判定

## 5. 評価

提案手法と CybreMinder[1]をスケジュールの主要機能と考えられる8項目を用いて評価する。評価を表1に示す。

提案手法の特徴としては CybreMinder[1]に比べてリマインド表示タイミング、能動性、スケジュール選択機能、スケジュール調整機能が向上していると筆者らは考える。

上述の向上理由としては、アクトのセマンティクスを SCROL[5]のように言葉の意味的階層として捉えるのではなく、アクトから参照されるオブジェクトの関連情報として捉えたことが考えられる。しかし、各条件を設定する必要があるため、入力複雑性が増加した。また、判定すべき条件が増加しているため処理速度の低下が否めない。

表1 評価

	提案手法	CybreMinder
処理速度	×	
入力複雑性	×	
リマインド表示タイミング		
能動性		×
スケジュール項目選択機能		×
スケジュール調整機能		×

## 6. おわりに

我々はオブジェクトの関連情報を定義し、タスク実行に関係する関連情報をタスクのセマンティクスと定義した。さらに、タスク間関連を分類し、タスクの関連情報とユーザの状態、スケジュールを利用することにより、あるタスクを実行する「ついで」に実行できるタスクを動的にスケジュールする手法を提案した。

今後、ユーザの状態と関連情報を自動取得することで入力複雑性の問題を解決する予定である。また、関連度の閾値決定手法についても今後の検討課題とする。

## 参考文献

- [1] A.K.Dey, G.D.Abowd, "CybreMinder: A Context-Aware System for Supporting Reminders", Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing (HUC2K), Bristol, UK, (2000)
- [2] 高橋 一成, 辻 貴孝, 中西 泰人, 大山 実, 箱崎 勝也, "iCAMS: 位置情報とスケジュール情報を用いたモバイルコミュニケーションツールの構築", マルチメディア, 分散, 協調とモバイル(DICOMO 2001)シンポジウム, pp.513-518, (2001)
- [3] 荒川 雅裕, 冬木 正彦, 井上 一郎, "納期遅れジョブの削減・解消を目的とする能力追加を考慮したシミュレーション生産スケジューリング法", システム制御情報学会論文誌, vol.16, No.9, pp.451-460, (2003)
- [4] Jane W.S.Liu, Kwei-Jay Lin, Wei-kuan Shih, Albert Chuang-shi Yu, Jen-Yao Chung, Wei Zhao, "Algorithms for Scheduling Imprecise Computations", Journal of IEEE Computer, vol.24, No.5, pp58-68, (1992)
- [5] Sudha Ram, Jinsoo Park, "Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data and Schema-Level Semantic Conflicts", IEEE Transactions on Knowledge and Data Engineering, vol.16, No.2, pp189-202, (2004)
- [6] 中村 絵美, 楓 仁志, 島川 博光, "関連を用いたコンテキストのデータ表現モデル", 第2回情報科学技術フォーラム(Fit2003), pp49-50, (2003)