6L-5 制約表現とその適用実行メカニズムに関する一考察

永井　保夫

（財）新世代コンピュータ技術開発機構

1. はじめに

最近，設計問題を対象としたエキスパートシステムの研究が事例を主体として盛んになり始めた。我々の研究はこのような設計問題を対象にしたエキスパートシステムのアーキテクチャの明確化とそのための構築支援ツールの実現が目標である。そのために，実際のアプリケーションによる設計過程のモデル化を制約表現に基づいた問題解決という側面から研究を行っている。ところで，このような制約表現をロジックプログラミングの枠組みに取り入れたアプローチが近年いくつか提案されている。これらの研究はプログラム言語としてのアプローチではあるが，制約表現に基づいた問題解決機能を考える上での基礎をなしており，設計型エキスパートシステム実現の上での一つの要素技術とみなす事が可能である。これらの制約表現は適用領域に対して，一様な評価に基づいて適用実行されるために不十分であると考えられ，設計問題という適用領域を念頭に置いた場合には実際に制約表現自体も分類化され，そのための適用実行（問題解決）メカニズムも異なる事が多い。本論文では，合成型問題の代表例である設計問題を対象にしたエキスパートシステムのアーキテクチャ実現上要求される制約表現に基づいた問題解決機構（制約論理プログラミングも含む）を研究していく第一段階として，制約表現の分類化とその適用実行メカニズムについて考察する。

2. 制約表現の分類

制約表現について以下の項目によって分類した。

(1) 生成方式による分類
(2) 重要性による分類
(3) 適用範囲の限定による分類
(4) 伝播される変数情報による分類

生成方式による制約には，静的な制約と動的な制約が考えられる。静的な制約とは，予め静的に与えられた制約であり絶えず一定で変化しないという不変性を保持している。動的な制約とは，ユーザによるインタラクションやシステムより制約が与えられることで，静的な制約とは逆に状況に応じて適用範囲や制約自身が動的に変化（追加や削除）するものである。これは Incompleteな知識と解釈可能であり，制約の変化による知識ベース中の信念の変化の管理を行うため，Truth maintenance system 的な機能が必要となる。重要性による制約には，義務的な又は必須な（obligatory or requisite）制約と示唆的（suggestive）制約が考えられる。その場合，すべての制約が対等なものとして選択・実行されるわけではない。つまり，その重要性はコンテクストや時間などの場合に依存する。義務的な制約はすべてが満足されねばならぬもので，一般的には最に与えられるものである。示唆的な制約は弱い制約（weak constraint）とも言われ，選択枝より最良の枝を選択するためのガイドとして使用される。これはルール形式で表現可能なものもあり，プライオリティーなどの重み付けが行われる。適用範囲の限定による制約には，ローカルな制約とグローバルな制約が考えられ，探索空間における状態（state）の評価に使用される。ローカルな制約はあるモデル，オブジェクト，プロセスなどの中で状態が変化するものに対して探索を行うために使用し，有効範囲もそのなかで閉じているものである。グローバルな制約は，ある状態の評価をするために，ローカルな制約だけを適用するのではなく，適用範囲を限定しないで関連するすべての制約を利用して評価をするものである。例えば，これは解空間を分割化して探索していく場合には現在の状態に至るまでに関連した状態に適用されたすべての制約を考慮したり，分割された解空間同志の評価を行わなうことに相当する。伝播される変数情報による制約には変数に対して，値を伝播する制約と値の取り得る領域を区間として伝播する制約が考えられる。現在，制約論理プログラムやCONSTRAINTといった殆どのシステムでは，値を伝播する制約だけを取り扱っており，value inference 又は，expression inference と呼ばれている。値の取り得る領域を区間とし伝播する制約は，不等式として表現された制約が変数をコンスタントな値ではなく区間集合とみなして伝播するものである。設計問題を考えた場合，従来のオペレーションズリサーチを利用して解を求めるような問題を含んでいる場合が少なくないので，区間集合による制約伝播という統一的な枠組みで取り扱う機能をアーキテクチャに対して考慮する事は非常に重要であると考える。

制約表現の分類を行ったが，実際にはひとつの制約表現が上記の機能項目を複数有しているものと考えられる。但し，現段階では以上の項目によりすべての制約表現を分類するには不十分であり，尚検討を要する。

3. 制約表現の有効な適用実行メカニズムに必要な機能

制約表現の有効な適用実行メカニズムに必要な機能を以下に示す。

(1) 制約伝播（constraint propagation）機構とその制御
　※ 制約間のインタラクション
　※ 制約の最小拘束原理（least commitment）
(2) 制約の緩和（relaxation）と 選択（selection）
(3) dependency の保持と管理機能
(4) 制約評価のモニタリング機構

制約伝播機構は制約を満足していく過程で，制約中の変数に値が割

Towards a constraint description and its application mechanism

Yasuo NAGAI

ICOT

り付けられると，その変数を媒介として他の制約中の変数の値も決定されていくというメカニズムである．代表例として，CONSTRAINT で導入されているデータの流れによる伝播法やSussmanらの連立方程式の消去法を基にしたthe method of constraint propagation がある．前者はローカルな制約の伝播といわれ，後者はローカルな制約の伝播だけでは問題が解決されない場合で，constraint solverにより処理がなされる．特に，データの流れによる伝播法では，TMS (Truth Maintenance System) のCSP(constraint satisfaction problem) と同様に制約のトレード・オフにより伝播が十分に行われるとは限らないという潜在性を含む．そのために，制約伝播の制御についての戦略が必要であるのは明白である．その際に，考慮すべき項目としては制約間のインタラクションと制約の最小拘束原理がある．制約間のインタラクションは制約伝播機構を実現していく上で不可欠なものである．step-wise refinement 法による設計においては分割されて解かれる部分問題間の制約のインタラクションが重大な問題となる．MOLGEN では部分問題間でのインタラクションを最小にするように設計がなされている．実際の設計問題に対しては，まずその問題が部分問題に分割された時の制約間のインタラクションについて考察する事が必要である．制約の最小拘束原理とはできるだけ制約の評価を遅延する形で計画の詳細化を進めていき，必要な時の評価を行っていくものであり，制約間のインタラクションと同様にMOLGEN で用いられている．制約の緩和と選択は前述の弱い制約に対して適用されるものである．制約の緩和は指定された制約に対するalternativeを求めることである．つまり，ある制約が満足されない(= fail)段階でその制約と同レベルか又は低レベルの制約をalternativeとして求めるものである．制約の選択は，競合する(conflicting) 制約が存在する時にどれを選択するか，すなわち制約の解釈とみなされる．このように，制約の緩和と選択という機能はプランニングの問題として定式化することが可能であると考えられる．dependency の保持とその管理は制約伝播機構により制約中の変数に値が伝播されていく過程において，値に対する contradiction が発生した時にその解消を行うためや伝播値の説明を行うために必要とされる．制約評価のモニタリング機構は制約表現に適した問題解決機構には欠くべからざるもので，制約の consistency をチェックするものである．これはデモン又は，付加手続きによってある程度実現可能である．

4. アーキテクチャのイメージ(位置付け)

いままで述べてきた制約表現とその適用実行（問題解決）メカニズムが設計型エキスパートシステムのアーキテクチャを考えた場合に，どのような位置付けとするべきかについて示す．方針としては従来のエキスパートシステムにおけるアーキテクチャー（ブラックボードモデル）をできるだけ損なわない形で上記で述べた概念を取り入れる事とする．つまり，ルール・ベースシステムとフレーム・ベースシステムが有しているアーキテクチャーは前提として仮定されているものとする．その上に，新たに制約を記述する言語とその適用・実行（問題解決）機構を付加する予定である（図を参照の事）．その際にはフレーム・ベースシステムと制約表現との結合メカニズムについて考慮されねばならない．従来のエキスパートシステムではこのような機能がシステム構築者には全く提供されていないのが現状であり（かなり適用・対象に依存するものだから），両者のインターフェイスを考案することも研究として価値があると思われる．

5. おわりに

制約表現とその適用実行メカニズムについて報告してきた．設計やスケジューリングといった合成型問題を対象としたエキスパートシステムを考えていく場合には制約と言う概念の明示的な導入と分類化が不可欠なものであると思われる．今後は様々な適用対象（特に機械設計）についての実際例を念頭に入れて研究を進める方針である．

参考文献

[Descotte 85] Yannick Descotte and Jean-Claude Latombe,Making Compromisesaaong Antagonist Constraints in a Planner,Artificial Intelligence 27, 1985

[Dincbas 86] M. Dincbas, CONSTRAINTS, LOGIC PROGRAMMING and DEDUCTIVE DATABASES, France-Japan Artificial Intelligence and Computer Symposium 86

[Fox 83] Mark S. Fox, Constaint-Directed Search: A Case Study of Job-Shop Scheduling, CMU-RI-TR-83-22

[Harris 86] David R. Harris, A HYBRID STRUCTURED OBJECT AND CONSTRAINT REPRESENTATION LANGUAGE, AAAI'86

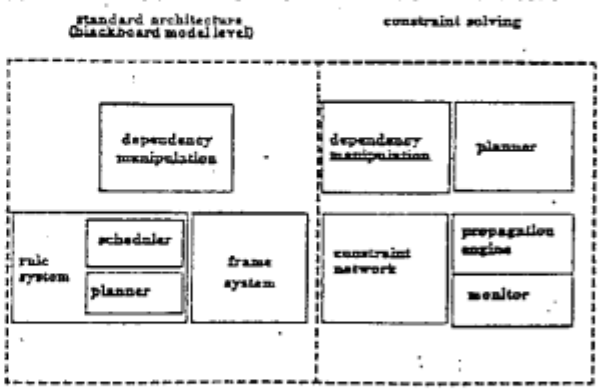
[Stallman 77] Richard M. Stallman and Gerald J.Sussman,Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis, Artificial Intelligence Vol 9, 1977

[Stefik 81a]·M. Stefik, Planning with Constraints (MOLGEN: Part 1), Artificial Intelligence, Vol.16, 1981

[Stefik 81b] M. Stefik, Planning and Meta-Planning (MOLGEN: Part 2), Artificial Intelligence, Vol.16, 1981

[Sussman 80] G. Sussman, L. Steel, CONSTRAINTS: a language for expressing almost-hierarchical descriptions, Artificial Intelligence, Vol.14, 1980

[長澤 86] 長澤 熙，設計システムのための知識表現と推論機構に関する研究，九大学位論文，1986



General Configuration (Standard Architecture Base)