

## SIMPOSおよびSmalltalkの名前づけ

青柳 龍也

(東大工学部)

我々は、個人向きオブジェクト指向計算機システムORAGA の、名前管理を行うモジュールNameMasterを開発している。NameMaster は、既存のライブラリ中の名前づけをもとに、新しい名前の記号性を検査する。NameMasterの開発には、実際のシステムの名前づけを調べる必要がある。そこで、ICOTの開発した推論マシンPSI 上のOS、SIMPOSと、Xerox 社のSmalltalk システムについて、クラス名、メソッド名のつけ方を調べた。本稿では、名前を構成する語彙、名前の長さ、英語の品詞に基づく名前の構成等について報告する。調査の結果、名前を構成する語彙はかなり小さく、名前を構成する規則の種類も少ないことがわかった。

"Name and Naming in Object Oriented Programming System" (in Japanese)

Tatsuya Aoyagi

Department of Electronic Engineering, The University of Tokyo, 7-3-1 Bunkyo-ku, Tokyo, 153, Japan

Name and naming are important to keep high readability of source programs. The NameMaster, which is one module in the object-oriented programming system ORAGA, helps user to keep names in programs consistent. The NameMaster assumes that a large library has already existed and names in it are consistent. Consistency of names is checked with the system name database.

To develop the NameMaster, we investigated actual name and naming in the large library of SIMPOS and Smalltalk system. SIMPOS is an OS on the personal inference machine PSI developed by ICOT. Smalltalk is an object-oriented language/system by Xerox. This paper reports the vocabulary, length of the names, and naming rule in the actual systems.

## 1. はじめに

我々は、個人向きオブジェクト指向計算機システムORAG AI[1]の、名前管理を行うモジュールNameMaster[2]を開発している。NameMasterは、名前に関するデータベースを保持し、名前に関する種々のサービスを提供する。最も重要なサービスは、新しくシステムに導入される名前の記号性の保持である。

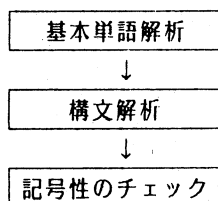
NameMasterは、すでに記号性を保持した名前づけがなされている、かなり大きなライブラリの存在を仮定している。新しい名前の記号性のチェックは、そのようなライブラリ中の名前に基づいて行われる。

このため、NameMasterの開発には、すでに存在するかなり大きなライブラリ中の名前づけを調べる必要がある。また、調査対象は、名前づけに十分な注意が払われていることが望ましい。今回は、ICOTの開発した推論マシンPSI上のOS、SIMPOS[3]と、Xerox社のSmalltalkシステム[4]について、クラス名、メソッド名(メッセージセレクタ)のつけ方を調査した。なお、SIMPOSはバージョン2.52、Smalltalkはバージョン2である。

以下、次節で記号性をチェックするNameMasterの構成について簡単に述べ、その後、調査の結果を報告する。

## 2. 記号性をチェックするNameMaster

記号性をチェックするNameMasterには、



という三つのフェーズがある。これは、一般のコンパイラが字句解析、構文解析、中間コード生成(意味づけ)の三つのフェーズを持つことに対比させることができる。

基本単語解析は、基本単語辞書を用いて名前を構成する基本単語を解析する。名前はいくつかの単語から構成される。NameMasterでは、無意味な綴りは許されない。

ここで、同意語、省略形の処理も行なわれる。同意語、省略形は、基本単語辞書を用いて、基本単語に変換される。このため、基本単語辞書には、同意語や省略形に関する記載も含まれている。

構文解析は、名前の構成規則に基づき名前の構文を解析する。名前の構成規則は、名前に許される文法である。コンパイラの場合には、生成される中間コード(意味)が構文によって決まるように、NameMasterでは使われた構成規

則により名前の持つべき記号性が決まる。

基本単語解析では、どのような基本単語辞書を用意するかが問題となる。構文解析では、どのような構成規則を用いるかが問題となる。今回の調査は、この二点の参考とするために行った。

## 3. 調査の対象

調査はSIMPOS及びSmalltalkシステムのクラス名、メソッド名について行った。

### 3.1 SIMPOS

SIMPOSのシステムクラスは1450個存在する。SIMPOSはESPで記述されている。クラス名は、クラスごとにつづつ存在するので、やはり1450個存在する。これが調査の対象である。

SIMPOSのクラス名は、ESPの文法上は、英字で始まる英数字の並びならば何でもよい。しかし、実際には無意味な綴りというのは少ない。英単語を省略しないで使い、単語と単語の境界は「\_」で示すことが一般的である。

SIMPOS(ESP)のメソッド(述語)には、主処理述語、前デーモン述語、後デーモン述語、ローカル述語等がある。デーモン述語は使われる回数が少なく、ローカル述語の名前づけは重要ではないので、今回の調査は、主処理述語に限定した。

主処理述語のメソッド名は、「:」で始まる英数字の並びであるが、「:」を除いた文字列をメソッド名とした。メソッド名の後には括弧の中にカンマで区切って引数が続くが、引数の個数も含めて、引数に関することは考慮しない。例えば、

```
:create(#window,W)
```

というメッセージでは、:createがメソッド名、#windowとWが引数である。この場合、createという名前だけを調査の対象とする。

ESPには、インヘリタンスの機能が導入されている。このため、一つのメソッドがいくつかのクラス、インスタンスで共有されることがある。今回の調査では、インヘリタンスによるメソッドの共有は考慮せず、実際にメソッドが定義されている場合だけを対象とした。

このようにしてメソッドを数えると、SIMPOSには15064個のメソッドが存在する。この中には、同じメソッド名を持つメソッドも多数存在し、メソッド名の種類は、4412種類である。

また、ESPのすべてのクラスには、new、class\_name、super、class\_objectの4つのメソッドが暗黙の内に定義されている。ソースプログラムにはこれらのメソッドの

記述は存在しない。そこで、これらのメソッドをそれぞれ一回しか数えないことにすると、SIMPOSには9268個のメソッドが存在する。メソッド名の種類は増減しない。

### 3.2 Smalltalk

Smalltalk のクラスは444個存在する。そのうち、半分はメタクラスだが、メタクラスは特に名前を持たないので(クラス名に'class'を付けてメタクラスを表す)、クラス名は222個存在する。

Smalltalk のクラス名は、大文字で始まる英数字の並びであり(表8)、SIMPOS同様、英単語をほとんど省略せず用いる。単語と単語の境界は、大文字を使って示す。

Smalltalk のメソッド名(メッセージセレクタ)は、小文字から始まる英数字の並びで、各引数の前にも文字列と':'が付く。例えば、

```
at: 1 put: 'one'
```

というメッセージで、1と'one'が引数であり、at:put:がメソッド名になる。引数のないメッセージは、例えば、

```
doIt
```

のように、':'を付けない。このときのメソッド名は'doIt'である。

Smalltalk のメソッドは、クラス、メタクラスのメソッドを合わせて、4425個存在する。インスタンスの扱いはSIMPOSと同様である。また、メソッド名の種類は、2769種類である。

表1に、調査対象の大きさをまとめる。

表1 調査対象

	SIMPOS	Smalltalk
クラス数	1450	444
クラス名数	1450	222
メソッド数	15064(9268)	4425
メソッド名数	4412	2769

## 4. 基本単語辞書用の調査

### 4.1 語彙

語彙の調査のためには、語彙の要素となる単語の認定法を決める必要がある。

前節で述べたように、SIMPOSおよびSmalltalkでは、単語は省略しないで使うことが一般的であり、単語と単語の境界も明示されている。SIMPOSでは`\_`で区切られた文字列を、Smalltalkでは大文字で区切られたものを単語とした。

また、単語と見出し語の区別も必要である。異なるクラス名中で使われている同じ文字列の単語は、単語としては別のものだが、見出し語としては、同一のものであるとする。

表2に語彙の大きさを示す。英単語かどうかの判断にはUNIXのspellを用いた。表2のメソッド名の単語数は、調査対象のすべてのメソッドのメソッド名中に含まれる単語の数である。すなわち、同じメソッド名を持つメソッドがいくつかのクラスで定義されているときには、同じ見出し語をクラスの数だけ数えている。また、SIMPOSの括弧中のデータは、すべてのクラスに暗黙の内に定義されるメソッドを、一つとして数えたときの単語数である。(すなわち、28200-1450\*6+6である)

表2 語彙の大きさ

クラス名	SIMPOS	Smalltalk
単語	3948	444
見出し語	697	183
英単語	575	179
非英単語	122	4
共通見出し語	96	

メソッド名	SIMPOS	Smalltalk
単語	28200(19506)	10278
見出し語	1261	1081
英単語	1025	1001
非英単語	236	80
共通見出し語		483

SIMPOS、Smalltalkのクラス名とメソッド名のそれぞれに関して、使用頻度の多い見出し語を表3に示す。頻度は、クラス名やメソッド名でその単語が使われた回数である。また、パーセントは、単語数(表2)に対する割合である。SIMPOSのメソッド名に関しては、暗黙の内に定義されているメソッドは一つとして数えている。従って、パーセントは19506語に対する割合である。

SIMPOSのクラス名では、asやwithの使用頻度が高い。これは、ESPのマルチプルインヘリタンスの機能を利用したプログラミングスタイルによるものである。ESPでは、部品クラスをいくつか継承して、完成品クラスを作る。このときの部品クラスには、asやwithが先頭に付いたクラス名を付けることが多い。

一方、Smalltalkのクラス名では、Model-View-Controllerという枠組でアプリケーションを開発するため、controllerやviewという単語を使ったクラス名が多くなる。

SIMPOSのメソッド名では、動詞の使用頻度が高いのに対して、Smalltalk では、前置詞の使用頻度が高いのが目立つ。これは、Smalltalk では引数の前に文字列を書くが、SIMPOSではカンマで区切るだけであるという、文法上の違いを反映している。Smalltalk では、引数の前に前置詞を使うことが多い。

SIMPOSとSmalltalk で、windowとview、manipulator とcontrollerのように、類似の概念を別の単語で表現していることも観察される。

図1に、メソッド名中の単語の使用頻度の順位とその順位までの単語の頻度の和のグラフを示す。頻度の和は単語数(表2)に対する割合で表している。ここでも、SIMPOSの暗黙の内に定義されるメソッドは一回として数えている。

SIMPOSとSmalltalk のグラフは、ほぼ完全に一致する。どちらも、メソッド名中の全単語の90%は高々500語程度、全単語の半分は高々50語程度の見出し語で占められている。

表4には、SIMPOSとSmalltalk に共通して使われている見出し語を示す。

クラス名の場合のように語彙が比較的小さいときは、共通する見出し語の割合は40%程度であるが、語彙がある

程度大きいメソッド名の場合には、共通する見出し語の割合は75%を越えている。

図1のグラフや共通する見出し語の割合から考えて、見出し語の数がそれほど多くなることはないと予想できる。実際、SIMPOSの全ソースコード(約75万単語)でも、見出し語は3500語程度、そのうち、英単語は2000語程度である。同意語の処理をすれば、その数はもっと少なくなるだろう。

今回集めた見出し語をもとに基本単語辞書を構築することが可能である。

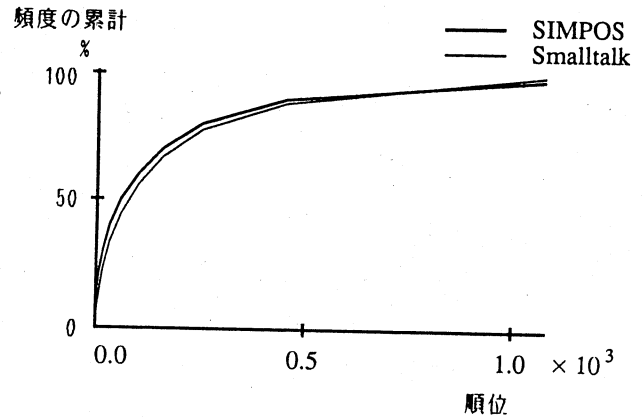


図1 見出し語の順位と頻度の累計

表3 使用頻度の多い見出し語

クラス名	SIMPOS		Smalltalk		メソッド名	SIMPOS		Smalltalk	
	見出し語	頻度 %	見出し語	頻度 %		見出し語	頻度 %	見出し語	頻度 %
as	177	4.5	controller	26 5.9	get	1455	7.5	on	298 2.9
with	146	3.7	view	22 5.0	set	1085	5.6	at	215 2.1
window	143	3.6	change	16 3.6	create	488	2.5	from	143 1.4
file	139	3.5	list	12 2.7	do	255	1.3	file	137 1.3
tap	77	2.0	node	11 2.5	file	213	1.1	class	136 1.3
menu	69	1.7	text	11 2.5	read	207	1.1	to	133 1.3
editor	65	1.6	class	9 2.0	to	199	1.0	display	126 1.2
buffer	49	1.2	file	9 2.0	size	192	1.0	is	125 1.2
error	47	1.2	form	9 2.0	write	190	1.0	print	111 1.1
manipulator	47	1.2	method	9 2.0	string	188	1.0	with	107 1.0
message	45	1.2	stream	8 1.8	add	180	0.9	initialize	104 1.0
handler	42	1.1	display	7 1.6	command	157	0.8	for	103 1.0
input	42	1.1	collection	6 1.4	remove	156	0.8	new	103 1.0
library	39	1.0	dictionary	6 1.4	code	146	0.7	add	90 0.9
command	35	0.9	in	6 1.4	delete	146	0.7	if	89 0.9
mail	35	0.9	scanner	6 1.4	type	145	0.7	name	87 0.9
character	30	0.8	switch	6 1.4	window	145	0.7	do	83 0.8
list	30	0.8	code	5 1.1	initiate	143	0.7	view	83 0.8
font	28	0.7	menu	5 1.1	status	143	0.7	all	78 0.8
table	28	0.7	string	5 1.1	message	142	0.7	remove	78 0.8

表4 共通の見出し語

クラス名 見出し語	SIMPOS		Smalltalk		平均%	メソッド名 見出し語	SIMPOS		Smalltalk		平均%
	頻度	%	頻度	%			頻度	%	頻度	%	
file	139	3.5	9	2.0	2.8	get	1455	7.5	17	0.2	3.8
change	1	0.03	16	3.6	1.8	set	1085	5.6	69	0.7	3.1
list	30	0.8	12	2.7	1.7	on	40	0.2	298	2.9	1.6
menu	69	1.8	5	1.1	1.4	at	106	0.5	215	2.1	1.3
class	27	0.7	9	2.0	1.4	create	488	2.5	2	0.0	1.3
text	9	0.2	11	2.5	1.4	file	213	1.1	137	1.3	1.2
node	4	0.1	11	2.5	1.3	to	199	1.0	133	1.3	1.2
editor	65	1.7	3	0.7	1.2	do	255	1.3	83	0.8	1.1
method	10	0.3	9	2.0	1.1	is	128	0.7	125	1.2	0.9
display	17	0.4	7	1.6	1.0	add	180	0.9	90	0.9	0.9
stream	6	0.2	8	1.8	1.0	name	177	0.9	87	0.9	0.9
message	45	1.2	3	0.7	0.9	from	68	0.4	143	1.4	0.9
character	30	0.8	4	0.9	0.8	with	136	0.7	107	1.0	0.9
input	42	1.1	2	0.5	0.8	class	77	0.4	136	1.3	0.9
string	13	0.3	5	1.1	0.7	size	192	1.0	76	0.7	0.9
error	47	1.2	1	0.2	0.7	remove	156	0.8	78	0.8	0.8
in	3	0.1	6	1.4	0.7	display	52	0.2	126	1.2	0.8
code	10	0.3	5	1.1	0.7	initialize	94	0.5	104	1.0	0.8
system	8	0.2	5	1.1	0.7	string	188	1.0	53	0.5	0.7
array	13	0.3	4	0.9	0.6	read	207	1.1	38	0.4	0.7
:	:	:	:	:	:	:	:	:	:	:	:
合計	1094	27.9	247	55.6	41.7	合計	14759	75.7	7868	76.6	76.1

4.2 名前の長さ

名前の長さに関する統計量を表5に示す。単語は省略しないという原則のため、名前の長さはかなり長い。このため、NameMasterは省略形をサポートする必要がある。クラス名で、SIMPOSの方が長いのは、`\_`で単語を区切ることに原因がある。メソッド名で逆にSmalltalkの方が長いのは、各引数の前に文字列を付ける必要があるためだろう。

5. 名前の構成規則用の調査

5.1 名前を構成する単語数

名前を構成する単語数の平均を表6に示す。表6の値は表1と表2にデータから計算される。

一つのクラス名を構成する単語数の分布を図2に示す。

4語以上の名前は少ないことがわかる。名前の構成規則は3語程度に制限できそうである。

表5 名前の長さ

クラス名	SIMPOS	Smalltalk
平均	16.6	12.2
標準偏差	6.6	5.6
最大	47	31
最少	2	3 (文字数)

メソッド名

	SIMPOS	Smalltalk
平均	10.0	12.5
標準偏差	5.4	9.0
最大	39	141
最少	1	1 (文字数)

表6 名前を構成する単語数の平均

	SIMPOS	Smalltalk
クラス名	2.7	2.0
メソッド名	1.8(2.1)	2.3 (単語/名前)

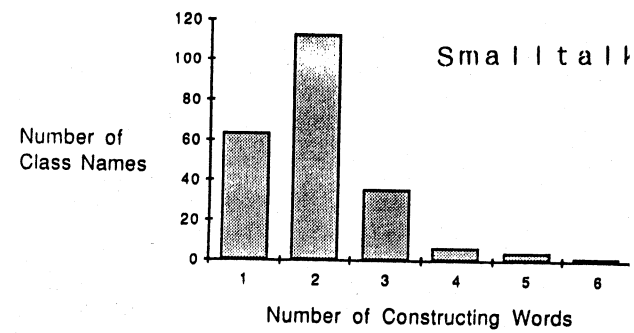
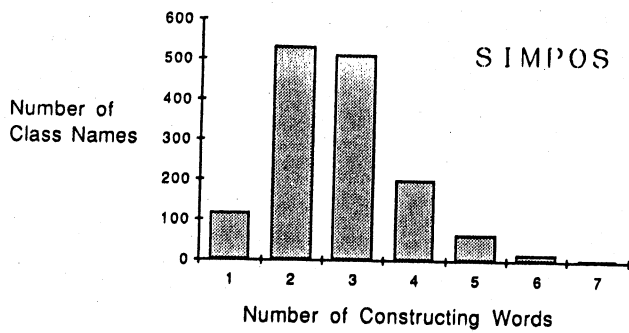


図2 クラス名を構成する単語数

5.2 英語の品詞に基づく名前の構成

Smalltalk の全クラス名に関して、どのような品詞から構成されているかを調べた。結果を表7に示す。表8にクラス名の例を示す。

Smalltalk のクラス名はすべて名詞句である。そこで、各単語の品詞に基づく構成を調べた。

Smalltalk のクラス名のほとんどは、少数の構成規則で作られていることがわかる。名詞のみできている名前が75%を占める。4つ以上の名詞から構成されているクラ

ス名は存在しない。また、最後の単語が名詞以外であるようなクラス名も存在しない。

表9にSIMPOS及びSmalltalk のメソッド名の構成を示す表10にメソッド名の例を示す。

メソッド名の構成は、出現頻度の高いメソッド名から順に、頻度の合計が全体の1/4程度になるようにメソッド名を選んで調査した。先のクラス名の場合は、すべて名詞句だったので、各単語ごとの品詞を調べたが、ここでは、名前全体でどのような動きをする句になっているかを調べた。

SIMPOS、Smalltalk とともに、動詞句の割合が最も高い。特に、SIMPOSでは、調べた範囲内では、90%近くのメソッド名が動詞句になっている。

メソッド名に動詞句が多いのは、オブジェクトにメッセージを送り、オブジェクトを動作させるといふ、オブジェクト指向のモデルからの当然の帰結である。メソッド名はオブジェクトへの命令であり、動詞の原形が使われる。

一方、オブジェクトには、データとデータに対するアク

表7 Smalltalk のクラス名の構成

	頻度	%
名詞+名詞	84	37.8
名詞	62	27.9
名詞+名詞+名詞	21	9.5
形容詞+名詞	9	4.1
動詞原形+名詞	8	3.6
過去分詞+名詞	6	2.7
形容詞+名詞+名詞	4	1.8
その他	28	12.6

表8 Smalltalk のクラス名の例

名詞+名詞	ActionMenu, AssignmentNode
名詞	Arc, Array
名詞+名詞+名詞	CharacterBlockScanner, ClassCategoryReader
形容詞+名詞	BanaryChoice, ExternalStream
動詞原形+名詞	ParseNode, ReadStream
過去分詞+名詞	LinkedList, SharedQueue
形容詞+名詞+名詞	BinaryChoiceController, StandardSystemView

表9 メソッド名の構成

SIMPOSのメソッド名(28.8%)の構成

	頻度	%
動詞句	2349	88.1
名詞句	233	8.7
形容詞副詞句	71	2.7
その他	12	0.5
合計	2665	

Smalltalk のメソッド名(26.7%)の構成

	頻度	%
動詞句	577	48.9
名詞句	292	24.8
形容詞副詞句	153	13.0
be動詞句	43	3.6
その他	115	9.8
合計	1180	

## SIMPOS

## 動詞句

create, initiate, cut\_off, get\_status

## 名詞句

class\_name, goal, size, command

## 形容詞副詞句

new, super, in, empty

## Smalltalk

## 動詞句

initialize, printOn:, release, add:

## 名詞句

size, defaultControllerClass, text, example

## 形容詞副詞句

new, at:, on:, asString

## be動詞句

isEmpty, isControlActive, isLiteral

## &lt; 参考文献 &gt;

- [1] 神田, "並列オブジェクト指向計算機システムORAGA"、東京大学情報工学博士論文、1986
- [2] 青柳他, "NameMasterの実現に向けて"、情報処理学会第33回全国大会50-4、1986
- [3] 新世代コンピュータ技術開発機構, "SIMPOS総合説明書(第一版)"、1986
- [4] A. Goldberg and D. Robson, "SmalltalkK-80 The Language and its Implementation", 1983

セスメソッドを一つにまとめた情報隠蔽モジュールとしての動きもある。そのようなアクセスメソッドとしてのメソッド名には動詞句でないものも考えられる。例えば、Smalltalkの名詞句、形容詞副詞句のメソッド名、size、at:、next等がその例である。

Smalltalkにはその他にも、データタイプ(クラス)の変換を表す、asで始まるメソッド名や、真偽値を返すメソッドを表す、isで始まるメソッド名などの規則化されたメソッド名が存在する。

## 6. おわりに

NameMasterの基本単語辞書と名前の構成規則を作るために、実際のオブジェクト指向システムの名前を調べた。

その結果、SIMPOSやSmalltalkシステムで名前づけに使われる見出し語の数はかなり少ないことがわかった。基本単語辞書には2000語程度の辞書を用意すれば十分実用的である。また、今回収集した見出し語は、基本単語辞書の最初のバージョンとして利用することができる。

名前の長さはかなり長いことがわかった。長い名前は、表示の際のインデントを乱し、入力も困難である。このため、NameMasterでは省略形のサポートを考えている。

名前を構成する規則の種類は限られている。明確な名前の構成規則を定め、それをもとに記号性を検査することが可能である。今回の結果は名前の構成規則を設計する上で参考になる資料である。

今後は、この結果をもとに、NameMasterを設計していく