

論文の内容の要旨

論文題目 動的構造変更機能を有する
基本設計支援用言語に関する研究

氏 名 今 村 聡

研究の背景

設計過程は概念設計、基本設計、詳細設計の設計段階から構成されると考えられている。これらの各設計段階における技術内容は大きく異なり、計算機支援には、それぞれ異なるアプローチが必要になると考えられる。

概念設計の内容は、対象によってかなり異なる。要求仕様もその実現手段も明確に与えられない建築、意匠設計などの分野では、それらを明確にする作業が行われる。一方、要求仕様が要求機能として明示される場合は、要求機能に対応する実現手段の概略構造を構築していく作業が行われる。

詳細設計は、設計対象の製造が可能なレベルまで詳細を決める段階で、製造関連情報の扱いが重要になる。

本研究は基本設計支援を主な対象としているので、ここでは、基本設計についてより詳細に検討する。基本設計は、要求仕様を満たす実現手段を定量的に評価し、実現手段の詳細を決める段階である。機械設計では、部品構成と機能に関係するパラメータのほとんどがここで決定される。シミュレーション、コンフィギュレーション設計、パラメトリック設計の三つの計算機支援方法が基本設計に対して多く用いられている。このうち、コンフィギュレーション設計及びパラメトリック設計は、設計解を求めるための技術であるが、シミュレーションは設計解の解析評価技術である。

コンフィギュレーション設計は、あらかじめ与えられた構成要素の集合から、

制約を満たす組合せを求めるもので、二つのレベルがあると考え。第1レベルは機械の構造は固定されており、構成部品の選択のみ行う設計、第2レベルは機械の構造、構成部品ともに決まっておらず、両者を並行的に決めていく設計である。従来のコンフィギュレーション設計システムの多くは第1レベルのものにとどまっておらず、第2レベルをサポートできるものは多くない。

パラメトリック設計は、対象の特性を表すパラメータ集合とそれらの関係を表す数式、論理式により定義されるパラメトリックモデルを用いて行う設計である。第1レベルのコンフィギュレーション設計に類似しているが、集合からの選択に依存しない値の決定も行われる点が異なる。パラメトリック設計は、あらかじめ適切な設計式を登録しておけば、設計対象の構造が確定している定型的設計を効率的に処理することができるが、設計対象の構造が未確定ないし変化する可能性のある設計への適用は困難である。

研究の目的

設計は要求機能を満たす構造を求めていく過程であるが、それは直線的に進むとは限らず、設計対象の概略構造がほぼ固まっているべき基本設計段階でも、軽微な条件変更で、構造を大幅に変更する必要が生じることがある。あるいは、概念設計段階で設計対象の概略構造を決定できるとは限らず、機能の定量的検討作業を相当進めてからでないと適切な構造を決定できない場合もある。このような性格を持つ設計に対しては、パラメトリック／コンフィギュレーション設計による設計支援は困難であった。

一方、個別のパラメトリック／コンフィギュレーション設計を見てみると、パラメータ間の関係記述力が不十分、設計対象の記述性が低い、設計解決能力が不十分、設計対象のモジュール構造の組み立てが煩わしいなどの問題がある。

以上の背景から、本研究の研究目的を次のように設定した。

(1) 基本設計支援技術に位置づけられる、パラメトリック／コンフィギュレーション設計技術に関わる以下の課題を総合的に解決する。

- (a) パラメータ間の関係記述力
- (b) 設計対象の記述性
- (c) 設計解探索能力
- (d) 設計対象のモジュール構造と組立

(2) 設計モデルの構造が未確定ないし動的に変化する設計に対して適用できるパラメトリック／コンフィギュレーション設計技術を開発する。

研究のアプローチ

本研究のアプローチは、上に掲げた研究目的を達成することが可能な設計支援用計算機言語 FDL (Flexible Design Language) を開発することによる。設計言語は次に示すような技術を導入したものになる。

(1) オブジェクト指向言語

物理実体との対応がとりやすいといわれる、オブジェクト指向言語の形式を採用する。オブジェクト単位でモジュール構造を持たせ、is_a 継承、has_a 関係の効果的な活用により、プログラムの再利用性を高める。データタイプも設計で必要となる多様なタイプを用意する。

(2) 多様な制約表現

オブジェクト間の情報交換に制約伝播を利用する。制約表現は方程式、不等式、ルール（起動条件付き制約記述）、Prolog 述語が使えるものとし、制約表現の多様性を確保する。また制約の宣言的記述の能力とオブジェクト指向の特徴を生かして、高い記述性を確保する。複雑な制約問題を効率よく解くことのできる制約解決系を開発する。

(3) 関係データベース機能

標準、カタログデータなどの表形式データを扱いやすいように関係データ検索機能を、制約と連動させた形で提供する。

(4) 動的構造変更用メタ操作

設計モデルの構造変更操作を動的に（プログラム実行系の上で、それまでの設計結果の連続性を保ちつつ）行うため、構造共有、構造交換、構造追加／削除などのメタ操作を導入する。構造融合はオブジェクト同士を融合することにより、モデルの組み立て作業を容易にする。構造交換はシャフトや軸受けを異なるタイプのモデルに交換するなど、オブジェクトの交換に使う。また、オブジェクトの追加や、削除をデータ整合性を維持しつつ行うことができる。

(5) 動的オブジェクトの継承関係管理メカニズム

クラス、インスタンスオブジェクトが動的に変更されてもオブジェクト間の継承関係に矛盾が生じないように、was_a リンク、was リンクを導入する。was_a リンクはインスタンスオブジェクトの動的構造変更により親クラスと対象インスタンスの間の is_a リンクの代わりに使われるもので、構造変更操作の履歴情報を使って、元親クラスからの選択的な情報継承を可能にする。was リンクはクラスオブジェクトの継承情報管理のために使われるもので、メカニズムは was_a リンクと同等である。

(6) 動的構造変更を自動的に行うための配置演算子

起動条件と動的構造変更操作列の組み合わせで定義される配置演算子を導入し、あらかじめ想定できる設計モデルの動的構造変更の自動化を可能にする。

