

Title	パラメータ化プログラミングにおけるモジュール変換に関する研究
Author(s)	浦上, 貴裕
Citation	
Issue Date	1997-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1069
Rights	
Description	Supervisor:二木 厚吉, 情報科学研究科, 修士

パラメータ化プログラミングにおける モジュール変換に関する研究

浦上貴裕

北陸先端科学技術大学院大学 情報科学研究科

1997年2月14日

キーワード: parameterized programming, reusability, module graph, CafeOBJ.

1 はじめに

規模や複雑さが増大するほどソフトウェアの構造化は不可欠なものとなり、プログラミング言語のモジュール化の機能は重要なポイントになる。自然言語で仕様を記述する際のモジュール構造の与え方は、通常章や節に分ける形で実現している。この場合、章や節の間での参照による依存関係が仕様全体の構造を決定する。全体の構造の見通しを良くするため、目次や索引が付けられる。

自然言語における目次や索引の利用方法と仕様の構造の関係について考える。例えば概略の記述部分から詳細が述べられた部分へ向けて索引付けがされている場合、これは詳細化の意味での階層構造の形成であると見ることができる。また、ある機能を利用する部分で行なわれるその機能の定義部分への索引付けは、機能的な依存関係を表しているとみなすことができる。さらに、仕様の再利用性の考慮はパラメータ化で実現される。ある機能の完全な定義を仕様の中で行わず、一部実装に依存するような記述がされる場合、これは機能のパラメータ化であると考えることができる。

抽象データ型の仕様記述法はプログラムをモジュール化する概念であるということが出来るため、本研究には抽象データ型の仕様記述法を起源とする代数仕様言語を用いる。抽象データ型は、データの集合とその上に定義された演算の集合を形成するデータ型の抽象である。抽象とは操作がどの様に定義されているかといった詳細の部分をブラックボックスとし、操作方法のみを明かす概念である。代数仕様におけるシステムの仕様は、モジュール(抽象データ型)の組み合わせで形成される。モジュールを組み合わせるための機構としては、自然言語で記述された仕様の場合と同様、あるモジュールが別のモジュール

ルを参照する構文をはじめ、パラメータ付きのモジュールを記述する構文などの実現が望まれる。しかし、パラメータ機構などの整備が進み複雑にモジュールの参照や詳細化が行なわれると、結果としてモジュール構造が複雑になりコード上で全体の構造の見通しが付きにくくなる場合がある。本稿では、こうして複雑になった仕様の詳細化の支援を目的として、自然言語の仕様における目次や索引と同等の役割を果たすモジュールグラフの定義を行なう。その後、モジュールグラフの利用によるプログラムの可読性の向上について検討を行なう。実際にはプログラムで用いられているデータ構造の挿替えをモジュールグラフ上で実現していく。

2 代数仕様言語 CafeOBJ

本研究では代数仕様言語 CafeOBJ を用いる。CafeOBJ は順序ソート代数に基づいた項書換え規則に従い、仕様の実行による検証を行なうことが可能である。記述する仕様には CafeOBJ の前身の OBJ2 と OBJ3 で本格的に導入されたパラメータ化プログラミング機構を用いることができる。例えば基本的なデータ構造であるリストや木構造などは、その要素となるソートそれぞれにつき定義する必要があるが(例えば自然数のリスト、実数のリストなど)、この機構の利用によりデータ構造の定義に必要であると思われる性質や操作のみ記述すればよくなるのである。モジュールをパラメータに束縛することで実体を作り出し、実行可能なモジュールを形成する。データ構造のように様々なソートに対して同様の定義を行なう必要があるモジュールを構築するには非常に有効な機構である。実パラメータとして束縛されるのはパラメータの制約を満たすモジュールである。制約を満たす様々なモジュールについての特定例を作り出すことが可能となることから、パラメータ化プログラミングを行なうことでモジュールの汎用性、再利用性および保守性が高められる。パラメータ付きモジュールの汎用性はモジュールの表現力を高めることにもつながる。例えばデータ構造として列を表現するモジュールをパラメータ付きで用意し、このモジュールを用いて並べ替えを行なう汎用的なモジュールを定義することが可能となるからである。この際に列のモジュールとそれを輸入している並べ替えのモジュールとの間で多重のパラメータ化が形成される。並べ替えのモジュールを特定化することで列のモジュールも同時に特定化されることから、これを多重の特定化と呼ぶ。

CafeOBJ では参照するモジュール要素の名前替えを行なうことにより、同一モジュールを構造の等しい別のモジュールとして参照することが可能である。多重の特定化や名前替えという機構はモジュールの表現力を高めるため仕様記述者にとっては非常に便利であるが、これらの際限ない利用がコード上からモジュール構造の可読性を失う原因になる。

3 モジュールグラフ

モジュールグラフの定義を行なう。モジュール構造は基本的にはモジュールを頂点とし参照関係を辺とした有向グラフで表現する(図1,2)。パラメータ付きモジュールはパラ

メータと制約モジュール名を楕円で囲んだパラメータ部を用意し、モジュールの上部にこれを記述する(図3)。不確定モジュールと特定化後の確定モジュールで形状を変える工夫を行なった。形状を変えた利点は、完成したプログラムの静的なグラフ化のみでなく、パラメータ付きモジュールを用いたプログラムの構築過程を動的に示すことが可能となったことがいえる。特定化のグラフ表現は「仮パラメータへの制約を満たしたモジュールをパラメータに束縛する」という動作のグラフ化である。あるモジュールが制約を満たしていることはビューを用いて宣言する。モジュールグラフでもビューのグラフ化が必要となる。ビュー宣言はモジュールと制約モジュールとのマップの記述である。グラフではビュー名を菱形で囲み実モジュールと制約モジュールを結ぶ線から矢印を引くこととした(図4)。特定化はここで定義したビューの菱形部分からパラメータ付きモジュールのパラメータ部へ矢印を引くことで表現する(図5)。多重のパラメータ化はモジュールの内部構造としてパラメータ付きモジュールを特定化せずに利用することである(図6)。内部のモジュールが特定化されていないため、外部モジュールも同様のパラメータを持つ。モジュールグラフではパラメータ付きモジュールの内部にさらにパラメータ付きモジュールを描き、パラメータ部同士を矢印で結ぶ。本論文で定義したモジュールグラフはCafeOBJのモジュール定義のヘッダ部分に記述された情報に基づく。ヘッダ部分に記述される情報は、モジュール名、参照モジュール名と参照の方法、パラメータと制約モジュール名、名前替えである。モジュールグラフではこれらの情報が全て再現されている。

4 成果

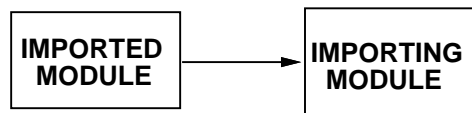
上の定義によりパラメータ付きモジュールや多重のパラメータ化を伴ったモジュールのグラフ化が可能となった。複雑なモジュール構造を持つ仕様の可視化が行なわれたことから仕様の可読性を高めることができたと言える。グラフ化された仕様からは、そこで用いられているモジュールの参照と利用の形態を読み取ることが可能となることから既存モジュールの再利用を促す役割も持つことになる。本稿ではデータ構造を利用した様々なプログラムをモジュールグラフで表示し、モジュール構造に変更を加えることなくデータ構造の挿替えを実行する可能性について研究を行なった。モジュール構造に変更が加わることはプログラムの安全性の損失につながる。このため、プログラムの保守作業と同時にモジュール構造の変更を確認できることは保守の際の安全性の確保に貢献することになる。

研究の結果、グラフ上でデータ構造の挿替えを実践することはパラメータに対する制約の変更の可視化につながるということが分かった。多重のパラメータ化を行なうモジュールの内部モジュールに変更が加えられた場合、内部モジュールの制約が外部モジュールよりきついものになるとモジュール定義自体が不可能となる。そのため外部モジュールの制約を内部と同様のきつい制約に変更する必要が生まれる。モジュールグラフによりモジュール変更の状態を示す一方で、制約変更の流れを示すことが可能となったり、保守作業時のプログラムの安全性を高めることができるようになった。

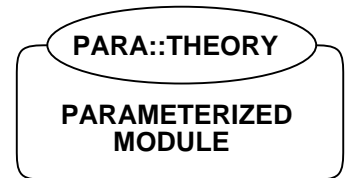
Module Graph Definition



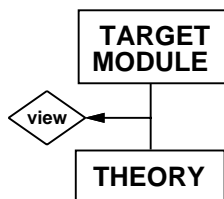
(1) module



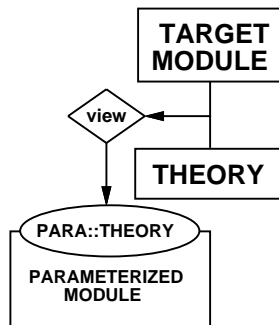
(2) module importing



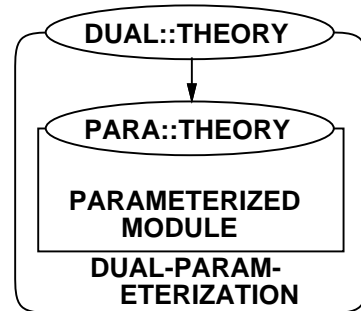
(3) parameterized module



(4) view



(5) instantiation



(6) dual-parameterized module