

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-160214
(P2012-160214A)

(43) 公開日 平成24年8月23日(2012.8.23)

(51) Int.Cl.		F I			テーマコード(参考)
G06F 9/45 (2006.01)		G06F 9/44	320F		5B081
G06F 9/44 (2006.01)		G06F 9/06	620A		5B376

審査請求 有 請求項の数 8 O L 外国語出願 (全 17 頁)

(21) 出願番号	特願2012-121089 (P2012-121089)	(71) 出願人	509123208 アビニシオ テクノロジー エルエルシー アメリカ合衆国 02421 マサチュー セッツ州 レキシントン スプリング ス トリート 201
(22) 出願日	平成24年5月28日(2012.5.28)	(74) 代理人	100079108 弁理士 稲葉 良幸
(62) 分割の表示	特願2008-519475 (P2008-519475) の分割	(74) 代理人	100109346 弁理士 大貫 敏史
原出願日	平成18年6月27日(2006.6.27)	(72) 発明者	スタンフィ爾 クレイグ ダブリュ アメリカ合衆国 01773 マサチュー セッツ州 リンカーン ハックルベリー ヒル ロード 43
(31) 優先権主張番号	11/169,247	Fターム(参考)	5B081 AA10 5B376 BC74 FA10
(32) 優先日	平成17年6月27日(2005.6.27)		
(33) 優先権主張国	米国 (US)		

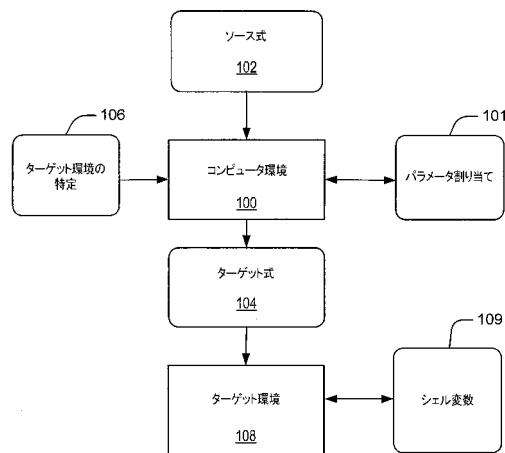
(54) 【発明の名称】 コンピュータ環境における式の変換

(57) 【要約】

【課題】 コンピュータシステムにおけるバインド環境でパラメータの値を格納する際のオーバーヘッドをなくすこと。

【解決手段】 本発明は、式を評価するための方法ならびに対応するソフトウェアおよびシステムに関する。本発明の方法は：パラメータ定義を格納する一つ以上のデータ構造(136)を受け取るステップ；前記式で参照される一つ以上のパラメータを特定するステップ；格納された定義に基づいて前記特定したパラメータそれぞれの値を決定するステップ；および、バインド環境(300)内に前記決定したパラメータ値を格納するステップ；を含む。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

式を変換するための方法であって：

第 1 シンタックスの第 1 式 (1 0 2) を受け取るステップであって、前記式は、前記第 1 シンタックスを用いて文字列の値を表す第 1 文字シーケンスを含み；

前記式を表すための第 2 シンタックスを特定するステップ；および、

前記第 2 シンタックスで前記第 1 式を表す第 2 式 (1 0 4) を生成するステップであって、前記第 2 文字シーケンスが前記第 2 シンタックスを用いた前記文字列の値を表すように、前記第 1 文字シーケンスを用いて第 2 文字シーケンスを生成するステップを含む；を

含む、

方法。

10

【請求項 2】

前記第 1 式は、複数の実行可能ステートメントを含む、請求項 1 の方法。

【請求項 3】

前記第 1 式は、コマンドライン入力を含む、請求項 1 の方法。

【請求項 4】

前記第 1 文字シーケンスは、参照される値へバインドされるパラメータを含む、請求項 1 の方法。

【請求項 5】

前記第 1 文字シーケンスを用いて第 2 文字シーケンスを生成するステップは：

前記パラメータを前記参照される値へバインドするステップ；および、

前記第 2 シンタックスの引用符シンタックスに基づいて前記参照値を含めるように前記第 2 文字シーケンスを生成するステップ；を含む、

請求項 4 の方法。

20

【請求項 6】

前記第 1 文字シーケンスは、前記パラメータを囲む区切り文字を含む、請求項 4 の方法。

【請求項 7】

前記区切り文字は、引用符シンボルがそのあとに続くドルシンボルを含む所定のプレフィックスを含む、請求項 6 の方法。

30

【請求項 8】

前記区切り文字は、引用符シンボルを含む所定のサフィックスを含む、請求項 6 の方法。

【請求項 9】

前記第 1 文字シーケンスは、前記文字列の値の表現を囲む区切り文字を含む、請求項 1 の方法。

【請求項 10】

前記文字列の値は、パラメータ参照を示す、前記第 1 シンタックスで用いられる文字を含む、請求項 9 の方法。

【請求項 11】

前記第 1 文字シーケンスを用いて第 2 文字シーケンスを生成するステップは、前記文字列の値内のパラメータのバインドを無効にするステップを含む、請求項 10 の方法。

40

【請求項 12】

前記第 2 シンタックスは、計算機言語と互換性をもつよう選定される、請求項 1 の方法。

【請求項 13】

前記計算機言語は、k s h、J a v a (登録商標)、D M L、C、および S Q L から成るグループのメンバーである、請求項 12 の方法。

【請求項 14】

前記第 2 シンタックスは、文字シーケンスを解釈するための規則を含む、

50

請求項 1 の方法。

【請求項 15】

前記第 2 シンタックスを特定するステップは、前記第 1 文字シーケンスに先行するトークンに基づいてシンタックスを特定するステップを含む、

請求項 1 の方法。

【請求項 16】

式を変換するためのコンピュータ可読媒体上に格納されるソフトウェアであって：

第 1 シンタックスの第 1 式 (1 0 2) を受け取るステップであって、前記式は、前記第 1 シンタックスを用いて文字列の値を表す第 1 文字シーケンスを含み；

前記式を表すための第 2 シンタックスを特定するステップ；および、

前記第 2 シンタックスで前記第 1 式を表す第 2 式 (1 0 4) を生成するステップであって、前記第 2 文字シーケンスが前記第 2 シンタックスを用いた前記文字列の値を表すように、前記第 1 文字シーケンスを用いて第 2 文字シーケンスを生成するステップを含む；

をコンピュータシステムに実行させるための命令を含む、ソフトウェア。

【請求項 17】

式を変換するためのシステムであって：

第 1 シンタックスの第 1 式 (1 0 2) を受け取るための手段であって、前記式は、前記第 1 シンタックスを用いて文字列の値を表す第 1 文字シーケンスを含み；

前記式を表すための第 2 シンタックスを特定するための手段；および、

前記第 2 シンタックスで前記第 1 式を表す第 2 式 (1 0 4) を生成するための手段であって、前記第 2 文字シーケンスが前記第 2 シンタックスを用いた前記文字列の値を表すように、前記第 1 文字シーケンスを用いて第 2 文字シーケンスを生成するための手段を含む；を含む、

システム。

【請求項 18】

式を評価するための方法であって：

パラメータ定義を格納する一つ以上のデータ構造 (1 3 6) を受け取るステップ；

前記式で参照される一つ以上のパラメータを特定するステップ；

格納された定義に基づいて前記特定したパラメータそれぞれの値を決定するステップ；

および、

バインド環境 (3 0 0) 内に前記決定したパラメータ値を格納するステップ；を含む、方法。

【請求項 19】

前記バインド環境は、前記特定したパラメータの値、および前記特定したパラメータを評価するのに必要なパラメータの値を格納し、前記バインド環境は、前記一つ以上のデータ構造内にその定義を格納する前記パラメータ全ての値を格納するとは限らない、請求項 18 の方法。

【請求項 20】

参照パラメータを前記バインド環境からの値へバインドさせた状態で、前記式を実行するステップを更に含む、

請求項 18 の方法。

【請求項 21】

前記式はプログラムまたはスクリプトの一部である、請求項 18 の方法。

【請求項 22】

前記式はコマンドライン入力を含む、請求項 18 の方法。

【請求項 23】

式を変換するためのコンピュータ可読媒体上に格納されるソフトウェアであって：

パラメータの定義を格納する一つ以上のデータ構造 (1 3 6) を受け取るステップ；

前記式で参照される一つ以上のパラメータを特定するステップ；

格納される定義に基づいて前記特定したパラメータそれぞれの値を決定するステップ；

50

10

20

30

40

および、

バインド環境 (3 0 0) 内に前記決定したパラメータ値を格納するステップ ; をコンピュータシステムに実行させるための命令を含む、
ソフトウェア。

【請求項 2 4】

式を評価するためのシステムであって :

パラメータ定義を格納する一つ以上のデータ構造 (1 3 6) を受け取るための手段 ;

前記式で参照される一つ以上のパラメータを特定するための手段 ;

格納された定義に基づいて前記特定したパラメータそれぞれの値を決定するための手段 ; および、

バインド環境 (3 0 0) 内に前記決定したパラメータ値を格納するための手段 ; を含むシステム。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明は、コンピュータ環境における式 (エクスプレッション) の変換に関する。

【背景技術】

【0 0 0 2】

発明の背景

シェルスクリプト言語 (例えば、k s h) では、シェルスクリプト内のパラメータは、プレフィックス (例えば、\$) を用いて参照される。k s h では、動的環境が、以前に定義されたパラメータの値を維持する。シェルスクリプトを実行するか、またはコマンドラインの式を評価する場合、参照されるパラメータは、動的環境内のパラメータ値にしたがってバインドされる。場合によっては、スクリプト言語でスクリプトを創成する場合、スクリプト言語に適した引用符シンタックスを用いて、リテラル値を維持するようにし、スクリプトを実行する時に引用符付きの式内のパラメータのバインドを無効にすることが有用である。マクロ展開のような、他のメカニズムを用いて、所定の値にパラメータをバインドすることもできる。

【発明の概要】

【課題を解決するための手段】

【0 0 0 3】

概要

一般的な態様では、本発明は式を変換するための方法、ならびに対応するソフトウェアおよびシステムを特徴とする。本方法は、第 1 シンタックスの第 1 式を受け取るステップであって、その式は、第 1 シンタックスを用いて文字列の値を表す第 1 文字シーケンスを含む ; その式を表すための第 2 シンタックスを特定するステップ ; および第 2 シンタックスで第 1 式を表す第 2 式を生成するステップであって、第 2 文字シーケンスが第 2 シンタックスを用いた文字列の値を表すように、第 1 文字シーケンスを用いて第 2 文字シーケンスを生成するステップを含む ; を含む方法である。

【0 0 0 4】

本態様には、以下の特徴の内の一つ以上を含めることができる :

第 1 式は複数の実行可能ステートメントを含む。

第 1 式はコマンドライン入力を含む。

第 1 文字シーケンスは参照値へバインドされるパラメータを含む。

第 1 文字シーケンスを用いて第 2 文字シーケンスを生成するステップは、パラメータを参照値へバインドするステップ ; および第 2 シンタックスの引用符シンタックスに基づいて参照値を含めるよう第 2 文字シーケンスを生成するステップ ; を含む。

第 1 文字シーケンスはパラメータを囲む区切り文字を含む。

区切り文字は、引用符シンボルがそのあとに続くドルシンボルを含む所定のプレフィックスを含む。

10

20

30

40

50

区切り文字は引用符シンボルを含む所定のサフィックスを含む。

第1文字シーケンスは文字列の値の表現を囲む区切り文字を含む。

文字列の値は、パラメータ参照を示す、第1シンタックスで用いられる文字を含む。

第1文字シーケンスを用いて第2文字シーケンスを生成するステップは、文字列の値内のパラメータのバインドを無効にするステップを含む。

第2シンタックスは、ksh、Java（登録商標）、DML、C、またはSQL等の計算機言語と互換性をもつよう選定される。

第2シンタックスは、文字シーケンスを解釈するための規則を含む。

第2シンタックスを特定するステップは、第1文字シーケンスに先行するトークンに基づいてシンタックスを特定するステップを含む。

10

【0005】

一般的な態様では、本発明は式を評価するための方法、ならびに対応するソフトウェアおよびシステムを特徴とする。本方法は、パラメータ定義を格納する一つ以上のデータ構造を受け取るステップ；式で参照される一つ以上のパラメータを特定するステップ；格納された定義に基づいて特定したパラメータそれぞれの値を決定するステップ；およびバインド環境に決定したパラメータ値を格納するステップを含む。

【0006】

本態様には、以下の特徴の内の一つ以上を含めることができる：

バインド環境は、特定したパラメータの値、および特定したパラメータを評価するのに必要なパラメータの値を格納するが、一つ以上のデータ構造内にその定義を格納するパラメータ全ての値を格納するとは限らない。

20

本方法は、参照パラメータをバインド環境からの値へバインドさせた状態で、式を実行するステップを更に含む。

式はプログラムまたはスクリプトの一部である。

式はコマンドライン入力を含む。

【発明の効果】

【0007】

本発明の態様には、以下の利点の内の一つ以上を含めることができる：

データ処理環境に、データ変換またはレコードフォーマットの定義（例えば、データ操作言語で記述された）等のメタデータを含めることができる。ランタイム時に取得されるリテラル値で定義できる動的メタデータを有することが有用である。特定したシンタックスに基づいてターゲットの式を変換すると、リテラル値を取得できるだけでなく、特定したシンタックスに適した方法でリテラル値を引用できるようになる。更に、特定したシンタックスに対する引用規則を順守することにより、ターゲットの式の評価に際して、幾つかのリテラル値を維持することができる。

30

【0008】

コンピュータシステムは、バインド環境でパラメータの値を格納できる。式（例えば、シェルスクリプトまたはコマンドラインの式）を評価する場合、参照パラメータは、バインド環境におけるパラメータ値に基づいてバインドされる。バインド環境内のパラメータ値維持にはオーバーヘッドが存在する。定義パラメータ毎の値を（例えば、パラメータおよびそれらの定義のリスト内に）有するバインド環境を維持する必要性を避けるため、コンピュータシステムは式をスキャンして、その式で実際に参照されるパラメータ、および参照されるパラメータを評価するのにその値が必要なパラメータ、に対してだけ値を決定できる。他の特徴および利点は、以下の説明、およびクレームから明らかになる。

40

【図面の簡単な説明】

【0009】

【図1】図1は、コンピュータ環境とターゲット環境との間の相互作用を表すブロック図である。

【図2】図2は、式を変換するためのコンピュータ環境のブロック図である。

【図3】図3は、データ収納庫内のパラメータ割り当てのための格納領域の内容を示すブ

50

ロック図である。

【図 4】図 4 は、ローカルメモリ内のバインド環境の内容を示すブロック図である。

【発明を実施するための形態】

【0010】

説明

1 概要

図 1 を参照すると、コンピュータ環境 100 は、ソースシンタックスを用いて、ソース計算式 102 の仕様（例えば、スクリプト、プログラム、パス名等）をサポートし、次いで、そのソース式 102 を、幾つかの可能性のあるターゲット環境の内の一における評価（例えば、実行、拡張等）に適したターゲットシンタックスを用いるターゲット式 104 へ変換する。コンピュータ環境 100 は、シェル環境またはコンピュータープログラミング言語環境等の、ターゲット式 104 を評価する特定のターゲット環境 108 を特定するためのプロセス 106 を含む。プロセス 106 は、ターゲット式 104 内の情報に基づくか、他のユーザ入力に基づくか、または既定の環境に基づいてターゲット環境を特定することができる。

10

【0011】

この変換の一部として、幾つかのパラメータ参照は、変換の際にパラメータ割り当て 101 に基づいて解決できるように、その一方で、他のパラメータ参照は、変換式を後で評価する際に（例えば、シェル変数 109 に基づいて）解決できるように、選択可能なパラメータ代入を実行する。この変換プロセスは、ターゲットシンタックスの特性、特に、引用符付き文字列およびパラメータ参照に関するこれらの特性に依存する。

20

【0012】

ターゲット式の評価に際して、リテラル値が維持されていることが望ましいことがある。一般に、ターゲットシンタックスそれぞれは、リテラル値を引用するための何らかの種類の規定を含み、演算子、パラメータ代入、または区切り文字に関するシンタックスの解釈等の、シンタックスの他の解釈に基づいてそれが評価されないようになっている。

【0013】

ターゲット式内のリテラル値の幾つかは、ソース式内のシンボル値に基づいて、例えば、パラメータ代入に基づいて、変換している間に取り込むことができる。すなわち、ターゲット式内の所望するリテラル式の値は、ソース式を規定するときに既知でない可能性がある。

30

【0014】

ソース式は、そのソース式を評価する時に引用される値の幾つかを取得することにより所望のターゲット式を生成することができる言語、例えば、パラメータ定義言語（PDL）で、定義することができる。このような PDL は、多様な可能性のあるターゲット言語の内の一の適切なシンタックスで、引用されるリテラルを生成するためのメカニズムを含む。「式」には、例えば、スクリプトもしくはプログラムの一部（スクリプトまたはプログラム全体を含む）、またはコマンドライン入力を含めることができる。

【0015】

図 2 は、PDL の式を評価し、一つ以上のターゲット言語（例えば、シェル言語または他のスクリプト言語）のシンタックスで得られたターゲット式を実行するための例示のコンピュータ環境 100 を示す。コンピュータ環境 100 は、ローカルメモリ 120（例えば、ランダムアクセスメモリ）およびデータ収納庫 130（例えば、一つ以上の不揮発性格納媒体に常駐する）を含むコンピュータシステム 110 を含む。データ収納庫 130 は、スクリプト 132、プログラム 134、ならびに多様なスクリプト 132 および / またはプログラム 134 の何れかに用いることができるパラメータセット内に編成することができるパラメータ割り当て 136（例えば、パラメータ名、パラメータ値、およびオプションでパラメータ属性を含む）等の機能データを格納する。ユーザは、グラフィカルユーザインターフェース（GUI）140 および / またはコマンドラインインターフェース（CLI）150 を介してコンピュータシステムと相互作用することができる。

40

50

【 0 0 1 6 】

パラメータ割り当て 1 3 6 には、多様なターゲットシンタックスの何れかに変換すべきパラメータ値をユーザが定義できる、整合性があるシンタックスを提供するソース P D L の式を用いて、定義されるパラメータを含めることができる。パラメータ定義は、他のパラメータの値を参照することができる。多様な規則の何れかを用いて、どのパラメータが他のどのパラメータを参照できるかを決定することができる（例えば、順序リストでは、パラメータ定義が、以前に定義したパラメータを参照でき；またはセットでは、一セットのパラメータが、別のセットのパラメータを参照できる）。

【 0 0 1 7 】

詳細に後述するソース P D L の式を評価するための手法は、P D L の式を受け取るステップ、P D L の式を表すためのターゲットシンタックスを特定するステップ、およびターゲットシンタックスに基づいて P D L の式の表現を生成するステップを含む。例えば、P D L の式を含むスクリプトには、P D L の式が続く一連のインライントークンを含めることができる。コンピュータシステム 1 1 0 は、続く P D L の式を解釈するのに用いられるターゲット言語を指示するスクリプト内の各トークンを読み取ることにより、ターゲットシンタックスを特定する。従って、スクリプトの異なる部分を解釈するために用いる異なるターゲット言語を指示する異なるトークンをスクリプト内に含めることにより、スクリプト内のターゲットシンタックスは変更し得る。

10

【 0 0 1 8 】

受け取った P D L の式には、文字列の値を表す文字シーケンスを含めることができる。文字シーケンスは、(a) パラメータ参照（例えば、パラメータの値が文字列の値へバインドされるべきであると示すパラメータのプレフィックス）により、文字列の値を表すことができる。代替として、文字シーケンスは、(b) 引用符付きリテラルの式（例えば、文字列の値の表現を囲む区切り文字）により、文字列の値を表すことができる。

20

【 0 0 1 9 】

P D L の式の文字シーケンスが、(a) パラメータ参照により文字列の値を表す場合、コンピュータシステム 1 1 0 は、パラメータ参照の形式を、例えば、参照のシンタックスに基づいて特定する。参照は、「直接代入」パラメータ参照とすることができ、その場合は、コンピュータシステム 1 1 0 はパラメータ参照をパラメータ値へバインドする。代替として、参照を「代入および引用」パラメータ参照とすることができ、その場合は、コンピュータシステム 1 1 0 は、詳細に後述するように（例えば、セクション 2 . 2 . 3 ）、ターゲットシンタックスで引用符付きリテラルを生成するためのメカニズムを用いて、パラメータ参照をパラメータ値へバインドする。

30

【 0 0 2 0 】

P D L の式の文字シーケンスが、(b) 引用符付きリテラルの式により文字列の値を表す場合、コンピュータシステム 1 1 0 は、代入を実行せずにリテラル式の文字列の値を維持する。コンピュータシステム 1 1 0 は、ターゲットシンタックスに基づいて何を引用符付きリテラルの式として解釈すべきかを決定する。通常であればパラメータ参照（例えば、直接代入パラメータ参照）を表す筈の文字シーケンスが、引用符付きリテラルの式内に組み込まれている場合、その文字シーケンスはパラメータ参照として扱われない（例えば、パラメータ値へバインドされない）。

40

【 0 0 2 1 】

コンピュータシステム 1 1 0 が、パラメータ参照を値にバインドする場合、定義されているパラメータ毎の値（例えば、パラメータおよびその定義のリスト内で）により、動的環境を設定する時間を費やす必要を避けるために、コンピュータシステム 1 1 0 は、式をスキャンして、その式の中で実際に参照されるパラメータに対する値だけを決定することができる。パラメータは他のパラメータで定義できるので、式で直接参照されないパラメータの幾つかも、直接参照されるパラメータを評価するプロセスで評価される可能性がある。直接または間接的に参照されるパラメータの値は、ローカルメモリ 1 2 0 内のバインド環境に格納される。コンピュータシステム 1 1 0 が、P D L の式を評価した後、コンピ

50

ユータシステムは、対応するターゲットシンタックスでターゲットの式を生成する。次いで、コンピュータシステム 1 1 0 は、ターゲットの式を実行し、および / またはデータ収納庫 1 3 0 にターゲットの式を格納する。

【 0 0 2 2 】

2 式の変換

ソースシンタックスからターゲットシンタックスへの式の変換は、変換時に既知のパラメータ値を用いるパラメータ代入を伴うことがある。ソースの P D L シンタックスは、幾つかの「\$ 代入」シンタックスを用いてパラメータを値で置換する。

【 0 0 2 3 】

ソースの P D L シンタックスは、「直接代入」パラメータ参照を用いる第 1 種の「\$ 代入」シンタックスを用いる。このシンタックスは、\$<名称>を、\$fooのように使用し、文字列\$fooの代わりにパラメータfooの値を直接代入する。注意すべきは、このような代入は、\$が次のように文字列リテラル内で用いられる場合は、実行されないということである。

```
'the price is $12.00'
```

【 0 0 2 4 】

値1.00を有するパラメータfooによる直接代入の例示は、次のソース式を取ることになると、

```
if($foo==3.14)
```

次のターゲット式が得られる。

```
if(1.00==3.14)
```

この場合、\$文字をプレフィックスとして用いて、パラメータfooへ割り当てられている値を、直接代入パラメータ参照\$fooに代入すべきであることを指示する。

【 0 0 2 5 】

しかし、場合によっては、パラメータ参照に代入される値が、ターゲットシンタックス（例えば、データ操作言語（DML）等のターゲット言語に基づく場合）で正当であるという保証はない。例えば、ソースパラメータバインド環境でfooに値「(」（すなわち、左丸括弧文字を表す長さ1の文字列）が割り当てられている場合、上記のソース式により次のターゲット式が得られる。

```
if((==3.14)
```

これは、ターゲット言語がDMLの場合、ターゲットシンタックスで不当な式となる。

【 0 0 2 6 】

代わりに、ソース式が次式である場合、

```
if("$foo"==3.14)
```

このソース式に同一パラメータを代入すると、次の正当なターゲット式が得られる。

```
if("("==3.14)
```

（コンピュータシステム 1 1 0 が、後述するように、二重引用符が保護区切り文字である処理モードにないと仮定する）。

【 0 0 2 7 】

しかし、fooに値「"」を割り当てる場合（すなわち、二重引用符を表す長さ1の文字列）、上記のソース式のfooにこの値をパラメータ代入すると次のターゲット式が得られる。

```
if(""==3.14)
```

これは、ターゲット言語がDMLの場合、不当なシンタックスである。

【 0 0 2 8 】

ソース P D L シンタックスは、「代入および引用」パラメータ参照を用いる第 2 種の「\$ 代入」シンタックスを用いる。コンピュータシステム 1 1 0 は、パラメータ参照をパラメータ値へバインドし、指定されたターゲットシンタックスで引用符付きリテラルを生成する。この種のパラメータ参照は、所定の区切り文字セットにより示すことができる。P D L では、区切られた「代入および引用」パラメータ参照は、\$"foo"により与えられる（

10

20

30

40

50

または '\$foo' としても等価である)。

【 0 0 2 9 】

この「代入および引用」パラメータ参照を用いると、fooに値「(」を割り当てる場合(すなわち、左丸括弧文字を表す長さ1の文字列)、次のソース式、

```
if("$foo"==3.14)
```

により、次のターゲット式を得る。

```
if("("==3.14)
```

これはターゲット言語がDMLの場合、正当なシンタックスである。

【 0 0 3 0 】

繰り返しになるが、この「代入および引用」パラメータ参照を用いると、fooに値「"」を割り当てる場合(すなわち、二重引用符を表す長さ1の文字列)、次のソース式、

```
if("$foo"==3.14)
```

により、例えば、ターゲット言語がDMLの場合、次のターゲット式を得る。

```
if("\\""==3.14)
```

これは、DMLでは、引用符付きの引用符号にエスケープ文字\が先行する必要があるからである。異なるターゲット言語を用いる場合、ターゲット式は異なるシンタックスを用いることができる。例えば、次のソース式、

```
if("$foo"==3.14)
```

により、ターゲット言語がkshの場合、次のターゲット式を得る。

```
if('"'==3.14)
```

【 0 0 3 1 】

ターゲットシンタックスは、ターゲット言語がkshであることを示す「\$-ksh」や、ターゲット言語がDMLであることを示す「\$-dml」等の、ソース式内に組み込まれたトークンにより示すことができる。トークンによっては、対応するターゲット言語を持たないターゲットシンタックスを示すこともあるが(例えば、\$-リテラル)、それでもセットの規則により定義される良好な定義シンタックスをもっている。コンピュータシステム110がターゲットシンタックスのトークンを認識した後、式の評価は、そのターゲットシンタックスと対応する「処理モード」内に置かれる。

【 0 0 3 2 】

「代入および引用」パラメータ参照メカニズムは、ソースPDLの式が書かれ、引用された後に、ターゲットシンタックスに適した処理モードにより決定される方法でターゲットシンタックス値を取得する方法を提供する。ソースPDLの式を書いている時に既知であるターゲットシンタックスの値は、所定の「保護区切り文字」(「\$」文字を先頭に付けない二重または一重の引用符等)の内側に閉じ込めることにより、ターゲットシンタックスと整合性がとれているリテラル値として維持することができる。すなわち、ソースシンタックスは区切り文字の付いた文字列内への代入を必要としない。

【 0 0 3 3 】

例えば、文字列\$PIが、DMLターゲット言語で(パイ)の値を表す特殊文字列である場合、DML文字シーケンス内に含まれる時は、この文字列を二重引用符で囲む("\$PI"のように)ことにより、例えば、"\\$PI"として直接代入を「エスケープ」する必要なく、変更されないようにすることができる。異なるターゲット言語では、異なる保護区切り文字を用いて、kshターゲット言語の'\$PI'のように文字シーケンスを保護することができる。PDL式を評価する場合、コンピュータシステム110は、保護区切り文字内の「直接代入」パラメータ参照、および「代入および引用」パラメータ参照をともにOFFにすることにより、保護区切り文字を順守する。コンピュータシステム110は、処理モードに基づいて保護区切り文字(それがあある場合)が何であるかを判定する。

【 0 0 3 4 】

2.1 ターゲットシンタックス処理モード

異なるターゲットシンタックス処理モードは、PDLの式を評価するための異なる振る舞いを生み出す。コンピュータシステム110は組み込まれたトークンを読むことにより

10

20

30

40

50

処理モードを認識できるが、しかし、トークンがない場合は、既定値の処理モードを仮定することができる。4つの例示の処理モードの式評価の振る舞いを下記のようにまとめる：

【0035】

ターゲットシンタックス：ksh

トークン：\$-ksh

処理モード振る舞い：コンピュータシステム110は、「代入および引用」に対するkshのターゲット言語と整合性がある引用符付きシンタックスを用いて、kshのターゲット言語が用いる保護区切り文字を認識する。

【0036】

ターゲットシンタックス：DML

トークン：\$-dml

処理モード振る舞い：コンピュータシステム110は、「代入および引用」に対するDMLのターゲット言語と整合性がある引用符付きシンタックスを用いて、DMLのターゲット言語が用いる保護区切り文字を認識する。

【0037】

ターゲットシンタックス：テキスト

トークン：\$-text

処理モード振る舞い：コンピュータシステム110は、「代入および引用」に対するDMLのターゲット言語と整合性がある引用符付きシンタックスを用いるが、どの区切り文字も保護区切り文字として認識しない。

【0038】

ターゲットシンタックス：リテラル

トークン：\$-literal

処理モード振る舞い：コンピュータシステム110は、「代入および引用」または「直接代入」パラメータバインドを実行せず、\$-literalトークンと、もしあれば、続くトークンとの間の全て（トークン直後のスペースを除く）を逐語的に出力する。

【0039】

これらのインライントークンは、対応するターゲットの式へ変換されない。

【0040】

2.2 PDL評価規則

PDLの式を用いて、パラメータ割り当てセット内に含まれる値を提供することができる。PDLの式の幾つかの部分は、PDLの式の変換に対してローカルの値を操作するステートメントを含むことができ、得られるターゲット式に直接反映されない。コンピュータシステム110は、PDL式内の文字を解釈する下記の例示の規則セットを用いる。

【0041】

2.2.1 パラメータ割り当て

ソースのPDLの式で参照されるパラメータ値は、PDLの式とは別に割り当てることができ（例えば、一セットのパラメータ割り当てを含むファイル内で）、またはPDLの式とインラインで割り当てることができる（例えば、一つ以上のPDLの式を含むスクリプト内で）。このようなインラインパラメータ割り当ては、対応するターゲットの式へ変換されない。PDLの式で参照されるパラメータの値を決定するステップは、他のPDLの式を変換するステップ、および得られるターゲット式を評価するステップを含むことができる。

【0042】

2.2.2 直接代入パラメータ参照

引用符なしでトークンに直接先行する「\$」文字は、「直接代入」パラメータ参照を指示する。パラメータ参照は、パラメータ参照に代入される値とバインドされる。例えば、式中の文字シーケンス\$BARが、文字シーケンスxxxを生成するよう評価されるとする。その場合、式中の文字シーケンスFOO\$BARは、文字シーケンスFOOxxxを生成するよう評価され

10

20

30

40

50

ることになる。

【0043】

区切り文字を用いて、参照されるパラメータを表す文字を区切ることができる。例えば、大括弧を用いることにより、文字シーケンスFOO\${BAR}BAZが、文字シーケンスFOO\$BARBAZではなく、FOOxxxBAZを生成すると評価することができる。

【0044】

2.2.3 代入および引用のパラメータ参照

一重または二重の引用符と組み合わせて用いてトークンを区切る「\$」文字は、「代入および引用」パラメータ参照を指示する。パラメータは、参照されるパラメータに代入される値にバインドされ、得られる文字シーケンスは、ターゲットシンタックスに基づいて引用される。例えば、式の文字シーケンス"\$BAR"または文字シーケンス'\$BAR'は、評価されて文字シーケンス'xxx'を生成する（一重引用符を用いるターゲットシンタックスについて、上記例示のようにBARに値xxxが割り当てられる場合）。同様に、パラメータ割り当てをTERMINATOR=|とした場合、式中の文字シーケンスstring("\$TERMINATOR")が評価されて、文字シーケンスstring('|')が生成される。

10

【0045】

別の実施例では、ターゲットシンタックスは、一本のリテラルバックスラッシュ文字「\」が、引用されるリテラル文字列内で2本のバックスラッシュ文字「\\」により指定されることを必要とする。この場合には、パラメータ割り当てをTERMINATOR=\とすると、式中の文字シーケンスstring("\$TERMINATOR")が評価されて、文字シーケンスstring('\\')が生成される。

20

【0046】

2.2.4 特殊文字のエスケープ

リテラル「\$」文字は、文字をプレフィックスとして用いてパラメータ代入の振る舞いを「エスケープ」することにより、ソースPDLシンタックスで規定することができる。例えば、エスケープ文字がバックスラッシュの場合、式内の文字シーケンスFOO\ \$BARが評価されて文字シーケンスFOO\$BARが生成される。

【0047】

「\$」文字の前にリテラル「\」文字を生成するために、二本のバックスラッシュ文字「\\」を用いることができる。例えば、式中の文字シーケンス\\ \$BARが評価されると、直接パラメータ代入が行われて文字シーケンス\ xxxが生成される。一般に、「\$」文字の前に偶数個の「\」文字がある場合、「\」文字は対にされ、「\$」文字がパラメータ参照として解釈される。「\$」文字の前に奇数個の「\」文字がある場合、隣接する「\」文字は、「\$」文字をリテラルに解釈させ、残りの「\」文字は対にされる。従って、式中の文字シーケンス\\\ \$BARは文字シーケンス\ \$BARを生成するよう評価され、式中の文字シーケンス\\\\ \$BARは、文字シーケンス\\ xxxを生成するよう評価される。

30

【0048】

2.2.5 保護区切り文字

コンピュータシステム110は、保護区切り文字内側の「直接代入」パラメータ参照および「代入および引用」パラメータ参照をともにOFFにすることにより、保護区切り文字を順守することが好ましい。例えば、文字シーケンス\$BARをリテラル文字として解釈するために、二重または一重引用符等の保護区切り文字を用いて、保護されたりテラル文字列"\$BAR"を得ることができる。更に、特殊文字のエスケープは、保護されたりテラル文字列内側では不要である。

40

【0049】

2.2.6 インライン実行

「インライン」で実行可能な計算またはスクリプトを、PDLの式を評価しながら実行することができる。このインライン実行は、別の種類の区切り文字を用いて示す。実行可能な計算またはスクリプトの名称、例えば、execを、式の中で\$(exec)として区切り、execを実行した結果（例えば、計算またはスクリプトの出力を取得したもの）で置換する。

50

【 0 0 5 0 】

インラインスクリプトは、「透明」なパラメータ参照を含むことができる。パラメータ PARMの透明な参照は、次の二つの形式、\$PARMまたは\${PARM}のどちらかの形式を取る。従って、式中の文字シーケンス\$(echo\$BAR)が評価されると文字シーケンスxxxが生成される。しかし、\$(eval echo '\$FOO)または\$(printenv FOO)のような他の種類の「不透明」なパラメータ参照の場合は、パラメータがそれらの割り当てられた値へバインドされることにはならない。

【 0 0 5 1 】

特定の種類の区切り文字を用いて、実行可能な計算またはスクリプトは特定言語で実行すべきであることを示すことができる。例えば、\$(exec)はkshに基づいてexecを実行することを示し、\$[exec]はDMLに基づいてexecを実行することを示すことができる。代替として、インライントークンを用いて特定言語での実行を指示することができる。例えば、\$(ksh exec)は、kshに基づいてexecを実行することを示し、\$(dml exec)は、DMLに基づいてexecを実行することを示すことができる。

10

【 0 0 5 2 】

3 パラメーターバインド手法

図3は、データ収納庫130内に格納される例示のパラメータ割り当て136を示す。パラメータ割り当て136は、パラメータ名称200、パラメータ値202、およびオプションで、パラメータ属性204を含む。パラメータ属性は、データ種類または記述等の、パラメータまたはその値と関係付けられる何らかの情報を含むことができる。本実施例では、パラメータに割り当てられた値は、\$FOOの値を有するパラメータDATE等の、何らかの先行するパラメータへの参照を含むことができる。

20

【 0 0 5 3 】

図4を参照すると、例示のパラメーターバインド手法では、コンピュータシステム110は、評価中のソーススクリプト302と関係付けられるパラメータに対して、ローカルメモリ120内にバインド環境300を維持する。コンピュータシステム110がソーススクリプト302を評価する時、コンピュータシステム110は、バインド環境300内に、ソーススクリプト302内で参照されるそれぞれのパラメータのコピーを維持する。バインド環境300内に格納される情報は、パラメータ名称304およびそのパラメータの現在の値306を含む。それぞれのパラメータは適切なスコープ規則に基づいて値へバインドされる。場合により、コンピュータシステム110は、バインド環境300内の多数のコンテキストを維持して、異なるスコープを有するパラメータを格納する。

30

【 0 0 5 4 】

本実施例では、ソーススクリプト302は、文字シーケンス"\$BASE/file.dml"を含むソースの第1式310を含む。この文字シーケンスは、引用符付きリテラルの式として文字列の値を表す(例えば、DML処理モードで)。コンピュータシステム110は、ソースシンタックス内の通常であればパラメータ参照\$BASEとなる筈のもののバインドを無効にする。そして、パラメータ参照\$BASEは、そのまま維持されて後でバインドされる(例えば、対応するターゲット式を評価するときに、値~\dirへバインドされる)。

40

【 0 0 5 5 】

ソーススクリプト302は、文字シーケンス\$"DATE"を含むソースの第2式312を含み、この文字シーケンスは、「代入および引用」のパラメータ参照により文字列の値を表す。参照されるパラメータDATEは、別のパラメータFOOで定義される(図3で)ので、コンピュータシステム110は、パラメータFOOをそれが割り当てられた値\$TODAYへ最初にバインドして、

May 1, 2000

と解決し、バインド環境300にその結果を格納する。次いで、コンピュータシステム110は、次の値へパラメータDATEをバインドし、

May 1, 2000

バインド環境300にその結果を格納する。

50

【0056】

ソーススクリプト302は、文字シーケンス"\$BAR"を含むソースの第3式314を含み、この文字シーケンスは、「代入および引用」のパラメータ参照により文字列の値を表す。コンピュータシステム110は、パラメータBARを次の値へバインドし、

```
test 1
```

バインド環境300にその結果を格納する。

【0057】

コンピュータシステム110は、ターゲットシンタックスでソースの第1式310を表す第1ターゲット式320を含むターゲットスクリプト308を生成する。ソース式310内の文字シーケンス"\$BASE/file.dml"は、保護区切り文字(DML処理モードに基づく)を含むので、ターゲット式320は、同一の文字シーケンスを含む。

10

【0058】

ターゲットスクリプト308は、引用される値"May 1, 2000"が、パラメータ参照"\$DATE"の代わりに代入されている第2のターゲット式322、および引用される値"test 1"が、パラメータ参照"\$BAR"の代わりに代入されている第3のターゲット式324を含む。

【0059】

コンピュータシステム110は、ソーススクリプト302をパラメータ参照に対してスキャンし、その参照した値をバインド環境300内に格納した後に、ターゲット式を生成する。代替として、コンピュータシステムは、ソーススクリプト302をスキャンしているときにターゲット式を生成することができる。次いで、コンピュータシステム110は、生成したターゲットスクリプト308を実行でき、および/または後で実行するためにターゲットスクリプト308を格納することができる。

20

【0060】

上記説明した式の評価技法は、コンピュータ上で実行するためのソフトウェアを用いて実装することができる。例えば、ソフトウェアは、一つ以上のプログラム済み、またはプログラム可能なコンピュータシステム(分散型、クライアント/サーバ型、またはグリッド型等の各種のアーキテクチャから構成される)を実行する、一つ以上のコンピュータープログラム内の手順を形成する。それぞれのシステムは、少なくとも一つのプロセッサ、少なくとも一つのデータ格納システム(揮発性メモリおよび不揮発性メモリおよび/または記憶素子を含む)、少なくとも一つの入力装置またはポート、および少なくとも一つの出力装置またはポートを含む。本ソフトウェアは、例えば、計算グラフの設計および構成と関連する他のサービスを提供する、規模の大きなプログラムの内の一つ以上のモジュールを形成してもよい。グラフのノードおよび要素は、コンピュータ可読媒体内に格納したデータ構造、またはデータ収納庫内に格納したデータモデルに準拠する他の体系化したデータ、として実装することができる。

30

【0061】

本ソフトウェアは、汎用または専用プログラム可能なコンピュータにより読み込むことができるCD-ROMのような媒体上で提供するか、またはネットワークを介してソフトウェアが実行される場所であるコンピュータへ配布する(伝搬信号で符号化される)ことができる。機能全てを専用コンピュータ上で、またはコプロセッサ等の専用ハードウェアを用いて、実行することができる。本ソフトウェアは、ソフトウェアにより規定される異なる計算部分を異なるコンピュータで実行する分散型で実装できる。このようなコンピュータープログラムはそれぞれ、汎用もしくは専用のプログラム可能なコンピュータに可読の格納媒体または装置(例えば、固体メモリもしくは媒体、または磁気もしくは光学媒体)に格納するか、ダウンロードして、格納媒体または格納装置をコンピュータシステムが読み出す際に、本明細書で説明した手順を実行するようにコンピュータを構成し、動作させることが好ましい。本発明のシステムは、コンピュータープログラムで構成されたコンピュータ可読格納媒体として実装し、そのように構成される格納媒体が、コンピュータシステムを規定の方法、所定の方法で動作させて、本明細書で説明した機能を実行させると考えることもできる。

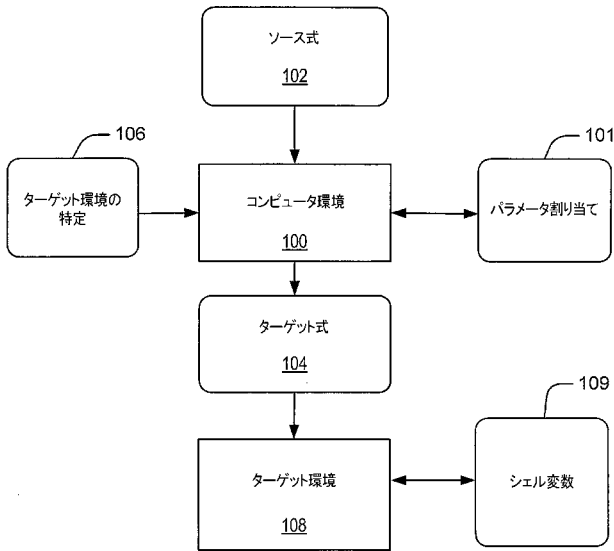
40

50

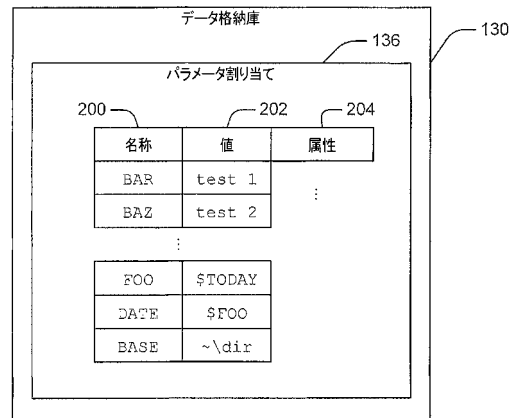
【 0 0 6 2 】

言うまでもなく、上記説明が説明を意図して本発明の範囲を限定する意図はなく、本発明の範囲は、付帯のクレーム範囲により定義されるということである。例えば、上記説明の機能ステップの幾つかは、全体の処理に実質的に影響を与えることなく異なる順序で実行してもよい。他の実施の形態は以下のクレームの範囲内にある。

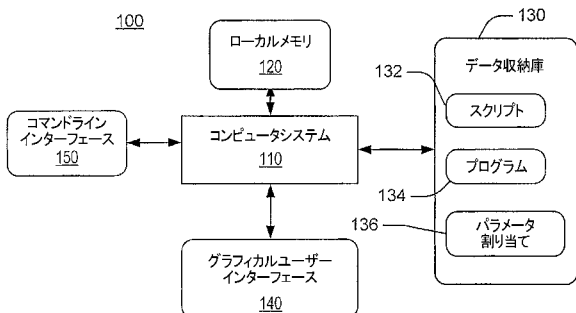
【 図 1 】



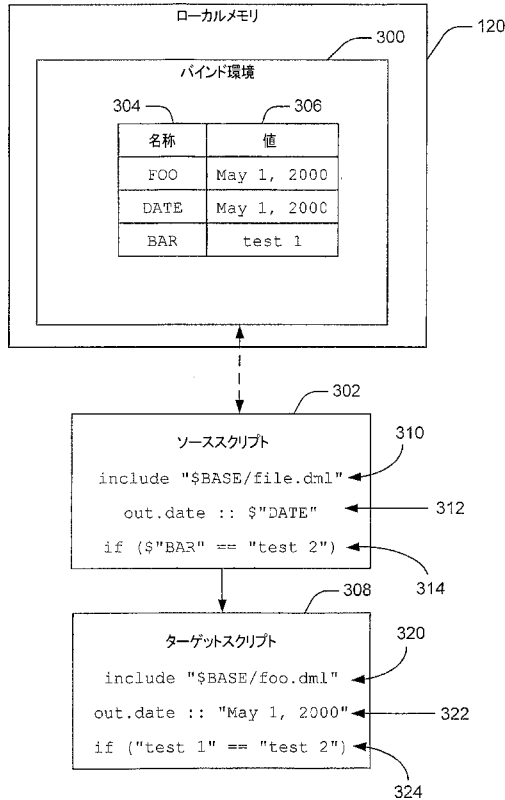
【 図 3 】



【 図 2 】



【図 4】



【手続補正書】

【提出日】平成24年6月13日(2012.6.13)

【手続補正 1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項 1】

式を評価するための方法であって、
 パラメータ定義を格納する一つ以上のデータ構造を受け取るステップと、
 前記式で参照される一つ以上のパラメータを特定するステップと、
 前記格納されたパラメータ定義に基づいて前記特定したパラメータそれぞれの値を決定するステップと、
 バインド環境内に前記決定したパラメータの値を格納するステップと、を含み、
 前記決定するステップは、前記格納されたパラメータ定義が第2パラメータを参照する場合の少なくとも一つのパラメータの値を決定することを含む、
 方法。

【請求項 2】

前記バインド環境は、前記特定したパラメータの値、および前記特定したパラメータを評価するのに必要なパラメータの値を格納し、

前記バインド環境は、前記一つ以上のデータ構造内にその定義を格納する前記パラメータ全ての値を格納しない、

請求項 1 の方法。

【請求項 3】

参照パラメータを前記バインド環境からの値へバインドさせた状態で、前記式を実行するステップを更に含む、請求項1の方法。

【請求項4】

前記式はプログラムまたはスクリプトの一部である、請求項1の方法。

【請求項5】

前記式はコマンドライン入力を含む、請求項1の方法。

【請求項6】

式を変換するためのコンピュータ可読記憶媒体上に格納されるプログラムであって、コンピュータシステムに、

パラメータの定義を格納する一つ以上のデータ構造を受け取るステップと、

前記式で参照される一つ以上のパラメータを特定するステップと、

格納された定義に基づいて前記特定したパラメータそれぞれの値を決定するステップと

、
バインド環境内に前記決定したパラメータの値を格納するステップと、
を実行させるための命令を含む、
プログラム。

【請求項7】

式を評価するためのシステムであって、

パラメータ定義を格納する一つ以上のデータ構造を受け取るための手段と、

前記式で参照される一つ以上のパラメータを特定するための手段と、

格納された定義に基づいて前記特定したパラメータそれぞれの値を決定するための手段と、

バインド環境内に前記決定したパラメータの値を格納するための手段と、を含む、
システム。

【請求項8】

前記式で参照される一つ以上のパラメータを特定するステップは、前記格納された定義
で参照される一つ以上のパラメータを特定するステップを含む、請求項1の方法。

【外国語明細書】

2012160214000001.pdf