様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２８ 年　 ６ 月 １６ 日現在

機関番号：１２１０２
研究種目：基盤研究(C)（一般）
研究期間：2013～2015
課題番号：２５３３００７６

研究課題名（和文）クラス構造のメタ記述による合成を許すプログラミング言語機構

研究課題名（英文）Unified Composition Mechanism for Abstraction Units Based on Pluggable Methods

研究代表者
久野　靖（KUNO, Yasushi）

筑波大学・ビジネスサイエンス系・教授

研究者番号：００１７００１９

交付決定額（研究期間全体）：（直接経費）　　1,300,000円

研究成果の概要（和文）：オブジェクト指向はプログラミング言語において主流となる流派であり、その中で継承機構はプログラムコードの柔軟性を高めるために多くの箇所で活用されている。しかし近年、継承機構では不足する機能を補うものとして型パラメタ機構やアスペクト指向などが提案され、その結果言語が複雑化していた。本研究ではこのような複雑さを解消する方向として、少数の基本プリミティブを持ち、これを通じて継承やアスペクトなどの動作を作り出して行くような言語設計を提唱するとともに実装・評価した。当該言語においてはクラスに相当するプログラム単位に対してメソッドやその断片をメタプログラミング記述を通じて挿抜することで継承などを記述できる。

研究成果の概要（英文）：Object-orientation is one of mainstream concept in programming languages, and inheritance is used extensively to utilize flexibility of the language. However, recent progress in programming language research has lead to many new language mechanisms such as generics or aspect orientation in addition to traditional inheritance. Aim of this research is to unify those multiple language mechanism and design a language with small and minimal "primitive set" of mechanisms. As the result of the research, the author has proposed a language with unified class-composition mechanism, in which methods associated with classes can freely be extracted through meta-programming facilities of the language.

研究分野：計算機ソフトウェア

キーワード：プログラミング言語　オブジェクト指向　継承　型パラメタ　アスペクト　メタプログラミング

１．研究開始当初の背景

(1) 現在のプログラミング言語における中心的パラダイムの１つであるオブジェクト指向は大きな成功を収め、多数のシステムの開発に用いられていた。

(2) しかしオブジェクト指向言語、とくにクラス方式の言語が提供する機能は継承を中心としてかなり恣意的に定められたものである。具体的には、継承とは「親クラスを指定してインタフェースを包含する型とし」「親クラスからメソッドの実装ならびにインスタンス変数定義を取り込んで来て」「ただし自クラスにおいて再定義しているメソッドについてはそちらを優先する」という複雑な操作から成り立っている。

(3) このような固定的な継承の機能でカバーできない部分について、新たな言語メカニズムが多く提案され広まって来ている。その代表的なものとして、型パラメタ機構が挙げられる。またアスペクト指向と呼ばれる言語研究の分野では、オブジェクト指向に収まらない部分をカバーできるような言語メカニズムの研究が行われ、AspectJ のような言語の実現に至っている。

(4) しかしこれらの結果として、オブジェクト指向言語の機能はより複雑で難しいものとなっている。このような複雑さは好ましいものではなく、より簡潔な言語設計に至る方法の探究が必要とされている状況であった。

２．研究の目的

(1) 本研究の第 1 の目的は、上述のように複雑化したオブジェクト指向言語の設計に対し、より基本的なプリミティブ群を用いて現在使用されている機構群を作り出せる言語を提案することで、簡潔な言語設計を可能とすることを示す点にあった。

(2) 本研究の第 2 の目的は、その過程を通じて実際に言語を設計し実装することにより、そのような新たなプリミティブを持った言語の使いやすさや柔軟性などに関する知見を得るところにあった。

(3) 本研究の第 3 の目的として、可能であれば、既存の言語機構の再構成にとどまらず、提案するプリミティブを用いて新たな、かつ有用な言語機構を作り出し、その有用性の評価を行うことも含まれていた。

３．研究の方法

(1) 既存言語の整理および分析 --- まず既存のオブジェクト指向言語ならびに関連するような拡張機能を持つ言語の調査をおこない、どのような言語メカニズムがあるかについて整理を行った。対象となる言語メカニズムとしては、継承、型パラメタ、アスペクトを最低限含むものとした。

(2) 新たな言語機構の検討ならびに考案 --- (1)において検討した主要な言語メカニズムと同等の効果を実現できるような、少数の基本的かつ汎用性の高い言語プリミティブの集まりをデザインする。この集まりににによって継承、型パラメタ、アスペクトなどの機能を記述し構築できることが必要となる。

(3) 言語としての設計ならびに実装 --- (2)において検討した言語プリミティブ群を備えたプログラミング言語を設計するとともに実装することで、その具体的な動作や実装上の問題点を洗い出す。

(4) 考案した言語の評価 --- (3)において実装した言語の評価を行う。評価の１つは継承、型パラメタ、アスペクトなどの主要な言語メカニズムを実際に作り出し利用できることの確認である。また評価のもう１つは、この言語メカニズムを用いて有用な言語メカニズムを新たに作り出すことができることの確認である。

４．研究成果

(1) 既存のオブジェクト指向言語およびその拡張機能の整理分類について --- オブジェクト指向言語で古くから使われて来た言語メカニズムは冒頭に述べたように継承であるが、その後新たに加わったものとして型パラメタ機構(C++言語ではテンプレート、Java 言語ではジェネリクス機構)があげられた。またオブジェクト指向の枠内に収まらない機能分解を記述するための枠組みはアスペクト指向の名称のもとに多く研究されており、AspectJ などの言語として具体化していた。

(2) 各種機能の整理の分類について --- 型パラメタについては、プログラムコード中の名前を書き換えることで１つのソースコードを多様な場面に適応させる、という汎用的なメカニズムであると考えられた。型パラメタ機構はオブジェクト指向言語の中では比較的後から普及した言語メカニズムであるが、その内容については比較的言語の根幹部分に相当すると考えてよい。これに対し、継承やアスペクトはプログラム単位(通常のオブジェクト指向言語では「クラス」と称される)のインタフェースならびに実装の要素(メソッドやそのメソッドにより実行されるコードの断片、ならびにプログラム単位に付随する変数など)を他の単位からコピーしてきたり部分的に組み合わせたり系統的に書き換えるなどの操作として一般化できた。具体的には、継承では子クラスは親クラスのサブタイプとなるとともに(インターフェース的側面)、親クラスからメソッドの実装を取り込んで来る機能として位置付けられる(実装上の側面)。さらに、子クラス側でメソッドの再定義(overriding)を行う場合は、取り込まれて来た実装は上書きされてキャンセルされる。一方、アスペクト指向では加工対象となるクラスに対し、単独のメソッドをよそから持って来て挿入することもあるが、むしろ既存のメソッドの冒頭や末尾に付加コードを挿入することが多い。

(3) 研究成果として提案した枠組みについて --- 上記の検討の結果、汎用の機能であるパラメタ機構はその根源的・汎用的な性質を考え、言語組み込みで持つことが良いと考えた。これに対し、プログラム単位における要素の組み立ては「プログラムを加工するコード」(メタ記述)により構成する、という基本的なアイデアを思い付き、それに基づく言語について検討した。その加工の内容は、既存の単位に存在しているメソッドを持って来て単独で新たな単位に挿入する場合と、単独ではなく単位が既に持っているメソッドの冒頭や末尾に付加する場合とを使い分ける。これにより、継承とアスペクトの両方が実現可能となる。なお、このような言語機構の枠組を、メソッド本体が「コードのかたまり」として働き、そのかたまり単位でのコピー・挿入や付加を行うことから、「挿抜可能メソッドに基づく統一的な抽象単位復号機構」と名付けた。

(4) 新たな枠組みに基づく言語の設計と実装 --- 上記の枠組みに基づく言語「o3」を設計し実装した。この言語では、プログラム単位は従来の言語におけるインタフェースやクラスに対応した単位に加え、メタ記述を用いて新たな単位を組み立てる「メタクラスタ」およびそこから参照される共通の手順を定義した「メタ手続き」を持つ。この言語の処理系は SableCC コンパイラコンパイラならびに Java 言語を用いて作成している。処理系そのものは抽象構文木をまず組み立て、それをもとに抽象単位を内部データ構造として構築した上で、メタなコードについてはそのコードをツリーインタプリタにより解釈実行しながら API の呼び出し(下記)をおこなうことで、メソッドの挿抜に相当する操作を実行していく。

(5) メタ操作のために必要な API の知見 --- 上記のメタ手続き群から使用可能な、言語のメタ操作を行うための API 群についてリストアップした。API の主要なものを次に示す。

(型に対する操作)
・型に対して、既存の単独メソッドシグニチャを追加する
・型に対して、既存の型のメソッドシグニチャ群全部を追加する
・型に対して、他の型をその親クラス(スーパタイプ)として追加する
・ある型が持つメソッド群のシグニチャを順次取得する
(抽象単位に対する操作)
・単位に対して、既存の単独メソッド本体を追加する
・単位に対して、別の単位のメソッド本体群全部を追加する
・ある単位が持つメソッド群の本体を順次取得する
(メソッドに対する操作)
・メソッド本体の末尾に、他のメソッドの本体を付加する
・メソッド本体の冒頭に、他のメソッド本体を挿入する
・メソッドの名称が文字列パターンにマッチするか調べる

(6) 新たな枠組に基づく言語の評価 --- 本研究の成果として設計・実装した o3 言語により、既存の言語機構である「継承」や「アスペクト」と同等の機能が記述できることを動作まで含めた確認した。そのメタ操作の概要は次の通りである。
・継承については、親クラスに相当する単位を参照し、また追加するメソッド群に対する単位を与えて、子クラスに相当する単位を構築するメタ手続きとして実現する。

このメタ手続きは、新たな単位を作り、それを親クラス単位のサブタイプとして定めた上で、親からまずメソッド群一式をコピーしたのち、追加メソッド群をコピーする。これにより、同一名のものがあれば上書きが行われ、一般的な継承と同様の働きとなる。メタ手続きの内容を変更することで、一般的な継承と異なるふるまいにさせることもできる。
・アスペクトについては、加工対象となる単位と、追加コードを持つ単位を与えて、加工対象となる単位の対象メソッド(メソッド名のパターンにより指定 --- 既存の AspectJ などにならった設計)の前や後に追加コードを挿入するようなメタ手続きとして定義できる。

(7) 既存の言語メカニズム以外の機構の探究 --- 既存の言語メカニズムとして知られている継承やアスペクト以外の機構について検討した結果、複数のオブジェクトによるデータの順次処理を行う「パイプライン」木構造に構成されたオブジェクト群による(並列性を含んだ)処理を行う「データを出し入れできる木構造」などがあり得るとの感触を得た。

５．主な発表論文等

〔学会発表〕(計 1 件)

①久野 靖, 挿抜可能メソッドに基づく統一的な抽象単位複合機構,情報処理学会論文誌プログラミング (PRO),8(1),13-13 (2015-06-02), 1882-7802.東京大学(東京都文京区), 情報処理学会プログラミング研究会.

〔リサーチレポート〕(計 2 件)

①久野 靖, プログラミング言語 o3 の言語設計と実装, GSSM ReserchReport No. 2016-01, Graduate School of Systems Management, Univ. of Tsukuba, 2016. http://hdl.handle.net/2241/00135271

② Yasushi Kuno, Unified Composition Mechanism for Abstraction Units Based on Pluggable Methods, GSSM Research Report No. 2015-01, Graduate School of Systems Management, Univ. of Tsukuba, 2015. http://hdl.handle.net/2241/00123048

〔その他〕
ホームページ等
http://www2.gssm.otsuka.tsukuba.ac.jp/staff/kuno/o3lang/

６．研究組織

(1)研究代表者

久野 靖 (KUNO, Yasushi)
筑波大学 ビジネスサイエンス系 教授
研究者番号: 00170019