

オブジェクト間の制約伝播による問題解決方式

1C-9

佐塚 秀人

NTT データ通信(株)

横山 孝典

(財) 新世代コンピュータ技術開発機構

1 はじめに

多くの知識システムの対象モデルの記述方式としてオブジェクト指向による知識表現が採用されている。しかし、オブジェクト指向プログラミングでは、対象の属性やオブジェクト間の関係をうまく表現することができないという問題がある。そこで最近、オブジェクト指向を基本とし、その属性値や構成要素などに関する制約を宣言的に記述できる知識表現が提案されている [1]。

しかしそれらの制約を効率的に、かつ自然な形で解く手法についてはまだ決定的なものは得られていない。従来の多くのシステムでは、制約をメソッドの形に変換する、制約オブジェクトという概念を導入する、制約を全て大域的なものとしてオブジェクト指向とは全く別の解法を用いるなどの手法を採用している。これに対しここでは、効率的で、かつオブジェクト指向の考え方や親和性がよく、インクリメンタルに制約を扱えるという特徴を持つ、オブジェクト間の制約式の伝播による制約充足方式を提案する。

2 制約に基づくオブジェクト表現

一般にオブジェクトは単独で存在することはまれで、全体・部分関係による階層構成を成す。そして複数のオブジェクトには依存関係が存在し、それを属性間の制約で表現できる。ここで、複数のオブジェクト間の制約はそれらを部分(構成要素)とする全体を表わすオブジェクトに記述するものとする。

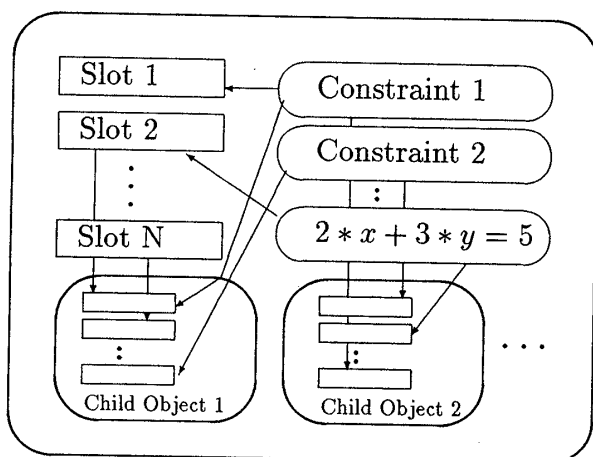


図 1: 制約によるオブジェクトの構造

オブジェクトは図 1 で示すように、属性値を格納するスロット (Slot)、部分を表わす子オブジェクト (Child

Object)、スロット間の制約式 (Constraint) で表現する。属性には公開属性と内部属性がある。外部から参照できるのは公開属性のみであり、内部属性、制約、部分構造についての情報は隠蔽される。従って制約式で参照できるのは、当該オブジェクトの属性とその子オブジェクトの属性のみである。なお、子オブジェクトに対し、全体を表わすオブジェクトを親オブジェクト (Parent Object) と呼ぶ。

従来のオブジェクト指向プログラミングでは、オブジェクトの性質や他のオブジェクトとの関係を、メソッドにより手続きで表現しなければならなかったが、この方法は記述量及び複雑さという点で問題がある。それに対し、関係やふるまいを制約記述することにより、簡潔でわかりやすい表現が可能となる。また制約についてもクラスの継承機能を利用することができ、モジュール性に優れた記述ができる。

3 制約伝播による問題解決

制約には、オブジェクトのクラス定義において静的に記述される制約と、個々のインスタンスに動的に付加される制約がある。前者によってそのクラスに属する全てのインスタンスが満足すべき制約、すなわち一般的な対象の構造、性質を記述し、後者によって、対象となるオブジェクトを具体化するための知識を与える。後者は通常のオブジェクト指向言語のメッセージに対応させることができる。また、オブジェクトの属性値の設定も、属性と定数の等式で表現される制約の付加として扱うことができる。

制約付加に対するオブジェクトのふるまいは制約の充足過程である。すなわちオブジェクトは常に制約充足状態を維持するように動作し、これにより、制約を満たす属性値の組み合わせを求めることができる。

複数のオブジェクトの属性間の制約は、それらのオブジェクトを部分として持つ共通の親オブジェクトに付加される。付加された制約は記憶されている制約式とともに後述するアルゴリズムに従って局所的な制約に簡略化される。

この制約の局所化は、大域的な制約式の集合から、1つの子オブジェクトのみに関係する局所的制約式へ、そしてさらに1つの属性値のみに関する制約式へと変形することである。ここで、1つの子オブジェクトのみに関する制約はその子オブジェクトに伝播することにより、制約解決を部分問題に分割することができる。そして局所化により最終的に属性値を決定することができる。

なお、この過程で得られた属性の値は、それを参照している親オブジェクト中の制約式があれば、その親オブ

ジェクトへと伝播され、これにより部分の制約解決の結果は全体へ反映される。

次に実際の制約の充足の過程について述べる。

4 全体部分関係における制約解決戦略

基本的に、オブジェクトは最初に述べたように、属性値とその間の制約式の階層的表現によって記述される。そこで、全体・部分関係を独立したオブジェクトのリンク構造で考え、問題解決をその間の制約式の伝播で実現することにする。ここで、全体を表現する親オブジェクトは、部分を表現する子オブジェクトを参照できる。また、子オブジェクトはどの属性が親オブジェクトの制約式により参照されているかを知っているものとする。

今回は問題を単純化するために、等式表現された制約のみを扱う。従って制約式の集合は連立方程式であり、制約の局所化を変数の消去法などの手法により実現できる。

以下に制約式付加時の処理手順を示す(図2)。

1. 付加された制約式を含むオブジェクトにおいて、以下に従って制約式の変形を行う。
 - (a) 1つの子オブジェクトのスロットのみを参照するような制約式を導く。
 - (b) 子オブジェクトのスロットを全く参照しない制約式を導く。
2. 1.の結果に基づき以下の処理を行う。
 - (a) 1の(a)で得られた制約式は対応する子オブジェクトに伝播する。
 - (b) 1の(b)でスロット値が決定されたら、それをスロットに格納する。また、そのスロットが親オブジェクトの制約式から参照されていれば、それを親オブジェクトに伝播する。
3. 制約を付加されたオブジェクトは1.から同様に処理を行う。

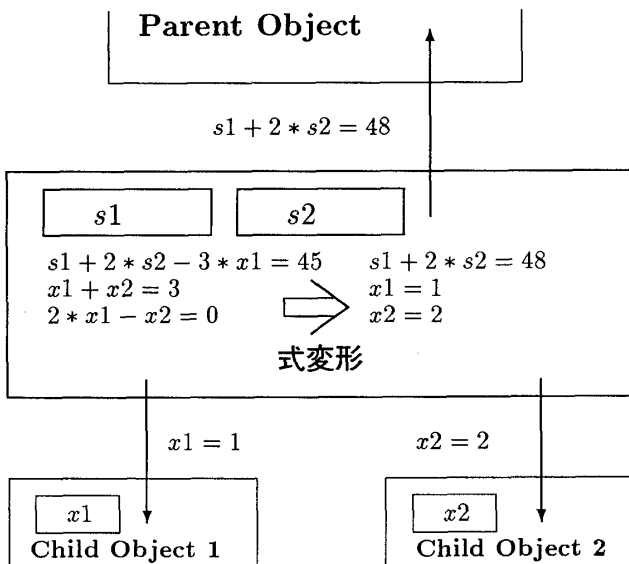


図2: オブジェクト間の制約伝播

ところで、部分を表現する子オブジェクトが複数の親に共有されていることがあり、制約式の関係が図3のようにループを作る。上記のアルゴリズムでは解くことができないことがある。この場合にはループを作っている制約式を1個所に集め同時に評価すればよいが、これは、制約式は隣接する子オブジェクトのスロットのみの参照しか許されないという制限に反する。

そこで、ここではスロット値を一時的な記号に置き換えて伝播し、共通な場所で消去することにより解く手法を提案する。すなわち、複数の親から参照されているスロットが存在する場合や、異なる親から参照されているスロット間に依存関係が存在する場合は、そのスロット値を記号化し、値が得られたものとして親に伝播する。そして制約式の評価において、記号を含む制約式が現われた場合は、優先的に記号の消去を行うことにより、自然に制約式を解くことができる。

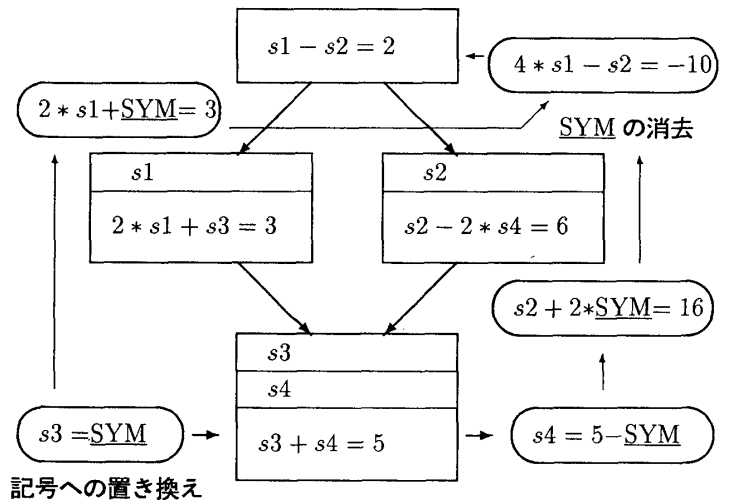


図3: 制約のループの解決方法

5 おわりに

以上、オブジェクト間の制約伝播による問題解決について述べた。本方式はオブジェクトの全体部分関係に着目し、制約の局所化によって制約解決の効率化を図ったものである。そして、この制約充足方式を制約に基づく知識表現システム FREEDOM[2] [3] の制約充足に適用した。今後の課題として、扱う制約の範囲を等式以外へ拡大すること、制約充足の並列化などがある。

参考文献

- [1] Borning, A. "The Programming Language Aspects of ThingLab, a Constraint-Oriented Simulation Laboratory", ACM Trans. on Prog. Lang. and Syst. vol.3, no.4, pp353-387,(1981)
- [2] 横山孝典 "設計対象記述のための知識表現システム FREEDOM の提案", 情報処理学会研究会資料, 88-AI-60, (1988)
- [3] 横山孝典, 佐塚秀人 "制約に基づくオブジェクト指向知識表現システム FREEDOM における制約充足方式", 情報処理学会第 38 回全国大会予稿集, 7D-5, (1989)