

令和3年2月4日判決言渡

令和元年（行ケ）第10106号 審決取消請求事件

口頭弁論終結日 令和2年11月25日

判 決

5

原 告 ソフトバンクロボティクス株式会社

同訴訟代理人弁護士 鮫 島 正 洋
和 田 祐 造
10 森 下 梓

被 告 株式会社パッセルインテグレーション

15

同訴訟代理人弁護士 中 村 隆 夫
加 藤 伸 樹
我 妻 崇 明

主 文

- 1 原告の請求を棄却する。
- 2 訴訟費用は原告の負担とする。

20

事 実 及 び 理 由

第1 請求

特許庁が無効2018-800104事件について令和元年6月25日に
した審決を取り消す。

第2 事案の概要

25

1 特許庁における手続の経緯等

(1) 被告は、平成16年8月13日、発明の名称を「情報管理方法、情報管理

プログラム，及び情報管理装置」とする発明について特許出願（特願 2004-235768号〔以下「本件特許出願」という。〕，優先日・平成16年7月28日。甲18）をし，平成17年12月22日，特許権の設定登録（特許第3754438号。請求項の数15。）を受けた（以下，この特許を「本件特許」という。甲29）。

5

(2) 原告は，平成30年8月20日，本件特許の請求項1及び14について特許無効審判を請求した（甲29）。

特許庁は，上記請求を無効2018-800104号事件として審理を行い，令和元年6月25日，「本件審判の請求は，成り立たない。」，との審決（以下「本件審決」という。）をし，その謄本は，同年7月4日，原告に送達された。

10

(3) 原告は，令和元年8月1日，本件審決の取消しを求める本件訴訟を提起した。

2 特許請求の範囲の記載

請求項1項及び14項の特許請求の範囲の記載は，次のとおりである（甲18）。なお，請求項1は，「A」ないし「H」なる記号により分説されているわけではないが，本件審決において便宜のためこれらの記号が付されたものである。

15

【特許請求の範囲】

【請求項1】

20

A コンピュータが情報を管理する情報管理方法であって，
B 前記コンピュータに複数のノードそれぞれに対応付けて入力された管理すべき情報を，前記ノードを識別するノード識別情報に対応付けられた複数のノードデータを含む文書ファイルとして前記コンピュータが記憶する情報記憶ステップと，

25

C 前記情報記憶ステップで記憶された前記文書ファイルの情報を前記コンピ

ュータが表示する情報表示ステップと、

- D 前記ノードデータに含まれるスクリプトを前記コンピュータが実行する情報評価ステップとを備え、
- E 前記ノードデータは、ルートノードを除いて、当該ノードの親ノードを特定する親ノード識別情報を含んでおり、
- F 前記スクリプトは、当該ノードデータに含まれる変数データである自ノード変数データと、当該ノードの直系上位ノードのノードデータに含まれる変数データである上位ノード変数データを利用した演算を行って、前記自ノード変数データの値を求める代入用スクリプトを含んでおり、
- G 前記情報表示ステップは、前記親ノード識別情報を利用して、前記ノードの木構造を表示する木構造表示ステップと、前記表示された木構造のノードのうちの選択されたノードの前記自ノード変数データ、前記上位ノード変数データ及び前記スクリプトを表示するノードデータテーブル表示ステップを含み、
- H 前記情報評価ステップは、前記代入用スクリプトの実行により、前記自ノード変数データの値を更新するステップを含む情報管理方法。

【請求項 1 4】

請求項 1 ないし 1 3 のいずれか 1 項記載の情報管理方法における各ステップを、コンピュータに実行させるための情報管理プログラム。

3 本件審決の理由の要旨

- (1) 本件審決の理由の要旨は、①本件特許の請求項 1 に係る発明（以下「本件発明 1」という。）及び本件特許の請求項 1 4 に係る発明（以下「本件発明 1 4」という。）は、本件特許出願の優先日前に頒布された刊行物である甲第 1 号証（特開平 6 - 1 7 5 8 5 2 号公報。以下、単に「甲 1」という。その他の刊行物についても同様に単に「甲 2」等という。）に記載された発明ではないから、原告主張の新規性欠如（特許法 2 9 条 1 項 3 号）の無効理由

は理由がない，②本件発明 1 及び本件発明 1 4 は，甲 1 に記載された発明において，相違点に係る構成を想起することは，当業者が容易になし得たものでないから，原告主張の進歩性欠如（同条 2 項違反）の無効理由は理由がない，③本件発明 1 及び本件発明 1 4 に係る本件明細書（甲 1 8）の発明の詳細な説明は，当業者に実施可能なように記載されており，同法 3 6 条 4 項 1 号に規定する要件（以下「実施可能要件」という。）に適合するから，原告主張の実施可能要件違反の無効理由は理由がない，④本件発明 1 及び本件発明 1 4 は，本件明細書の発明の詳細な説明に記載されたものであり，発明の詳細な説明の記載により当業者が当該発明の課題を解決できる範囲のもので，同条 6 項 1 号に規定する要件（以下「サポート要件」という。）に適合するから，原告主張のサポート要件違反の無効理由は理由がないというものである。

(2) 本件審決が認定した甲 1 に記載された発明（以下「甲 1 発明」という。），並びに本件発明 1 と甲 1 発明の一致点及び相違点は，次のとおりである。

ア 甲 1 発明

甲 1 - a コンピュータが知識ベースを構築する知識ベース構築方法であって，

甲 1 - b 前記コンピュータに複数のノードそれぞれに対応付けて入力された知識を，前記ノードを定義するノード名称に対応付けられた複数のノードデータを含むデータベースに前記コンピュータが記憶し，更に前記知識を知識記述言語に変換して知識ベースとして記述する記憶ステップと，

甲 1 - c 前記記憶ステップで記憶された知識を前記コンピュータが表示する表示ステップと，

甲 1 - d 前記ノードデータに含まれる属性値の数式の計算を前記コンピュータが実行する実行ステップとを備え，

甲 1 - e 前記ノードデータは、最上位のノードを除いて、当該ノードの親ノードを特定する親ノードの名称を含み、前記ノードデータには前記数式のほか条件分岐も含み得、

5 甲 1 - f 前記数式は、当該ノードの親ノードのノードデータに含まれる属性値である親ノード属性値（『内箱の本体の高さ』）を当該ノードデータに含まれる属性値である自ノード属性値（『高さ』）として代入する数式を含んでおり、

10 甲 1 - g 前記表示ステップは、前記親ノードの名称を利用して、前記ノードのツリーを表示するツリー表示ステップと、前記表示されたツリーのノードのうちの選択されたノードの前記自ノード属性値、前記親ノード属性値を表示する属性値表示ステップを含み、

15 甲 1 - h 前記実行ステップは、前記数式の計算により、親ノード属性値を自ノード属性値として代入して、自ノード属性値を求めるステップを含む知識ベース構築方法。

イ 本件発明 1 と甲 1 発明の一致点及び相違点
(一致点)

「コンピュータが情報を管理する情報管理方法であって、

20 前記コンピュータに複数のノードそれぞれに対応付けて入力された管理すべき情報を、前記ノードを識別するノード識別情報に対応付けられた複数のノードデータを含む情報として前記コンピュータが記憶する情報記憶ステップと、

前記情報記憶ステップで記憶された前記情報を前記コンピュータが表示する情報表示ステップと、

25 前記ノードデータに含まれる値に係る演算をコンピュータが実行する実行ステップとを備え、

前記ノードデータは、ルートノードを除いて、当該ノードの親ノードを特定する親ノード識別情報を含んでおり、

5 前記情報表示ステップは、前記親ノード識別情報を利用して、前記ノードの木構造を表示する木構造表示ステップと、前記表示された木構造のノードのうちの選択されたノードの前記自ノード変数データ、前記上位ノード変数データを表示するノードデータテーブル表示ステップを含み、

前記実行ステップは、前記自ノード変数データの値を更新するための演算を実行する情報管理方法。」である点。

10 (相違点 1)

本件発明 1 が、「管理すべき情報」を「文書ファイル」として記憶し、当該「文書ファイルの情報」をコンピュータが表示するものであるのに対し、甲 1 発明は、「知識」を「データベース」に記憶し、「更に前記知識を知識記述言語に変換して知識ベースとして記述」し、当該「知識」
15 をコンピュータが表示するものである点。

(相違点 2)

本件発明 1 が、「ノードデータ」に「スクリプト」を含み、当該スクリプトは「当該ノードデータに含まれる変数データである自ノード変数データと、当該ノードの直系上位ノードのノードデータに含まれる変数データである上位ノード変数データを利用した演算を行って、前記自ノード変数データの値を求める代入用スクリプト」を含むものであり、情報表示ステップでは、前記「スクリプト」も表示されるのに対し、甲 1 発明では、ノードデータにスクリプトが含まれることは特定されておらず、更に当該スクリプトを表示することも特定されていない点。

25 (相違点 3)

本件発明 1 が、「情報評価ステップ」で「前記代入用スクリプトの実

行により，前記自ノード変数データの値を更新する」ものであるのに対し，甲1発明は，「実行ステップ」において「ノードデータに含まれる属性値の数式の計算」が行われ，計算されたノードデータに基づき「知識ベース」の更新が行われるものの，「情報評価ステップ」で「代入用スクリプト」の実行によって「自ノード変数データの値を更新」を行うものではない点。

第3 当事者の主張

1 取消事由1（新規性の判断の誤り）

(1) 原告の主張

ア 相違点2の認定に誤りがあることについて

(ア) a 甲1発明にはスクリプトが存在するから，本件審決の相違点2の認定は誤りである。

b 本件明細書では，「次に，代入用スクリプト及び生成用スクリプトの具体例を，図8の「*」を付したノードをターゲットノードとして説明する。図9に示すように，ターゲットノードは，要計算の公開変数として，「スライス数」と「色」を有しており，代入用スクリプトとして，「スライス数＝同一面数；」と「色＝同一面数」を有している。評価前は，図10に示すように，公開変数「スライス数」と「色」の値は空欄となっている。」とし（【0072】），スクリプトの具体的実施態様として，「スライス数＝同一面数；」及び「色＝巾木色；」という2つの数式を開示する。本件発明1において，スクリプトとは，単なる代入表現をその範囲に含むものである。

一方，甲1発明では，「“内箱”のノードでは，“内箱本体”に入力すべき3つの属性（“高さ” “幅” “長さ”）を自己の大きさから決定する。又，“内箱”自身の大きさは，親ノードである“引き出し”にその決定を委ねる。」（【0030】），「この様にして全てのデ

ータ入力作業が終了すれば、データベース内には、図6に示す如きイメージでデータ構造が登録され、最上位のノードである“引き出し”の“高さ” “幅” “長さ”が、知識ベースに対する入力データとなる。」（【0031】）として、例えばコンピュータが、「内箱本体の高さ」（自ノードである「内箱本体」の入力属性である「高さ」を意味する属性値の名称である。）に対し、内箱の「自己の高さ」（親ノードである「内箱」の入力属性である「高さ」を意味する属性値の名称である。）の値を代入する態様が記載されている。具体的には、甲1の【図6】の

内箱本体の高さ	自己の高さ
---------	-------

との記載がそれに該当する。これは、「自己の高さ」という名称を持つ属性の値を、「内箱本体の高さ」という名称を持つ属性の値に代入することを示す表記であるから、単なる代入表現をスクリプトの範囲に含む本件発明1との関係では、このような記載もまたスクリプトを開示するものといふことができ、プログラム言語で表現すると、実質的に、

内箱本体の高さ＝自己の高さ；

という代入式が記載されているに等しいと考えることができる。

さらに、甲1には、「属性値の定義（S6）では、決定属性に分類された属性に対して属性値（“黒”，“スチール”）の入力が行なわれる。ここで、属性値としては、数値に限らず、数式や条件分岐が入力可能である。」（【0022】）との記載もあり、属性値として、数式だけでなく、条件分岐も記載することができ、これがスクリプトであることに疑いはない。

(イ) さらに、甲1に、本件発明1の構成要件Fの「当該ノードデータに含まれる変数データである自ノード変数データと、当該ノードの直系上

位ノードのノードデータに含まれる変数データである上位ノード変数データを利用した演算を行って、前記自ノード変数データの値を求める代入用スクリプトが開示されていることは、以下のとおりである。

甲1発明では、自ノードを「内箱本体」とした場合、この直系上位ノード「内箱」の変数データ「高さ」の値を、自ノード「内箱本体」の変数データ「高さ」に代入する甲1の【図6】の

内箱本体の高さ	自己の高さ
---------	-------

との表記が、上位ノード変数データ「自己の高さ」（＝直系上位ノード「内箱」の入力属性「高さ」）を、自ノード変数データ「内箱本体の高さ」（＝自ノード「内箱本体」の入力属性「高さ」）に代入して、その値を求める代入用スクリプトである。仮に、「内箱本体の高さ」、「内箱本体の幅」及び「内箱本体の長さ」は上位ノード変数データに該当し、自ノード「内箱本体」の入力属性である「高さ」、「幅」及び「長さ」は自ノード変数データに該当すると解したとしても、上位ノード変数データ「内箱本体の高さ」を、自ノード「内箱本体」の自ノード変数データ「高さ」に代入して、その値を求めていることになる。

ところで、この代入用スクリプトは、自ノード変数データ「高さ」が含まれている自ノード「内箱本体」ではなく、直系上位ノード「内箱」に記載されている。しかし、本件発明1において、代入用スクリプトと自ノード変数データとが、同一のノードデータに含まれることは要件とされていないから、この点が、本件発明1と甲1発明との相違点となることはない。すなわち、本件発明1では、構成要件Bで「複数のノードデータを含む文書ファイル」が規定されていることから、本件発明1は、文書ファイル中に、1つではなく、2つ以上のノードデータを具備するものであり、これを受けた構成要件Dの「前記ノードデータ」、構成要件Fの「当該ノードデータ」は、いずれも構成要件Bの「複数のノード

5

データ」を意味する。構成要件Dは、①「複数のノードデータのいずれかがスクリプトを含む」ことを規定し、構成要件Fは、②「代入用スクリプトが、複数のノードデータのいずれかに含まれる自ノード変数データと、その上位ノード変数データとを利用した演算により自ノード変数データを求める」ことと、③「スクリプトが代入用スクリプトを含む」こととを規定している。代入用スクリプトも、自ノード変数データも、共に複数のノードデータのいずれかに含まれていれば足りるものであり、両者が同じノードに含まれていることは、一切規定されていない。

10

仮に甲1発明において、代入用スクリプトと自ノード変数データとが同じノードに含まれることが要件であるとしても、甲1には代入用スクリプトの開示がある。すなわち、甲1の【図6】の自ノード「内箱本体」における、

入力属性	高さ
	…
	…

15

との記載は、自ノード「内箱本体」の変数「高さ」に対し、上位ノード変数の値を入力するスクリプトを意味する。そして、その際に入力に用いられるのは、

内箱本体の高さ	自己の高さ
---------	-------

との代入用スクリプトである。そうすると、甲1の【図6】における自ノード「内箱本体」には、結局のところ、

$$\text{「高さ」} = (\text{「内箱本体の高さ」} = \text{「自己の高さ」}) ;$$

20

という、親ノード変数データの値「内箱本体の高さ」を、自ノード変数データ「高さ」に代入するための、代入用スクリプトが含まれている。

(ウ) このように、甲1には、代入用スクリプトの開示があることを前提とすれば、甲1発明は、正しくは以下のとおり認定されるべきである。

- 甲 1 - a コンピュータが知識ベースを構築する知識ベース構築方法であって、
- 5 甲 1 - b 前記コンピュータに複数のノードそれぞれに対応付けて入力された知識を、前記ノードを定義するノード名称に対応付けられた複数のノードデータを含むデータベースに前記コンピュータが記憶し、更に前記知識を知識記述言語に変換して知識ベースとして記述する記憶ステップと、
- 甲 1 - c 前記記憶ステップで記憶された知識を前記コンピュータが表示する表示ステップと、
- 10 甲 1 - d 前記ノードデータに含まれる属性値のスク립トを前記コンピュータが実行する実行ステップとを備え、
- 甲 1 - e 前記ノードデータは、最上位のノードを除いて、当該ノードの親ノードを特定する親ノードの名称を含み、前記ノードデータには前記スク립トを含み、
- 15 甲 1 - f 前記スク립トは、当該ノードの親ノードのノードデータに含まれる属性値である親ノード属性値（『内箱の本体の高さ』）を当該ノードデータに含まれる属性値である自ノード属性値（『高さ』）として代入するスク립トを含んでおり、
- 甲 1 - g 前記表示ステップは、前記親ノードの名称を利用して、前記ノードのツリーを表示するツリー表示ステップと、前記表示されたツリーのノードのうちの選択されたノードの前記自ノード属性値、前記親ノード属性値を表示する属性値表示ステップを含み、
- 20 甲 1 - h 前記実行ステップは、前記スク립トにより、親ノード属性値を自ノード属性値として代入して、自ノード属性値を求めるステップを含む
- 25

知識ベース構築方法。

(エ) 本件審決が、相違点2を本件発明1と甲1発明の実質的相違点とした根拠は、専ら甲1の「数式や条件分岐」が「スクリプト」とは異なるという点に依拠したものであるから、甲1の数式や条件分岐がスクリプトに該当するものである以上、本件審決の相違点2の判断には誤りがある。

この点は、本件発明14についても同様である。

イ 相違点3の認定に誤りがあることについて

(ア) 本件審決が、相違点3を本件発明1と甲1発明の実質的な相違点とした理由の1つは、甲1にスクリプトが開示されていないというものであり、これが誤りであることは、前記アで主張したとおりである。

(イ) 本件審決は、本件発明1の構成要件Hについて、「シミュレーション動作のために具体的には、図6の「操作ボタン表示領域20a」にある、「評価」ボタンを押下し、その結果として「自ノード変数データの値」の値が更新されることを意味するものと解される。」とし、この点が甲1発明に記載されていないことも、実質的な相違点と認定している。

a 本件発明1の記載を正しく解釈すれば、構成要件Hの「更新」ステップは、文書ファイルに含まれている自ノード変数データの値を変更するものであれば足りる。本件審決が、本件発明1の更新ステップが「操作ボタン表示領域20aにある評価ボタンを押下する」との態様に限定される根拠とする本件明細書の記載は、いずれも実施例にすぎない。

b 甲1発明では、構成要件1-hは「前記実行ステップは、前記スクリプトにより、親ノード属性値を自ノード属性値として代入して、自ノード属性値を求めるステップを含む」というものであるが、ここで

5 スクリプトにより代入された自ノード属性値は、甲1の「…最上位のノードに入力属性として定義されている属性名が、知識ベース運用の際にデータを入力すべき項目であり、該データ入力によって知識ベースが動作を開始することになる。」（【0023】）、「…あるノードの知識を変更する場合は、詳細知識定義装置(6)を用いてデータベース(4)の該当箇所を書き換えるだけで、これが知識ベース(1)の知識に反映される。」（【0026】）との記載から明らかなように、知識ベースの知識に反映されることとなる。

10 そして、知識ベースが文書ファイルであることについては、審決において、相違点1が実質的相違点でないとして適切に認定されたとおりのため、甲1には、文書ファイルに記載された自ノード属性値をスクリプトにより変更することが開示されている。

ウ 新規性の判断に誤りがあることについて

15 前記ア及びイのとおり、正しく認定された甲1発明を前提とすれば、本件発明1と甲1発明の相違点2及び相違点3は存在しない。

同様に、本件発明14と甲1発明の相違点2及び相違点3も存在しない。したがって、本件審決の判断は誤りである。

エ 小括

20 以上のとおり、本件発明1と甲1発明に相違点は存在しないから、本件発明1は、本件特許出願の優先日前に頒布された刊行物である甲1に記載された発明ではないとした本件審決の判断は誤りである。また、同様に、本件発明1の発明特定事項を全て含む本件発明14について、本件特許出願の優先日前に頒布された刊行物である甲1に記載された発明ではないとした本件審決の判断は誤りである。

25 (2) 被告の主張

ア 相違点2の認定に誤りがないことについて

(ア) 本件発明1における「スクリプト」の意義

- a スクリプトとは「コンピューターで、一連の処理手順を記述した簡易プログラム」（甲14）、「プログラムやマクロを記述したテキストファイル」、「コンピューターに処理を自動的に実行させるための命令を記述したファイル」（乙1）を意味し、このようなスクリプトを記述する言語を「スクリプト言語」という（乙2）。
- b 本件発明1において、「スクリプト」は、スクリプト言語として使用されている任意の言語を使用して作成されるものである（本件明細書【0016】、【0019】、【0034】、【0057】）。
- c 本件発明1の構成要件Gの「…スクリプトを表示するノードデータテーブル表示ステップ」の記載及び本件明細書【0046】の記載から明らかなおり、スクリプトは画面に表示される。
- d 本件明細書【0057】の記載から、「スクリプト」は、本件発明を利用する者がスクリプト領域又は別のウィンドウを通して直接入力できるものである。
- e 本件明細書【0067】以下及び【図12】のおり、「スクリプト」はそのまま実行される。
- f 以上によれば、本件発明における「スクリプト」は、①任意のスクリプト言語を使用して作成される簡易なプログラムであって、②作業者が、スクリプトの画面表示を通じて、直接入力し、実行できるプログラムをいう。

(イ) 本件発明1における「代入用スクリプト」の意義

本件明細書【0032】の記載から、本件発明1における「代入用スクリプト」は、(ア)のおりの性質を有する簡易なプログラムであって、かつ、自ノードの変数の値を演算するために、自ノードの変数の値である自ノード変数データと、そのノードの直系上位ノードの公開変数の値

である上位ノード変数データを利用して記述されるものをいう。

(ウ) 甲1発明における「スクリプト」ないし「代入用スクリプト」の不存在

5 a 甲1発明において、「知識ベース」は「知識記述言語」で記述されるところ、この「知識ベース」は、所定の「知識記述言語」で記述された一種のコンピュータプログラムであり、「知識記述言語」はプログラミング言語である。

10 甲1発明において、データベース内のデータは、「データ変換装置」により「知識記述言語」に変換されるから、データベース内のデータは、「知識記述言語」すなわちプログラミング言語、スクリプト言語で記載されたものではなく、「スクリプト」に当たらない。

15 b 原告は、甲1の【図6】がスクリプトであると主張するが、そこから読み取れるのは、「内箱」ノードの「内箱本体の高さ」と「自己の高さ」が同じ値となるという限度であり、「内箱」ノードの「高さ」を「内箱本体」ノードの「高さ」に代入しているとはいえない。

また、甲1には、原告の指摘する記載（内箱本体の高さ・自己の高さ）が、知識記述言語、すなわちプログラミング言語、スクリプト言語であるとの記載も示唆もない。

20 甲1発明は、「知識ベースを構築する際には、知識記述言語についての知識や経験が必要である」（【0002】）という従来技術の課題に対し、「知識記述言語（すなわちプログラミング言語、スクリプト言語）についての知識や経験を有しない知識所有者であっても、階層構造定義装置(5)、詳細知識定義装置(6)及び知識参照装置(7)を操作することによって、容易に知識ベース(1)を構築することが出来る」ことを作用効果とするものであり（【0032】）、データベース内に知識記述言語（すなわち、プログラミング言語、スクリプト言

語)の記載を求めることは、上記作用効果を阻害するものであって、
甲1発明の【図6】に「スクリプト」が開示されているとはいえない。

イ 相違点3の認定に誤りがないことについて

5 本件審決において、相違点3は、「情報評価ステップ」で「代入用スクリプト」の実行によって「自ノード変数データの値を更新」する点に求められており、ボタンの押下についてのみ指摘する原告の主張は失当である。

ウ 新規性の判断に誤りがないことについて

原告主張の甲1発明を認定することはできないから、本件発明1と甲1発明の間には、本件審決認定のとおり相違点2及び相違点3が存在し、これらは実質的なものである。

10 同様に、本件発明14と甲1発明の間には相違点2及び相違点3が存在し、これらは実質的なものである。

エ 小括

15 以上のとおり、本件発明1と甲1発明には相違点が存在し、それは実質的なものであるから、本件発明1は、本件出願の優先日前に頒布された刊行物である甲1に記載された発明ではないとした本件審決の判断には誤りがない。また、同様に、本件発明1の発明特定事項を全て含む本件発明14について、本件出願の優先日前に頒布された刊行物である甲1に記載された発明ではないとした本件審決の判断には誤りがない。

20 したがって、原告主張の取消事由1は理由がない。

2 取消事由2 (甲1発明を主引用例とする進歩性判断の誤り)

(1) 原告の主張

25 ア 相違点2及び相違点3に係る容易想到性判断に誤りがあることについて
甲1発明は、正しくは前記1(1)ア(ウ)のとおり認定されるべきであるところ、この場合、本件発明において、代入用スクリプトと自ノード変数データとが、同一のノードデータに含まれるのに対し、甲1発明ではそう

ではないことが、本件発明 1 と甲 1 発明の相違点に当たることになり、甲 1 発明において、直系上位ノード「内箱」に含まれている代入用スクリプトを、直系上位ノードではなく、自ノードに含ませることとすることの容易想到性が問題となる。

5 (ア) 甲 3 には、代入用スクリプトを、当該スクリプトによって値が代入されることになる変数データを含んだ自ノードに記載することが開示され、甲 7 には、階層構造を有するデータにおいて、スクリプトを、当該スクリプトによって値が演算されることになるノード変数データを含んだ自ノードに記載することが開示されている。さらに、甲 2 には、ノードデータにおいて、代入用スクリプトを、当該代入用スクリプトによつて値が代入されることになる変数データを含んだ自ノードに記載することが示唆されている。

10 したがって、当業者において、甲 1 発明について、本件発明 1 との相違点に係る構成である「代入用スクリプトが自ノードに含まれる」ものとするのは、設計事項にすぎないものであり、容易に想到できる。

15 (イ) 被告は、甲 1 発明では下位から上位への順に定義する旨が記載されており、これを上位から下位という順に変更することについて動機付けがなければ構成の変更は容易とはいえないと主張するが、被告の指摘する「下位ノードから上位ノードに向かって、必要な属性値の定義等を行っていく」という態様は、甲 1 発明の一実施例にすぎないものである。

20 イ 本件審決認定に係る相違点 2 の容易想到性判断に誤りがあることについて

相違点 2 を本件審決のとおり認定したとしても、本件審決の相違点 2 の容易想到性の判断には誤りがある。

25 本件審決は、知識記述言語についての知識や経験を持たない知識所有者自身によって操作されるデータベースを介して操作することにより、知識

ベース構築者自身が知識ベースを容易に構築することが出来るようにする
という甲1発明の課題や目的に照らせば、「ノードデータ」に「スクリ
プト」を含ませ、さらに当該スクリプトに「代入用スクリプト」を含むよ
う構成することは、その動機付けを欠き阻害要因を有するとか、数式や条
件分岐を「スクリプト」に変換して表示させる必要性はないと判断した。

5

しかし、甲1発明によって達成されるのは、システム開発者が間に立つ
ことなく、知識をデータベース経由で知識ベースに入力可能となる点及び
知識の体系化が自動的に行われる点に限られるのであり（【0012】）、
甲1発明において、ノードデータに記載される数式や条件分岐をスクリ
プトで記載し、表示することができることとしても、①知識を知識所有者が
自らデータベースに入力可能である、②知識の体系化が自動的に行われる、
という甲1発明の効果が阻害されることはない。

10

また、本件発明1の「スクリプト」は、単なる代入表現をその範囲に含
むものであり、かかる表現であれば、知識記述言語についての知識や経験
を有することのない知識所有者であっても、容易に記述することができる。

15

仮に、甲1発明が、プログラムの知識や経験がない知識所有者が直接知
識を追加・修正することができるよう、データベースに記載する数式や条
件分岐をスクリプトで記述することを認めないものであるとしても、知識
記述言語についての知識や経験を有するシステム開発者が、知識ベースの
知識を直接見ることができることとするため、知識ベースにスクリプトと
して反映された数式や条件分岐等の属性値を表示するための機能を付加
することは、上記甲1発明の課題や目的に反するものではない。

20

したがって、本件審決における相違点2に関する容易想到性の判断は誤
りである。

25

ウ 本件審決認定に係る相違点3の容易想到性判断に誤りがあることについ
て

甲1発明において、ノードデータにスクリプトを記載し、表示すること、及び「評価」ボタンにより「自ノード変数データの値を更新」することは、いずれも当業者の設計事項であり、阻害要因はない。

したがって、本件審決の相違点3の容易想到性の判断は誤りである。

5 エ 小括

以上によれば、本件発明1は、甲1発明及び甲2ないし10に記載された発明に基づいて、当業者が容易に発明をすることができたものであるから、これを否定した本件審決の判断は誤りである。また、同様に、本件発明1の発明特定事項を全て含む本件発明14についても、甲1発明及び甲
10 2ないし10に記載された発明に基づいて、当業者が容易に発明をすることができたものであるから、これを否定した本件審決の判断は誤りである。

(2) 被告の主張

ア 原告主張のとおり甲1発明を認定したとしても、本件審決の容易想到性判断に誤りがないことについて

15 仮に、甲1発明が前記1(1)ア(ウ)のとおり認定されるべきであるとしても、甲3には、本件発明1における意義の「ノード」、すなわち「ノードの木構造」を「表示」できる態様のノードの記載はないし、甲2及び甲7には「スクリプト」は開示されていない。

したがって、当業者において、甲1発明について、本件発明1との相違
20 点に係る構成である「代入用スクリプトが自ノードに含まれる」ものとすることが設計事項にすぎないとはいえない。

また、甲1発明では下位から上位への順に定義する旨が記載されており、これを上位から下位という順に変更することについて動機付けがなければ、原告のいうような構成の変更は容易とはいえないし、甲1【0029】
25 及び【0030】が、自ノードの属性を「親ノード…にその決定を委ねる」旨記述し、【図6】で「親ノード…にその決定を委ねる」態様として、親

ノード側に代入関係を記載していること，甲1【0028】のとおり，当初のデータ作成において下位から上位にデータを入力していく態様が取られていることからすれば，甲1発明において上位ノードから下位ノードに数値を引き継ぐ場合の代入関係は上位ノードに記述されることが予定され，下位ノードに記述することは想定していないと考えるのが自然であり，当業者がソフトウェアの効率性を犠牲にしてまで代入関係を上位ノードに記述する態様に加えて下位ノードに記述できる態様を加える技術的意義はない。

イ 本件審決認定に係る相違点2の容易想到性判断に誤りがないことについて

ノードデータにスクリプトを記載させるということは，甲1発明の利用者がスクリプトを記載できることを意味するところ，甲1発明は，プログラミング言語を扱えない知識所有者による利用を想定した技術なのであるから，甲1発明の数式や条件分岐をスクリプトとして記載することはむしろ甲1発明の目的に反する。

ウ 本件審決認定に係る相違点3の容易想到性判断に誤りがないことについて

相違点3に関する事項は，当業者の設計事項であり，阻害要因はない旨の原告の主張は，その前提を誤り，かつ理由がないものであり，本件審決における相違点3に関する容易想到性の判断に誤りはない。

エ 小括

以上によれば，本件発明1及び本件発明14は，甲1発明及び甲2ないし10に記載された発明に基づいて，当業者が容易に発明をすることができたものではないから，本件審決の判断に誤りはない。

したがって，原告主張の取消事由2は理由がない。

3 取消事由3（実施可能要件違反）

(1) 原告の主張

ア 本件発明 1 は，被告の主張によれば，文書ファイルの内容を変更しない態様を含むものであるが，この態様は，明細書に一切記載がない。

したがって，本件明細書は，当業者が過度の試行錯誤を行う必要なく，
5 本件発明を実施することができる程度に記載されたものとはいえず，本件
審決の判断には誤りがある。

イ なお，原告は，審判手続において，被請求人（被告）の主張によれば，
本件発明は，文書ファイルに含まれる自ノード変数データの値を書き換え
ない態様を含むものであるから，実施可能要件違反であるとの内容を主張
10 したが，本件審決は，上記原告の主張が「専ら本件無効審判事件と関連す
る侵害訴訟においてなされる，被請求人の主張に基づくものであり，当該
訴訟は，本件無効審判事件と関連するものであることは推察されるものの，
本件無効審判事件の，とりわけ無効理由 3 について検討するにあたっては
考慮されない」と判断している。しかし，本来，本件審決は，本件発明が
15 文書ファイルに含まれる自ノード変数データの値を書き換えない態様を
含むものであるか，それとも含まないものであるかについて判断した上で，
無効理由 3 についての結論を示すべきであったものであり，不当である。

ウ 本件発明 1 は実施可能要件に適合しないから，本件発明 1 4 も実施可能
要件に適合しない。

したがって，これと異なる本件審決の判断は誤りである。

(2) 被告の主張

ア 本件明細書には，自ノード変数データの値を書き換えない実施例が開示
されており，原告の主張には理由がない。

すなわち，実施例では，表示画面（【図 6】）がツリービューアとデザ
インテーブルから構成される（【0 0 4 4】）。デザインテーブルは，ツ
25 リービューアに表示されたノードのうち選択されたノードが有する情報

を表示する領域であり、このうち操作ボタン表示領域20aには、デザインテーブルに関する各種操作を行うための操作ボタンが表示されている（【0046】）。表示ボタンには、「更新」ボタンと「評価」ボタンが含まれている。

5 「評価」ボタンは、代入用スクリプトを実行するものである（【0067】）。「評価」ボタンを押下することにより、代入用スクリプトが実行される。これにより、公開変数「スライス数」の値に、上位ノードの公開変数である「同一面数」の値「1」が代入される（「色」「巾木色」も同様）（【0073】）。そして、その結果はデザインテーブルに表示される（【図13】）。

10

このようにして変更されたデザインテーブルの編集内容は、「更新」ボタンを押すことによって、文書情報に反映される（【0058】）。

15

構成要件Hの文言は「前記情報評価ステップは、前記代入用スクリプトの実行により、前記自ノード変数データの値を更新するステップを含む」であり、特許請求の範囲としてクレームされているのは代入用スクリプトの実行による情報評価ステップである。これは、上記実施例でいえば、「評価」ボタンの機能を指すものであり、「更新」ボタンの機能を含むものではない。すなわち、情報を表示画面（実施例ではデザインテーブル）に表示するところまでをクレームしているのであり、その後の文書情報に反映する機能はクレームには含んでいない。

20

この点は、本件審決も「評価」ボタンの押下が、「代入用スクリプト」の実行に該当することを適切に認定しているところである。

25

本件審決は、構成要件Hの「更新」が、「評価」ボタンの押下により実行されるものであり、文書ファイルの書換えを伴うものではないことを前提に実施可能性要件を判断したものであり、本件発明1が実施可能要件に適合するとした本件審決の判断に誤りはない。

イ 本件発明 1 は、実施可能要件に適合するから、本件発明 1 が実施可能要件に適合することを前提に、本件発明 1 4 が実施可能要件に適合するとした本件審決の判断に誤りがあるとの原告の主張も理由がない。

したがって、原告主張の取消事由 3 は理由がない。

5 4 取消事由 4 (サポート要件違反)

(1) 原告の主張

ア 本件発明 1 は、被告の主張によれば、文書ファイルの内容を変更しない態様を含むものである。ところが、本件明細書を参酌しても、文書ファイルに含まれる自ノード変数データの値を書き換えることなく、どのようにして情報の更新を行うのか、どのように多数の利用者間で更新した情報を共有し、再利用することができるのか全く明らかでない。

イ 本件審決は、被請求人（被告）の主張によれば、本件発明が文書ファイルに含まれる自ノード変数データの値を書き換えない態様を含むものであるにもかかわらず、サポート要件違反であるとの原告の主張について判断を示しておらず、不当である。

ウ 本件明細書は、当業者が本件発明 1 により、その課題を解決できるものと認識できる程度に記載されたものとはいえず、本件発明 1 はサポート要件に適合しない。

本件発明 1 はサポート要件に適合しないから、本件発明 1 4 もサポート要件に適合しない。

したがって、これと異なる本件審決の判断は誤りである。

(2) 被告の主張

ア 原告の主張は、本件明細書において、文書ファイルに含まれる自ノード変数データの値を書き換えることなく、どのようにして情報の更新を行うのかが明らかでないというものであるが、前記 3 (2) で主張したとおり、「評価」ボタンの押下による更新が文書ファイルの書換えを含まないことは、

本件明細書の記載から明らかであるから、原告の主張は前提を欠く。

イ 以上のとおり、本件発明 1 は、サポート要件に適合するから、本件発明 1 がサポート要件に適合することを前提に、本件発明 1 4 がサポート要件に適合するとした本件審決の判断に誤りがあるとの原告の主張も理由がない。

5

したがって、原告主張の取消事由 4 は理由がない。

第 4 当裁判所の判断

1 明細書の記載事項について

(1) 本件明細書（甲 1 8）の発明の詳細な説明には、別紙 1 の記載がある。

10 (2) 前記(1)の記載事項によれば、本件明細書には、本件発明に関し、次のような開示があることが認められる。

ア コンピュータを用いて各種情報の管理を行う場合、それぞれの情報を記憶したファイルを、所定のフォルダに保管することによって行うのが一般的であるが、作成したフォルダの構造及びそれぞれのフォルダに保管するファイルの種類等は、任意であってフォルダの作成者に依存するため、作成者以外の者が必要な情報に適確にアクセスすることは、必ずしも簡単ではない（【0002】）。このような問題に対処するため、案件（プロジェクト）毎にツリーを作成して表示し、作成した文書ファイルを、表示されたツリーの任意のノードに付随させて、サーバコンピュータに保管する文書情報管理システムや、異なる計算機やアプリケーションで共通に取扱うことができる XML のような構造化文書規格を用いた構造化文書を木構造として捉えて処理する構造化文書処理システムも存在したが、木構造のノードに含まれる情報は、相互に関連するものが多いにもかかわらず、これらの管理システムにおいては、それぞれの文書の該当する部分を個別に更新する必要があるが、十分効率的とはいえなかった（【0003】ないし【0005】）。

25

イ 「本発明」は、上記事情に鑑み、管理すべき情報の更新を、簡単かつ効率的に行うことができる情報管理方法を提供することを目的とし、その課題を解決する手段として、コンピュータが情報を管理する情報管理方法であって、前記コンピュータに複数のノードそれぞれに対応付けて入力された管理すべき情報を、前記ノードを識別するノード識別情報に対応付けられた複数のノードデータを含む文書ファイルとして前記コンピュータが記憶する情報記憶ステップと、前記情報記憶ステップで記憶された前記文書ファイルの情報を前記コンピュータが表示する情報表示ステップと、前記ノードデータに含まれるスクリプトを前記コンピュータが実行する情報評価ステップとを備え、前記ノードデータは、ルートノードを除いて、当該ノードの親ノードを特定する親ノード識別情報を含んでおり、前記スクリプトは、当該ノードデータに含まれる変数データである自ノード変数データと、当該ノードの直系上位ノードのノードデータに含まれる変数データである上位ノード変数データを利用した演算を行って、前記自ノード変数データの値を求める代入用スクリプトを含んでおり、前記情報表示ステップは、前記親ノード識別情報を利用して、前記ノードの木構造を表示する木構造表示ステップと、前記表示された木構造のノードのうちの選択されたノードの前記自ノード変数データ、前記上位ノード変数データ及び前記スクリプトを表示するノードデータテーブル表示ステップを含み、前記情報評価ステップは、前記代入用スクリプトの実行により、前記自ノード変数データの値を更新するステップを含むという構成を採用した（【0007】、【0008】）。

ウ 「本発明」においては、利用者が入力したデータに含まれるスクリプトを利用して、ノードデータを更新することができるので、管理すべき情報の更新を、簡単かつ効率的に行うことができ、複数のノードデータを含む1つの文書データを用いて、個々の業務や案件に関する情報を管理してい

るので、多数の利用者による情報の共有化、再利用を、簡単かつ効率的に行うことができるとともに、文書データに基づいて、簡単にノードの木構造を表示させることができるため、業務や案件全体の把握を簡単に行うことができ、さらに、表示された木構造の個々のノードに対応付けられた詳細情報を簡単に表示することができる（【0009】）ため、「本発明」
5 によれば、管理すべき情報の更新を、簡単かつ効率的に行うことができるという効果を奏する（【0024】）。

2 取消事由1（新規性の判断の誤り）について

(1) 甲1の記載事項について

10 ア 甲1には、別紙2の記載がある。

イ 前記アの記載事項によれば、甲1には、次のような開示があることが認められる。

(ア) 知識ベースを構築する際には、知識記述言語についての知識や経験が必要であり、従来は、知識ベース(1)に接続した知識ベース構築装置(13)は、知識記述言語についての知識や経験を有するシステム開発者(14)が操作する運用形態を採っていた。システム開発者(14)は、知識ベース構築に際して知識所有者(15)との口頭でのインタビューを行なって、知識所有者(15)から知識を獲得し、これに基づいて資料を作成し、知識を体系化した上、該資料に基づいて、知識ベース構築装置(13)を操
15 作し、知識ベース(1)を構築していた。しかし、インタビューが口頭で行われるため、知識ベース(1)を一旦構築した後も、知識の追加や修正が必要となり、また、資料作成のために多大な労力と時間を要し、その上、知識所有者(15)の知識が知識体系化の際に作成された資料と知識ベース(1)内に記述された知識とによって保存され、管理されているから、
20 知識を追加し、或いは修正する必要がある場合には、システム開発者(14)が再度、知識所有者(15)とのインタビューを行なって、知識の獲得、

体系化，及び知識ベース構築のプロセスを繰り返さねばならず，一連の処理が極めて煩雑となる問題があった（【0002】ないし【0006】）。

5 (イ) 「本発明」は，知識ベースと知識ベース構築者の間に，知識所有者自身が操作可能なユーザインターフェースを設けることによって，知識ベース構築者自身が知識ベースを容易に構築することが出来る知識ベース構築システムを提供することを目的とし，課題を解決するための手段として，複数のノードを互いに関係づけてなる階層構造を有して各ノードには詳細な知識が定義されるデータ構造のデータベース(4)と，該データベース(4)内のデータを前記知識記述言語に変換して知識ベース(1)に記述するデータ変換装置(3)と，データベース(4)内に知識の階層構造を定義するための階層構造定義装置(5)と，データベース(4)内に定義された階層構造の各ノードに対して詳細な知識を定義するための詳細知識定義装置(6)とを具えた構成を採り，これにより，知識所有者自身が知識ベース構築者となって操作することができ，従来の知識ベース構築者による知識獲得プロセスは不要となり，知識の体系化は自動的に行なわれ，その結果が全てデータベースに登録されるため，知識の統一的な保存，管理が可能となり，さらに，知識ベースの知識を変更する必要が生じた場合にも，データベースを修正する処理だけで済むという効果を奏する（【0007】，【0008】，【0012】）。

10
15
20 (2) 相違点2の認定の誤りについて

ア 原告は，甲1発明にはスクリプトが含まれているとして，本件審決が相違点2を実質的なものと認定したのを誤りと主張する。

(ア) 甲1発明の実施例では，製品設計に関する知識ベースを作成する場合，先ず階層構造定義装置(5)の操作によって，製品を構成する複数の部品のそれぞれにノード名称を付し，これをツリー上に連結して親子関係を定義する（【0015】ないし【0017】）。

次に、部品構成ツリー上の各ノード、すなわち部品や部位についての知識が詳細知識定義装置(6)によって定義される。この作業は階層構造の下位から上位へ向かってノード毎に行われ、各ノード毎に図3に示す処理が実行される。まず、1つのノードが選択され(S3)、選択されたノードについて、必要な属性の定義(S4。例えば部品の高さ、幅、長さ、色、材質等の属性名の定義)、属性の分類(S5。自己の親ノードからその属性値を入力すべき属性である入力属性(部品の高さ、幅、長さ)と、自己のノードで属性値を決定すべき決定属性(色、材質等)への分類)及び属性値の定義(S6。決定属性への属性値(黒、スチール等)の入力)が行われる(【0019】ないし【0022】)。この属性値の入力について、「数値に限らず、数式や条件分岐が入力可能である。」と明記されている(【0022】)ところ、ここでいう数式や条件分岐は、記述に基づいてコンピュータが処理を実行するためのものであり、スクリプトの定義である「コンピューターで、一連の処理手順を記述した簡易プログラム」(甲14)に当てはまり得る。

(イ) しかしながら、本件発明1の構成要件Fは、「前記スクリプトは、当該ノードデータに含まれる変数データである自ノード変数データと、当該ノードの直系上位ノードのノードデータに含まれる変数データである上位ノード変数データを利用した演算を行って、前記自ノード変数データの値を求める代入用スクリプトを含んでおり、」というものであるところ、甲1発明に「スクリプト」が開示されているとしても、これに「代入用スクリプト」が含まれているかが問題となる。

a 甲1発明では、数式や条件分岐が含まれると明示されているのは、自己のノードで属性値を決定すべき「決定属性」に分類された属性に対する属性値の入力である「属性値の定義」のみである(【0022】)。

甲1の【図6】の「引き出し」ノードの「決定属性」中の「内箱の高

さ」欄の「自己の高さー外箱の肉厚×2」との表記がこれに当たるが、これは自ノード変数データと上位ノードの変数データを利用した演算ではない。

5 甲1発明では、入力属性は、自己の親ノードからその属性値をそのまま入力すべき属性であり、例えば甲1の【図6】では、「内箱本体」ノードの入力属性欄の変数「高さ」に対し、上位ノード「内箱」の変数「内箱本体の高さ」の値を入力しているが、これは入力がされているだけで、これに加えて、自ノード変数データと上位ノードの変数データを利用した演算を行っているものではない。

10 そして、これ以外に、甲1において、自ノード変数データと、当該ノードの直系上位ノードのノードデータに含まれる変数データである上位ノード変数データを利用した演算を行っていることの開示はない。

b 次に、原告は、甲1には、例えばコンピュータが、自ノードである「内箱本体」の入力属性である「高さ」を意味する属性値の名称である「内箱本体の高さ」に対し、親ノードである「内箱」の入力属性である「高さ」を意味する属性値の名称である内箱の「自己の高さ」の値を代入する態様が記載されているとし、具体的には、甲1の【図6】の「内箱」にある

内箱本体の高さ	自己の高さ
---------	-------

との記載がそれに該当すると主張する。しかし、甲1発明については、部品構成ツリー上の各ノードを定義する作業は階層構造の下位から上位へ向かってノード毎に行われるとされているのであり(【0019】)、上記記載の自ノードは「内箱本体」でなく「内箱」であり、内箱ノードについての自ノード変数データである「自己の高さ」の値を、同じく内箱ノードについての自ノード変数である「内箱本体の高さ」に代入しているというべきである。上記記載が、自己のノードにて属

性値を決定すべき（【0021】）「決定属性」の欄にあることもこれを裏付ける。換言すれば、同記載は、「内箱」ノードに対する「引き出し」ノードのような親（上位）ノードで定義された属性の属性値、すなわち上位ノード変数データを利用しておらず、「当該ノードデータに含まれる変数データである自ノード変数データと、当該ノードの直系上位ノードのノードデータに含まれる変数データである上位ノード変数データを利用した演算を行って、前記自ノード変数データの値を求める代入用スクリプトを含んでおり」という本件発明の構成要件Fと一致するとはいえない。

c さらに、原告は、本件発明1では、代入用スクリプトも自ノード変数データも、共に複数のノードデータのいずれかに含まれていれば足りるものであり、両者が同じノードに含まれていることは、一切規定されていないので、甲1発明で、本件記載が、自ノード変数データ「高さ」が含まれている自ノード「内箱本体」ではなく、直系上位ノード「内箱」に記載されていることは、本件発明と甲1発明との相違点となることはないと主張する。

しかしながら、構成要件Fに記載された「当該ノードデータ」を、原告が主張するように文書ファイルに含まれる「複数のノードデータ」と解釈した場合、「複数のノードデータ」に対する「当該ノードの直系上位ノードのノードデータ」がどのノードデータを意味するのか特定できなくなり、「当該ノードデータに含まれる変数データである自ノード変数データ」及び「当該ノードの直系上位ノードのノードデータに含まれる変数データである直系上位ノード変数データ」が特定できなくなることによって構成要件F「代入用スクリプト」が特定できなくなるから、原告が主張するように解釈する余地はない。また、特許請求の範囲の記載を素直に読めば、本件発明1の構成要件Fの「当

該ノードデータ」とは、ある代入用スクリプトが値を求める変数データ（自ノード変数データ）を含むノードデータであって、特定の一つのノードデータを意味することは明らかというべきである。さらに、本件明細書の【図9】でも、「MW70 巾木（表）」とのラベルが付いたノードのノードデータとして、「スライス数=同一面数；」との代入用スクリプト、及び上記代入用スクリプトが値を求める変数である「スライス数」が記載されている。そうすると、上記代入用スクリプトが値を求める「スライス数」を含む「当該ノードデータ」が、「MW70 巾木（表）」とのラベルが付いたノードのノードデータに該当することは明らかであるから、本件明細書及び図面に基づいても、特定の一つのノードデータを意味する構成要件Fの「当該ノードデータ」を、原告が主張する「複数のノードデータ」と解釈することはできない。

なお、原告は、仮に、甲1発明において、代入用スクリプトと自ノード変数データとが同じノードに含まれることが要件であるとしても、甲1には代入用スクリプトの開示があるとし、甲1の【図6】における自ノード「内箱本体」には、結局のところ、

「高さ」＝（「内箱本体の高さ」＝「自己の高さ」）；
という、親ノード変数データの値「内箱本体の高さ」を、自ノード変数データ「高さ」に代入するための、代入用スクリプトが含まれていると主張する。

しかし、そもそも甲1の【図6】においては、「内箱本体」ノードにおける「高さ」は「入力属性」とされており、「決定属性」とはされていないのであるから、代入用スクリプトが含まれているものとは解し難いものであるし、「内箱」ノードと「内箱本体」ノードが実質的には同位で一体性の高いものであることを前提にその中に代入用スクリプトが含まれていると解する場合には、前記bで示したとおり、

上記代入用スクリプトが上位ノード変数データを利用するものではないことが一段と明確になるものというべきであるから、いずれにしても原告の上記主張は結論を左右し得ない。

d 以上によれば、いずれの点からみても、原告の主張は採用し得ない。

5 (ウ) まとめ

そうすると、「甲1発明では、ノードデータに当該ノードデータに含まれる変数データである自ノード変数データと、当該ノードの直系上位ノードのノードデータに含まれる変数データである上位ノード変数データを利用した演算を行って、前記自ノード変数データの値を求める代入スクリプトが含まれることは特定されておらず、更に当該スクリプトを表示することも特定されていない点。」が、本件発明1と甲1発明の相違点となり、これは実質的なものであるから、結局本件発明1と甲1発明は、相違点2に対応する相違点を有することになる。

そして、本件発明14は、本件発明1を引用し、プログラムの発明として特定したものであり、本件発明1と実質的に同じ構成を有するから、本件発明1と同様、甲1発明と実質的な相違点を有することになる。

(3) 相違点3の認定の誤りについて

ア 原告は、甲1には、スクリプトの開示があり、また、文書ファイルに記載された自ノード属性値をスクリプトにより変更することも開示されているから、相違点3を実質的な相違点と認定した本件審決の判断は誤りであると主張する。

イ しかし、甲1発明には代入用スクリプトの開示がないことは上記(2)のとおりであるし、甲1発明は、「「実行ステップ」において「ノードデータに含まれる属性値の数式の計算」が行われ、計算されたノードデータに基づき「知識ベース」の更新が行われるものの、「情報評価ステップ」で「代入用スクリプト」の実行によって「自ノード変数データの更新」を行

うものではない」から、いずれにしても原告の主張は採用し得ず、本件審決の相違点3の認定に誤りはなく、かつ、相違点3は実質的なものである。

そして、本件発明14は、本件発明1を引用し、プログラムの発明として特定したものであり、本件発明1と実質的に同じ構成を有するから、本件発明1と同様、甲1発明と実質的な相違点を有することになる。

(4) 小括

そうすると、本件発明1と甲1発明には相違点2及び相違点3に相当する相違点があり、かつ、これらは実質的なものである。

したがって、本件発明1は甲1に記載された発明ではなく、本件発明1が新規性を欠くということとはできない。

同様に、本件発明1の発明特定事項を全て含む本件発明14は、甲1に記載された発明ではなく、本件発明14が新規性を欠くということもできない。

よって、本件審決の判断に誤りはない。

3 取消事由2（甲1発明を主引用例とする進歩性判断の誤り）について

(1) 原告は、本件発明1において、代入用スクリプトと自ノード変数データとが、同一のノードデータに含まれるのに対し、甲1発明ではそうではないことが相違点に当たるとした場合であっても、甲1発明において、直系上位ノードに含まれている代入用スクリプトを、直系上位ノードではなく、自ノードに含ませることとするのは、当業者の技術常識ないし周知技術に基づく設計事項であるから、当業者は、相違点2及び相違点3に係る構成を容易に想到することができる」と主張する。

しかし、前記2において判示したとおり、甲1には、本件発明1の構成要件Fの代入用スクリプトに相当する事項自体が開示されていないから、原告が主張するように、単に代入用スクリプトを自ノードに含むか含まないかという点のみが相違点となるのではない。

そして、甲1には、ノードデータに当該ノードデータに含まれる変数デー

タである自ノード変数データと、当該ノードの直系上位ノードのノードデータに含まれる変数データである上位ノード変数データを利用した演算を行って、前記自ノード変数データの値を求める代入用スクリプトが含まれるようにする方法について記載も示唆もない。

5 したがって、その他の点について判断するまでもなく、当業者が、甲1発
明において、「スクリプトは「当該ノードデータに含まれる変数データであ
る自ノード変数データと、当該ノードの直系上位ノードのノードデータに含
まれる変数データである上位ノード変数データを利用した演算を行って、前
記自ノード変数データの値を求める代入用スクリプト」を含む」という相違
10 点2に係る本件発明1の構成及び「前記代入用スクリプトの実行により、前
記自ノード変数データの値を更新する」という相違点3に係る本件発明1の
構成を容易に想到することができたとはいえない。

同様に、当業者が、甲1発明において、本件発明1の発明特定事項を全て
含む本件発明14について、前記相違点2に係る本件発明14の構成及び前
15 記相違点3に係る本件発明14の構成を容易に想到することができたとはい
えない。

(2) 小括

以上によれば、相違点2及び相違点3についての容易想到性を否定した本
件審決の判断に誤りはないから、原告主張の取消事由2は理由がない。

20 4 取消事由3（実施可能要件違反）について

(1) 原告は、本件発明1（構成要件H）に、文書ファイルに含まれる自ノード
変数データの値を書き換えない態様が含まれることを前提に、本件明細書に、
自ノード変数データの値を書き換えない態様の開示がないとして、実施可能
要件違反を主張する。

25 ア 本件発明1の実施例では、表示画面（【図6】）がツリービューアとデ
ザインテーブルから構成される（【0044】）。デザインテーブルは、

ツリービューアに表示されたノードのうち選択されたノードが有する情報を表示する領域であり、このうち操作ボタン表示領域20 aには、デザインテーブルに関する各種操作を行うための操作ボタンが表示されている（【0046】）。本件明細書の【図6】をみると、表示ボタンには、「更新」ボタンと「評価」ボタンが含まれ、20 aに示された10個の長方形のうち、左上が「更新」ボタンであり、上段の右から2つ目が「評価」ボタンである。

本件明細書の【0058】には、「デザインテーブル20の編集内容は、領域20 aの更新ボタンを押すことによって、文書情報に反映される。」と記載されている。このように更新ボタンを押すのが、自ノード変数データを書き換える態様に当たると解される。

一方、代入用スクリプト及び生成用スクリプトは、操作ボタン表示領域20 aの「評価」ボタンを押すことにより実行される（【0067】）。

代入用スクリプト及び生成用スクリプトの具体例をみると、本件明細書の【図9】に示すように、ターゲットノードは、要計算の公開変数として、「スライス数」と「色」を有しており、代入用スクリプトとして、「スライス数=同一面数;」と「色= 同一面数」を有している。評価前は、同【図10】に示すように、公開変数「スライス数」と「色」の値は空欄となっている（【0072】）。

この状態で、このノードを選択し、評価ボタンを押し、評価条件として代入用スクリプトの実行を設定すると、記憶された代入用スクリプトを実行する。したがって、公開変数「スライス数」の値は、上位ノードの公開変数である「同一面数」の値「1」となり、公開変数「色」の値は、同様に上位ノードの公開変数である「巾木色」の値「F-205」となる。代入用スクリプト実行後のデザインテーブルの公開変数表示領域21の表示例が、同【図13】である（【0073】）。

すなわち、本件明細書には、ノードを選択し、評価ボタンを押すことにより代入用スクリプトが実行され、デザインエリア20の公開変数表示領域21の表示が【図10】から【図13】に変化することが開示されているが、これは、更新ボタンを押さないと文書情報に反映されないので（【0058】）、これが、自ノード変数データの値を書き換えない態様に当たると解される。

ウ 本件明細書には、文書情報に反映せずに代入用スクリプトを実行し、公開変数表示領域の表示を【図10】から【図13】のように変更する態様が明確に記載されており、当業者がこれを実装することに特に困難性はないから、本件明細書の発明の詳細な説明の記載は実施可能要件に適合するものと解される。

(2) 原告は、審判合議体は、本件発明が文書ファイルに含まれる自ノード変数データの値を書き換えない態様を含むものであるか、それとも含まないものであるかについて判断した上で、無効理由3についての結論を示すべきであったと主張するが、本件審決は、本件発明1及び14の構成要件AからHの全てについて、その方法を実施するための具体的な記載が発明の詳細な説明にあるか、そのような記載がない場合には、本件明細書及び図面の記載及び出願時の技術常識に基づき、当業者が過度の試行錯誤や複雑高度な実験等を行う必要なく、その方法を実施することができる程度にその発明が記載されているかを検討しており、別個の手続である侵害訴訟において被告が主張した具体的な態様が本件発明1及び14に属することを前提とする判断を示さなかったからといって不当とはいえないし、原告主張の前提において本件明細書を検討しても、実施可能要件に適合することは前記(1)のとおりである。

(3) 小括

以上によれば、本件発明1は実施可能要件に適合するものと認められ、本件発明14も同様であるから、原告主張の取消事由3は理由がない。

5 取消事由 4 (サポート要件違反)

(1) 原告は、本件発明 1 は、被告の主張によれば、文書ファイルの内容を変更しない態様を含むものであるところ、本件明細書を参酌しても、文書ファイルに含まれる自ノード変数データの値を書き換えることなく、どのようにして情報の更新を行うのか、どのように多数の利用者間で更新した情報を共有し、再利用することができるのか全く明らかでないとしてサポート要件違反を主張する。

しかし、前記 4 (1) のとおり、本件明細書には、文書情報に反映せずに代入用スクリプトを実行し、公開変数表示領域の表示を【図 1 0】から【図 1 3】のように変更する態様が明確に記載されており、当業者が本件発明 1 により、その課題を解決できるものと認識できるから、本件発明 1 はサポート要件に適合する。この点に関するその他の原告の主張については、原告は、別個の手続である侵害訴訟において被告が主張した具体的な態様が本件発明 1 及び 1 4 に属することを前提とする判断を示さなかった点が不当である旨主張するが、この主張が理由がないことは、前記 4 (2) において説示したところからも明らかである。

(2) 小括

以上によれば、本件発明 1 は、サポート要件に適合するものと認められ、本件発明 1 4 も、これと同様であるから、原告主張の取消事由 4 は理由がない。

6 結論

以上のとおり、原告主張の取消事由はいずれも理由がないから、本件審決を取り消すべき違法は認められない。

したがって、原告の請求を棄却することとして、主文のとおり判決する。

知的財産高等裁判所第4部

5

裁判長裁判官

菅 野 雅 之

10

裁判官

本 吉 弘 行

15

裁判官

岡 山 忠 広

20

(別紙1)

1 【技術分野】

【0001】

5 本発明は、コンピュータを用いて情報を管理する情報管理方法、情報管理プログラム、及び情報管理装置に関する。

2 【背景技術】

【0002】

10 コンピュータを用いて各種情報の管理を行う場合、それぞれの情報を記憶したファイル（文書ファイル、画像ファイル等）を、所定のフォルダに保管することによって行うのが一般的である。しかし、作成したフォルダの構造及びそれぞれのフォルダに保管するファイルの種類等は、任意であってフォルダの作成者に依存するため、作成者以外の者が必要な情報に適確にアクセスすることは、必ずしも簡単ではない。すなわち、多数の者が情報を共有化し、再利用できるように、情報管理を行うことは容易ではない。

15 【0003】

特許文献1には、情報の共有化、再利用を効率よく実現することができる文書情報管理システムが記載されている。この文書情報管理システムは、案件（プロジェクト）毎にツリーを作成して表示し、作成した文書ファイルを、表示されたツリーの任意のノードに付随させて、サーバコンピュータに保管するものである。

20 【0004】

また、異なる計算機やアプリケーションで共通に取扱うことができるデータ形式として、XML (Extensible Markup Language) 等の構造化文書規格があるが、特許文献2には、このような構造化文書を木構造として捉えて処理する構造化文書処理システムが記載されている。

25 【0005】

しかし、管理される各種情報の更新については、上記した管理システムにおい

ても、効率化が十分図られているとはいえない。すなわち、木構造のノードに含まれる情報は、相互に関連するものが多いが、上記した管理システムにおいては、それぞれの文書の該当する部分を個別に更新する必要があり、十分効率的とはいえない。

5 【0006】

 【特許文献1】特開2002-207772号公報

 【特許文献2】特開2003-178049号公報

 【発明の開示】

3 【発明が解決しようとする課題】

10 【0007】

 本発明は、上記事情に鑑みなされたもので、管理すべき情報の更新を、簡単かつ効率的に行うことができる情報管理情報を提供することを目的とする。

4 【課題を解決するための手段】

 【0008】

15 本発明の情報管理方法は、コンピュータが情報を管理する情報管理方法であつて、前記コンピュータに複数のノードそれぞれに対応付けて入力された管理すべき情報を、前記ノードを識別するノード識別情報に対応付けられた複数のノードデータを含む文書ファイルとして前記コンピュータが記憶する情報記憶ステップと、前記情報記憶ステップで記憶された前記文書ファイルの情報を前記コンピュータが表示する情報表示ステップと、前記ノードデータに含まれるスクリプトを前記コンピュータが実行する情報評価ステップとを備え、前記ノードデータは、
20 ルートノードを除いて、当該ノードの親ノードを特定する親ノード識別情報を含んでおり、前記スクリプトは、当該ノードデータに含まれる変数データである自ノード変数データと、当該ノードの直系上位ノードのノードデータに含まれる変数データである上位ノード変数データを利用した演算を行って、前記自ノード変数データの値を求める代入用スクリプトを含んでおり、前記情報表示ステップは、
25

前記親ノード識別情報を利用して、前記ノードの木構造を表示する木構造表示ステップと、前記表示された木構造のノードのうちの選択されたノードの前記自ノード変数データ、前記上位ノード変数データ及び前記スクリプトを表示するノードデータテーブル表示ステップを含み、前記情報評価ステップは、前記代入用スクリプトの実行により、前記自ノード変数データの値を更新するステップを含む
5 ものである。

【0009】

本発明によれば、利用者が入力したデータに含まれるスクリプトを利用して、ノードデータを更新することができるので、管理すべき情報の更新を、簡単かつ
10 効率的に行うことができる。また、複数のノードデータを含む1つの文書データを用いて、個々の業務や案件に関する情報を管理しているので、多数の利用者による情報の共有化、再利用を、簡単かつ効率的に行うことができるとともに、文書データに基づいて、簡単にノードの木構造を表示させることができ、業務や案件全体の把握を簡単に行うことができる。さらに、表示された木構造の個々のノ
15 ードに対応付けられた詳細情報を簡単に表示することができる。

【0010】

本発明の情報管理方法は、前記木構造表示ステップが、前記ノードを示すノードラベルと、前記親ノードの前記ノードラベルとの間を接続する階層リードとの表示を含むものを含む。本発明によれば、ノードの階層関係を容易に識別するこ
20 とができる。

【0011】

本発明の情報管理方法は、前記情報評価ステップが、前記表示された木構造のノードのうちの選択されたノードの前記代入用スクリプトを実行するに際して、前記選択されたノードの直系下位ノードの前記代入用スクリプトを合わせて実
25 行するものを含む。本発明によれば、関連するノードの評価を同時に行うことができる。また、上位ノードの変数値を変更した時の、下位ノードの変数値を簡単

に出力することができるので、シミュレーション動作を簡単に行うことができる。

【0015】

本発明の情報管理方法は、前記ノードデータテーブル表示ステップが、当該ノードの直系下位ノードの前記スクリプトでも利用される公開変数と、当該ノード
5 前記スクリプトでのみ利用される限定変数を含むものを含む。本発明によれば、必要な変数データのみを下位ノードに継承させることができる。

【0016】

本発明の情報管理方法は、前記ノードデータテーブル表示ステップが、前記公開変数の表示に際して、前記自ノード変数データと前記上位ノード変数データと
10 を区別して認識可能な態様で表示するものを含む。本発明によれば、自ノードの変数の入力及び確認、スクリプトの作成を容易に行うことができる。

【0017】

本発明の情報管理方法は、前記情報記憶ステップで記憶される文書が、前記コンピュータに入力された参照元ノードと参照先ノードとの関連を示す参照情報
15 を含んでおり、前記参照元ノードの前記代入用スクリプトは、前記自ノード変数データと前記上位ノード変数データに加えて、前記参照情報によって特定される前記参照先ノードの変数データを利用した演算を行って、前記自ノード変数データの値を求めるものであるものを含む。本発明によれば、参照先ノードの変数データを一次的な変数として利用することができ、代入用スクリプトの演算機能を
20 拡張することができる。

【0018】

本発明の情報管理方法は、前記木構造表示ステップが、前記参照元ノードと前記参照先ノードとを接続するリードを選択的に表示するものであるものを含む。
本発明によれば、参照先の有無、及び参照先ノードを簡単に確認することができる。
25

【0019】

本発明の情報管理方法は、前記ノードデータテーブル表示ステップが、前記参照先ノードの変数データを合わせて表示するものであるものを含む。本発明によれば、代入用スクリプトに利用される変数データの確認を確実に行うことができ、代入スクリプトの作成及び確認が容易になる。

5 **【0022】**

本発明の情報管理プログラムは、上記した情報管理方法における各ステップを、コンピュータに実行させるための情報管理プログラムである。

【0023】

10 本発明の情報管理装置は、上記した情報管理プログラムをインストールしたコンピュータを含むものである。

5 **【発明の効果】**

【0024】

以上の説明から明らかなように、本発明によれば、管理すべき情報の更新を、簡単かつ効率的に行うことができる。

15 6 **【発明を実施するための最良の形態】**

【0025】

以下、本発明の実施の形態について、図面を用いて説明する。

【0026】

20 本発明の情報管理方法及び情報管理装置は、所定の情報管理プログラムをインストールしたコンピュータを用いて実施する。ここで利用されるコンピュータは、スタンドアロン型のものでよいし、クライアント-サーバ型のものでよい。

【0027】

25 コンピュータにインストールする情報管理プログラムは、管理すべき情報をノードに対応付けて入力する情報入力ステップ、情報入力ステップで入力されたデータを、各ノードを識別するノード識別情報に対応付けられた複数のノードデータを含む文書として記憶する情報記憶ステップ、情報記憶ステップで記憶された

文書の情報を表示する情報表示ステップ，及びノードデータに含まれるスクリプトを実行する情報評価ステップを実行するためのプログラムを含んでいる。

【0028】

図1に，ノードデータとして記憶される情報の一例を示す。記憶される情報は，
5 ノード番号， ページ番号， 親ノード番号， ノードラベル， ノード表示属性情報，
変数情報， 代入用スクリプト， 生成用スクリプト， リンク情報を含む。

【0029】

ノード番号は，ノードを識別する情報であり，ノード生成時に自動的に一意の
番号が付与される。ページ番号は，文書に含まれるノードを複数の木構造として
10 表示するためのもので，そのノードが所属するページを識別する所属ページ番号
に，そのノードが別のページを形成する場合にそのページを識別する自己ページ
番号を含む。したがって，両方のページ番号が記憶されているノードは，2つの
ページに属することになる。親ノード番号は，そのノードの親ノードを識別する
番号であり，ノード生成時に親ノードを指定することにより，その指定された親
15 ノードのノード番号が自動的に記憶される。

【0030】

ノードラベルは，木構造表示時にそのノードを示す情報であり，ノード名称等
任意の情報を記憶することができる。ノード表示属性情報は，ノード表示時の背
景，文字の色，フォント等の文字属性，枠の形状，大きさ等の枠属性等を指定す
20 る情報である。ここで，JPG画像等をノードに表示したい場合は，その画像フ
ァイルの場所を指定するURL等が記憶される。

【0031】

変数情報は，各ノードが保持するデータであって，変数名に対応させて記憶さ
れる。記憶される変数は，下位ノードから参照される公開変数と，自ノード内
25 のみ使用する限定変数を含む。また，変数の値（「変数データ」と記述する場合
もある。）は，固定値が設定されても，スクリプトの実行によって演算された値

が設定されてもよい。また、URLが設定されてもよい。どのような値が設定されるかは任意である。

【0032】

代入用スクリプトは、自ノードの変数の値を演算するためのものである。代入用スクリプトは、自ノードの変数の値である自ノード変数データと、そのノードの直系上位ノードの公開変数の値である上位ノード変数データを利用して記述することが可能である。

【0033】

生成用スクリプトは、辞書に登録してある別のノードやノード群（木構造の複数ノード）を利用して、新規にそのノードの下位のノードを生成するものである。生成用スクリプトを条件文と併用することにより、代入用スクリプトの実行により求められた変数データの結果によって、子ノードや孫ノードを生成することができる。生成用スクリプトを利用することにより、例えば、部品管理を行う場合、親部品のサイズによって子部品が変わるケースの子部品のデータを簡単に生成することができる。

【0034】

なお、代入用スクリプト及び生成用スクリプトに使用する言語としては、スクリプト言語として使用されている任意の言語を使用することができる。

【0035】

リンク情報は、各ノードにリンクするファイルに関する情報である。スタンドアロン型のコンピュータで実施する場合、この情報はリンクファイルのインデックス情報である。また、クライアント-サーバ型のコンピュータで実施する場合、リンクファイルをサーバに転送後、インデックス情報を作成し、記憶する。リンク情報を記憶することにより、各ノードをフォルダとして利用することが可能となる。

【0036】

ノードデータは、例えばタグ付き文書情報として記憶される。図2に、ノードデータの一例を示す。図2のデータは、ルートノードのノードデータの例であり、ノード番号 (nodeNo) が「3450」、自己ページ番号 (ownPageNo) が「10」、ノードラベル (label) が「パッセル操作マニュアル」である。所属ページ番号を示す (belongPageNo) が「0」、親ノード番号を示す (parentNodeNo) が「0」である
5 ことで、ルートノードであることを示している。図2の「x =” 100”」から「color =” 0”」までは、ノードの表示位置等のノード表示属性情報である。

【0037】

この形式では、変数情報が、「<DataDivision>」と「</DataDivision>」の間に
10 挿入され、代入用スクリプトが、「<ProcesureDivision>」と「</Procesure Division>」の間に挿入され、生成用スクリプトが、「<GenerateDivision>」と「</GenerateDivision>」の間に、リンク情報が、「<LinkageDivision>」と「</Linkage Division>」の間に挿入される。ただし、図2の例では、変数情報、代入用スクリプト、生成用スクリプトは、記憶されていない。

【0038】

図3に、ノードデータの別の例を示す。図3のデータは、ルートノード以外の
15 ノードデータの例である。所属ページ番号が「3484」、親ノード番号が「3488」となっており、ルートノード以外のノードのノードデータであることが把握できる。また、自己ページ番号が「3526」となっていることから、別ページの木構造
20 の先頭ノードであることも把握できる。

【0039】

図4に、管理すべきデータを複数のノードデータを含む文書情報として記憶させたものの一例を示す。図4の文書は、ヘッダ部40、ノードデータ部41a～
41n、ライン部42、レポート部43を備える。

【0040】

ヘッダ部40は、管理される案件等を指すプロジェクトのプロジェクト番号、

名称（プロジェクト名）等を示す情報を含んでいる。図4の例では、プロジェクト名が「Manual ver2」，プロジェクト番号が「10」であることを示している。

【0041】

5 ノードデータ部41aは、ルートノードのノードデータを示し、ノードデータ部41b～41nは、ルートノード以外のノードのノードデータである。

【0042】

10 ライン部42は、ノード間を接続するリードを定義する情報が記憶される領域である。ノード間を接続するリードは、親子のノード間を接続する階層リードと、階層関係とは無関係に一時的に変数を参照する参照元ノードと参照先ノード間を接続する参照リードがあるが、ライン部42は、参照リードの存在及び、位置、表示属性等を規定する。

【0043】

15 次に、記憶された文書情報の表示について説明する。図5に、文書情報の表示を行う場合の概略動作フローを示し、図4に示す文書の表示を行った場合の表示画面の例を図6に示す。

【0044】

20 図6の表示画面は、ツリービューア10とデザインテーブル20を有する。ツリービューア10は、ノードの木構造を表示する領域であり、情報管理時の各種操作を行うためのプルダウンメニュー、及びポップアップメニューを表示する領域も兼ねる。ノードの木構造の表示は、ラベルとリードの表示によって行い、図6の例では、ルートノードのラベル表示11aとルートノード以外のノードのラベル表示11b、11c、11dと、それらの間を接続する階層リード12b、12c、12dが表示されている。

【0045】

25 ラベル表示11a～11dは、ノード表示属性情報に基づいて表示されるが、ルートノードのラベル表示11aには、ルートノードであることを示すマーク1

3 a が付加される。また、ルートノード以外のノードで自己ページ番号を有するノードは、その旨を示すマーク 1 4 b, 1 4 d が付加される。マーク 1 4 b, 1 4 d が付加されていることで、別ページの木構造が存在することを簡単に認識することができる。

5 **【0 0 4 6】**

 デザインテーブル 2 0 は、ツリービューア 1 0 に表示されたノードのうちの選択されたノードが有する情報を表示する領域であり、公開変数表示領域 2 1, 限定変数表示領域 2 2, 代入用スクリプト表示領域 2 3, 生成用スクリプト表示領域 2 4, 操作ボタン表示領域 2 1 a, 2 2 a, 2 0 a を有する。操作ボタン表示領域 2 1 a の各操作ボタンは、公開変数に対する各種操作を行うためのものであり、操作ボタン表示領域 2 2 a の各操作ボタンは、限定変数に対する各種操作を行うためのものである。また、操作ボタン表示領域 2 0 a の各操作ボタンは、デザインテーブル 2 0 に関する各種操作を行うためのものである。

【0 0 4 7】

15 プロジェクト毎に作成された文書ファイルを選択し、開くと、図 5 に示す手順で表示処理が行われる。ステップ S 1 0 1 では、文書データからルートノードを認識する。既述のように、ルートノードのノードデータは、所属ページ番号及び親ページ番号が「0」であるので、そのようなノードを探すことにより、ルートノードを認識することができる。なお、文書番号を一義的に割り当て、ルートノードの自己ページ番号を文書番号と一致させておくと、ルートノードの認識がさら

20 らに簡単になる。

【0 0 4 8】

 ステップ S 1 0 2 では、認識したルートノードの自己ノード番号を認識し、ステップ S 1 0 3 では、認識したページ番号のページに属するノードを認識する。すなわち、ルートノードの自己ページ番号を所属ページ番号として有するノード

25 を認識する。

【0049】

次いで、ステップS103で認識したノードのラベル表示を行う（ステップS104）。ラベル表示は、そのノードのノードラベル及びノード表示属性情報に基づいて表示する。そして、表示したノードの親子関係に基づいて階層リードを表示し、さらに、文書のライン部42の情報を参考にして、参照リードを表示する（ステップS105）。

【0050】

この状態では、ツリービューA10に木構造が表示された状態である。ステップS106では、ツリービューA10に表示されたノードが選択されたかどうかを判定し、選択されている場合、デザインテーブル20の表示を行い（ステップS107）、そのノードの変数、スクリプト等を表示する（ステップS108）。図6は、ルートノードを選択した場合の例であり、ルートノードには、変数情報等が記憶されていないので、デザインテーブル20に、データの表示はない。この状態で、別のノードを選択すると、そのノードの変数等が表示される。

【0051】

記憶された変数の値が、URL又はファイルパスである場合は、その変数を選択した状態で領域21a、22aの「実行」ボタンをおすことにより、そのURL又はファイルパスの内容を表示する。

【0052】

デザインテーブル20は、ノードを選択することで表示されるが、各ノードのリンク情報、及びノードレポートの表示は、ノード選択後メニューを表示させて行う。リンク情報表示が指示されると、そのノードのノードデータのリンク情報のリストを別ウィンドウで表示する。リンク情報が記憶されていない場合は、リストが空欄である。表示されたリストの中のファイルが選択されると、そのファイルの内容に応じた情報の表示を行う。ノードレポート表示が指示されると、別ウィンドウに表示するとともに、レポート領域43を参照し、該当ノードの情報

が記憶されている場合には、その情報を表示した別のウィンドウに表示する。

【0053】

ノードのリンク情報、及びノードレポートは、リンク情報のリスト、あるいはレポート表示ウィンドウを表示した状態で、追加、削除を行うことができる。

5 【0054】

図6に示した状態で、表示された木構造及びノードデータの編集が可能であり、編集操作に対応した表示を行う（ステップS109）。ツリービューア10上では、ノードの追加、削除、表示位置移動、表示属性変更、ノードラベル変更等を、プルダウンメニュー、ポップアップメニューの設定により行う。例えば、表示位置の変更は、ラベル表示をドラッグすることによって行い、ノードラベル及び表示属性の変更は、変更用のウィンドウを表示させて行う。また、ノードの削除は、削除したいノードを選択した状態で、メニューを表示させて削除する。ノードの追加は、メニューで追加モードに設定後、追加したいノードの親となるノードを選択し、そのままドラッグすることにより、新規ノードを生成する。また、ノードの繋ぎ換えは、繋ぎ換えたいノードを選択し、メニューを表示させて「ノード繋ぎ換え」を選択し、変更したい繋ぎ先のノード（親ノードとしたいノード）を選択することによって行う。生成されたノードのノードラベル、表示属性情報は、修正用のウィンドウを表示させて設定する。

20 【0055】

それぞれのノードに関するデータは、ノードデータとして1まとまりになっているので、これらの編集を行った場合でも、編集を行ったノードのノードデータに反映させるだけでよく、軽い処理負担で編集作業を行うことができる。

【0056】

デザインテーブルの情報を追加、修正する場合は、領域21a、22a、20aの該当するボタンを押すことによって行う。変数の追加は、領域21a又は22a「追加」ボタンを押して、新規の変数を生成し、変数名、値、修飾情報を

入力する。変数の入力は、別ウィンドウで入力フォームを表示させて行ってもよい。

【0057】

代入用スクリプト及び生成用スクリプトは、公開変数領域 2 1 および限定変数領域 2 2 に表示された変数を利用して作成する。公開変数領域 2 1 には、自ノードの公開変数だけでなく、直系上位のすべてのノードの公開変数が表示される。直系上位のノード以外のノードの変数を参照したい場合は、参照リードを生成して、参照先のノードと関連付けておく。代入用スクリプト及び生成用スクリプトは代入用スクリプト領域 2 3 及び生成用スクリプト領域 2 4 に直接入力してもよいし、別のウィンドウを開いて入力するようにしてもよい。

【0058】

なお、デザインテーブル 2 0 の編集内容は、領域 2 0 a の更新ボタンを押すことによって、文書情報に反映される。

【0059】

続いて、ステップ S 1 1 0 では、表示終了、すなわち文書クローズが指示されたかどうかを判定し、表示終了が支持されていない場合は、階層移動指示がされているか否かを判定する（ステップ S 1 1 1）。階層移動は、マーク 1 4 b, 1 4 d 等別の木構造の先頭ノードとなるノードを選択した状態で、ポップアップメニューから指示する。

【0060】

階層移動が指示されていると判定した場合、ステップ S 1 1 2 で別ページの木構造を表示する。木構造の表示は、表示すべきページ番号を認識後、ステップ S 1 0 3 ~ ステップ S 1 0 5 の処理と同様の処理を行う。木構造の先頭でないノードに対して階層移動が指示される（階層を降りる旨の指示）と、そのノードを先頭とするノードを表示する。その時のページ番号は、そのノードの自己ページ番号である。木構造の先頭ノードを選択して階層移動を指示すると、そのノードの

所属ページ番号が示すページの木構造を表示する。

【0061】

別ページの木構造についても、ノードの追加、削除は同様に行うことができる。また、特定のノードを先頭ノードとする新規の木構造を作成する場合は、先頭と
5 なるノードを選択し、メニューから別ページ生成を指示する。そして、別ページ
に表示された先頭ノードに、下位ノードを追加して、木構造とする。

【0062】

次に、新規プロジェクトの作成について、図7を用いて説明する。図示しない
プロジェクトツリーを表示させ、新規プロジェクトの作成が指示されると、新規
10 プロジェクト作成用のダイアログボックスを表示する（ステップS 201）。
次いで、ダイアログボックスを介して少なくとも新規プロジェクトのプロジェクト
名が入力されると（ステップS 202）、ツリービューア10にルートノード
のラベル表示を行う（ステップS 203）。次いで、ルートノードのノードデー
15 タを入力する（ステップS 204）。そして、ルートノードに下位ノードを順次
追加し、ノードデータを入力する（ステップS 205）。ノードデータの入力、
及びノードの追加は、図6で説明した編集処理と同様の手順で行う。

【0063】

続いて、ノードデータに含まれる変数情報、代入用スクリプト、生成用スクリ
プトについて、部品管理を例に説明する。図8は、部品管理に適用した場合の木
20 構造の一例を示す図である。図8の木構造は、製品番号「990070」、品名「MW
-70」の製品の部品管理に適用したもので、製品は、「fore」、「center」、
「bone」、「back」の部分からなり、さらにそれぞれの部分が、複数の部品から
構成されることを示している。ただし、図8では、部分「fore」の部品のみを記
載し、他の部分については、省略してある。

【0064】

図8から明らかなように、部分「fore」は、3つの「MW70 巾木(表)」と3つ

の「MW70 パネル(表)」から構成される。図9に、1つの部品「M W7 巾木(表)」に対応するノード（図8では、便宜的に右肩に「*」を付してある。）のノードデータの一部を示し、図10に、そのノードが選択された場合の公開変数表示領域21の表示例を、図11に、限定変数表示領域の表示例を示す。

5 【0065】

公開変数表示領域に表示される公開変数は、自ノードの公開変数51と、直系上位ノードの公開変数52を含み、直系上位のノードの公開変数52は、自ノードの公開変数51と異なる色で表示される（図10では、フォントを変えて示してある。）。また、公開変数には、固定値が入力される公開変数と、代入用スクリプトの実行によって計算される公開変数があり、修飾領域に「なし」あるいは「要計算」を表示することによりに区別される。

15 【0066】

要計算の公開変数の値は、後述するように代入用スクリプトが実行されるまでは空欄であり、図9及び図10は、代入用スクリプト実行前の状態を示している。なお、直系上位ノードに要計算の公開変数が含まれ、その公開変数の値が代入スクリプトの実行前で定まっていない場合、その公開変数は、下位ノードの公開変数領域21に表示されない。すなわち、他のノードからの参照が一時停止される。

20 【0067】

代入用スクリプト及び生成用スクリプトは、操作ボタン表示領域20aの「評価」ボタンを押すことにより実行される。図12に、スクリプト実行時の概略動作フローを示す。スクリプトを実行したいターゲットのノード選択し、「評価」ボタンが押されると、評価条件設定用のダイアログボックスを表示する（ステップS301）。このダイアログボックスでは、評価対象スクリプトの種類と、評価階層が少なくとも可能となっている。すなわち、代入用スクリプトのみの実行、生成用スクリプトの実行、両スクリプトの実行の設定と、自ノードのみを評価するか直系下位ノードの指定階層まで評価するかを設定する。

【0068】

ステップS302では、ダイアログボックスの設定内容にしたがって、評価条件を認識する。そして、代入用スクリプトの実行が設定されているかどうかを判断し（ステップS303）、設定されている場合は、代入用スクリプトを実行する（ステップS304）。

【0069】

代入用スクリプトの実行が終了した後、及び代入用スクリプトの実行が設定されていない場合は、生成用スクリプトの実行が設定されているかどうかを判断する（ステップS306）。そして、生成スクリプトの実行が設定されている場合は、その生成用スクリプトを実行する。

【0070】

生成用スクリプトの実行が終了した後、及び生成用スクリプトの実行が設定されていない場合は、ノードデータを参照して、直系下位のノードに評価対象ノードがあるかどうかを判断する（ステップS307）。ここで、判断対象となるノードには、生成用スクリプトの実行によって生成されたノードも含まれる。

【0071】

ステップS307で、スクリプトを実行すべきノードが残っていると判断された場合は、ステップS303に戻り、下位のノードについて同様の判断を行い、スクリプトの実行を行う。

【0072】

次に、代入用スクリプト及び生成用スクリプトの具体例を、図8の「*」を付したノードをターゲットノードとして説明する。図9に示すように、ターゲットノードは、要計算の公開変数として、「スライス数」と「色」を有しており、代入用スクリプトとして、「スライス数=同一面数;」と「色=同一面数」を有している。評価前は、図10に示すように、公開変数「スライス数」と「色」の値は空欄となっている。

【0073】

この状態で、このノードを選択し、「評価ボタン」を押し、評価条件として代入用スクリプトの実行を設定すると、記憶された代入用スクリプトを実行する。したがって、公開変数「スライス数」の値は、上位ノードの公開変数である「同一面数」の値「1」となり、公開変数「色」の値は、同様に上位ノードの公開変数である「巾木色」の値「F-205」となる。代入用スクリプト実行後のデザインテーブルの公開変数表示領域21の表示例を、図13に示す。

【図1】

ノード番号
ページ番号
親ノード番号
ノードラベル
ノード表示属性情報
変数情報
代入用スクリプト
生成用スクリプト
リンク情報

10

【図2】

```
<node nodeNo="3450" ownPageNo="10" belongingPageNo="0" parentNodeNo="0"
label="パズル操作マニュアル" x="100" y="100" width="124" height="18"
ownX="345" ownY="63" ownWidth="130" ownHeight="28" sideNo="0"
decorate="ncolor=-1;fcolor=-16777216;llcolor=-16777216;dir=0;f=10,0;resizable;
frameshape=1;icon=40;" line="message=";fcap=0;tcap=3;f=10,0;icolor=-16777216;
color=-4144960;lpos=0;" color="0">![CDATA[
<DataDivision>
</DataDivision>
<ProcedureDivision></ProcedureDivision>
<GenerateDivision></GenerateDivision>
<LinkageDivision>
</LinkageDivision>
</Node>
]]></node>
```

【 図 3 】

```

<node nodeNo="3526" ownPageNo="3526" belongingPageNo="3484" parentNodeNo=
"3488"label="判断文" x="311" y="223" width="44" height="18" ownX="94" ownY="228"
ownWidth="44" ownHeight="18" sideNo="0" decorate="ncolor=-3355393;fcolor=-16777216;
llcolor=-16777216;dir=0;f=10,0;frameshape=0;" line="message=;fcap=0;tcap=3;f=10,0;
lcolor=-16777216;color=-4144960;lpos=0;" color="0"><![CDATA[
<DataDivision>
</DataDivision>
<ProcedureDivision></ProcedureDivision>
<GenerateDivision></GenerateDivision>
<LinkageDivision>
</LinkageDivision>
</Node>
]]></node>

```

5

【 図 4 】

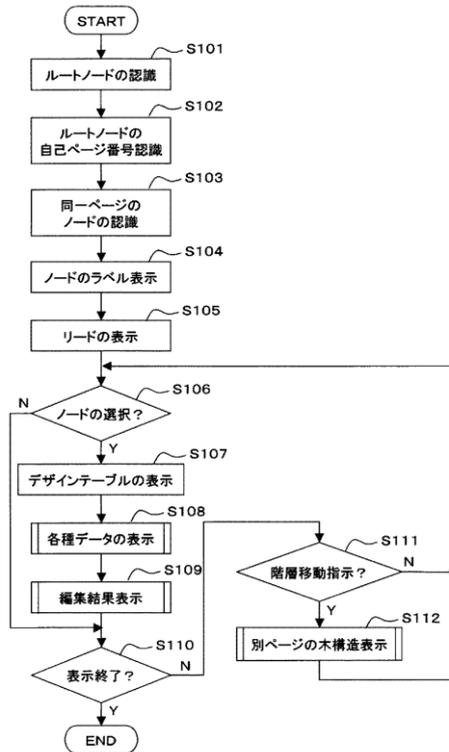
```

<project projectNo="10" name="Manual ver2" createDate="2004/05/01" note="" >40
<nodearea>
<node nodeNo="3450" ownPageNo="10" belongingPageNo="0" parentNodeNo="0"
label="バツセル操作マニュアル" **** >41a
****</node>
<node nodeNo="3451" ownPageNo="3451" belongingPageNo="10" parentNodeNo
="3450" label="PSL文書編集" **** >41b
****</node>
<node nodeNo="3479" ownPageNo="0" belongingPageNo="10" parentNodeNo=
"3451" label="バツセルデータはすべて構造化文書です&#x0A;ノードは1つの文
章の単位でもあります&#x0A;ここでは、スクリプトの書式や内部&#x0A;データの
使用方法を説明します" **** >41c
****</node>
<node nodeNo="3623" ownPageNo="3623" belongingPageNo="10" parentNodeNo
="3450" label="PSLオブジェクトの操作" **** >41d
****</node>
<node nodeNo="3623" ownPageNo="3623" belongingPageNo="10" parentNodeNo
="3450" label="PSLオブジェクトの操作" **** >41e
****</node>
<node nodeNo="3623" ownPageNo="3623" belongingPageNo="10" parentNodeNo
="3450" label="PSLオブジェクトの操作" **** >41n
****</node>
</nodearea>
<linearea>
<line lineNo="1" fromNodeNo="3502" toNodeNo="3503" pageNo="3489" decorate
="fcap=0;tcap=3;f=10,0; **** Target="" Parameter="" Message="" Options="" />
**** >42
****
****
</linearea>
<reportarea>
<report reportNo="2" nodeNo="3833" title="hello" contents="aaa" createDate=
"2004/06/23" /> >43
****
</reportarea>
</project>

```

10

【図5】

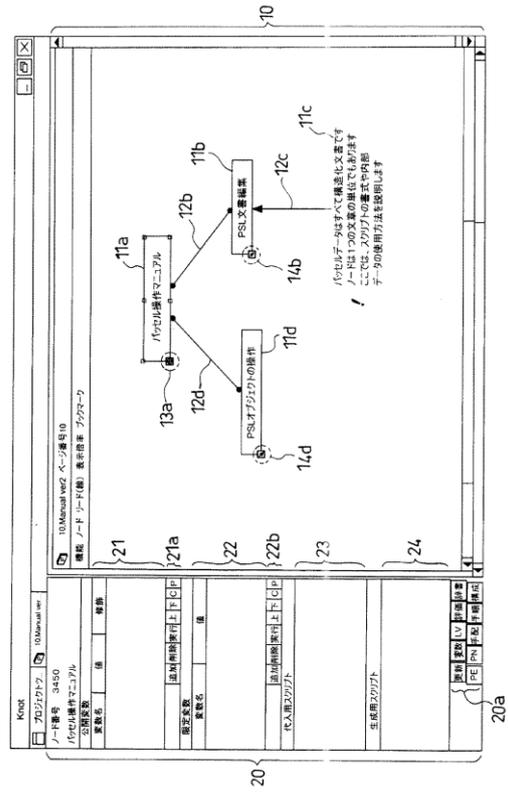


5

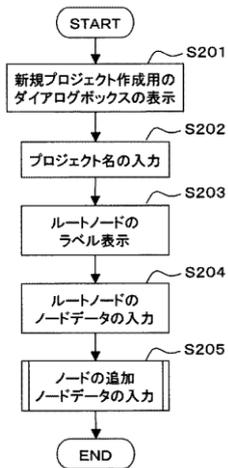
10

15

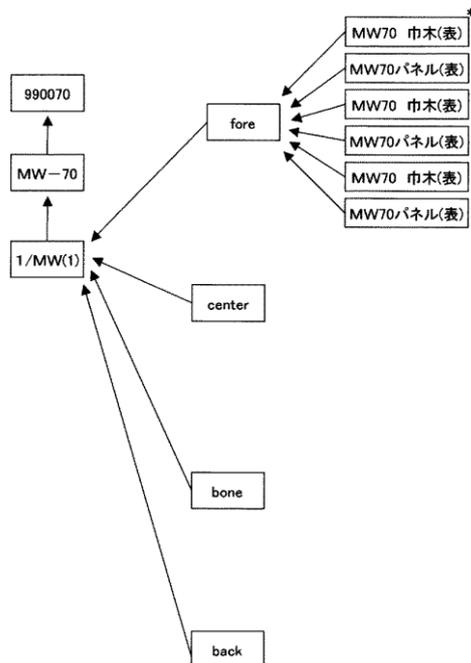
【図6】



【図7】



【図 8】



【図 9】

```

<node nodeNo="158643" ownPageNo="0" belongingPageNo="20" parentNodeNo="
123282" label="MW70 巾木(表)"...
...
<DataDivision>
<Variable name="_x" value="930.0" />
<Variable name="_y" value="0.0" />
<Variable name="_w" value="800.0" />
<Variable name="_h" value="75.0" />
...
<Variable name="スライス数" value="" needcalc="" />
<Variable name="色" value="" needcalc="" />
...
<Variable name="_unfixed" value="false" private="" />
<Variable name="_slicetype" value="巾木" private="" />
<Variable name="_sliceweight" value="0.0" private="" />
...
</DataDivision>
<ProcedureDivision>
スライス数=同一面数;
色=巾木色;
...
</ProcedureDivision>
<GenerateDivision>
if (_w>1400){
  if (巾木材質==&quot;スチール&quot;){
    use(&quot;C07000400000000004&quot;,1);
    use(&quot;C07000400000000005&quot;,1);
  } else if (巾木材質==&quot;ステン&quot;){
    use(&quot;C07000400000000006&quot;,1);
    use(&quot;C07000400000000007&quot;,1);
  }
} else {
  if (巾木材質==&quot;スチール&quot;){
    use(&quot;C07000400000000001&quot;,1); // 巾木(ST)
  } else if (巾木材質==&quot;ステン&quot;){
    use(&quot;C07000400000000002&quot;,1); // 巾木(SUS)
  }
}
...
</GenerateDivision>
</LinkageDivision>
</LinkageDivision>
</Node>
]}</node>
  
```

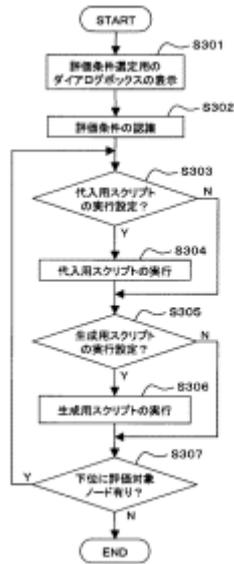
【図 1 0】

変数名	値	修飾
x	930.0	なし
_y	0.0	なし
_w	800.0	なし
_h	75.0	なし
...
スライス数		要計算
色		要計算
巾木材質	スチール	なし
...		
_layername	fore	なし
面NO	1	なし
面ID	MW	なし
同一面数	1	なし
...		
パネル色	KW-400	なし
巾木色	F-205	なし
...		
登録年月	200310	なし
製造番号	000000990070	なし
...		

【図 1 1】

変数名	値
...	
_unfixed" value	false
_slicetype	巾木
_sliceweight	0.0
...	

【図 1 2】



【図 1 3】

変数名	値	修飾
51 { _x	930.0	なし
_y	0.0	なし
_w	800.0	なし
_h	75.0	なし
...
スライス数	1	要計算
色	F-205	要計算
巾木材質	スチール	なし
...		
52 { _layername	fore	なし
面NO	1	なし
面ID	MW	なし
同一面数	1	なし
...		
パネル色	KW-400	なし
巾木色	F-205	なし
...		
登録年月	200310	なし
製造番号	000000990070	なし
...		

(別紙 2)

1 【0001】

【産業上の利用分野】本発明は、知識所有者自身が知識ベース構築者となって操作することが可能な知識ベース構築システムに関するものである。

5 2 【0002】

【従来の技術】知識ベースは、所定の知識記述言語で記述された一種のコンピュータプログラムであって、知識ベースを構築する際には、知識記述言語についての知識や経験が必要である。そこで、従来は図7に示す如く、知識ベース(1)に接続した知識ベース構築装置(13)は、知識記述言語についての知識や経験を有するシステム開発者(14)が操作する運用形態を採っている。

【0003】システム開発者(14)は、知識ベース構築に際して知識所有者(15)との口頭でのインタビューを行なって、知識所有者(15)から知識を獲得する。更にシステム開発者(14)は、獲得した知識に基づいて資料を作成し、知識を体系化する。そして、該資料に基づいて、知識ベース構築装置(13)を操作し、知識ベース(1)を構築するのである。

3 【0004】

【発明が解決しようとする課題】ところが、従来の知識ベース構築方法では、インタビューは口頭で行なわれるから、一度のインタビューでの完全な知識獲得は困難であり、特に大規模なシステムの構築においては、口頭のインタビューでは完全な知識獲得は不可能に近い。この結果、知識ベース(1)を一旦構築した後も、知識の追加や修正が必要となる。

【0005】又、知識を体系化するプロセスでは、資料作成のために多大な労力と時間を要する問題がある。

【0006】更に、知識所有者(15)の知識は、知識体系化の際に作成された資料と知識ベース(1)内に記述された知識とによって保存され、管理されているから、知識を追加し、或いは修正する必要がある場合には、システム開発者(14)が再

度、知識所有者(15)とのインタビューを行なって、知識の獲得、体系化、及び知識ベース構築のプロセスを繰り返さねばならず、一連の処理が極めて煩雑となる問題がある。

5 【0007】本発明の目的は、知識ベースと知識ベース構築者の間に、知識所有者自身が操作可能なユーザインターフェースを設けることによって、知識ベース構築者自身が知識ベースを容易に構築することが出来る知識ベース構築システムを提供することである。

4 【0008】

10 【課題を解決する為の手段】本発明に係る知識ベース構築システムは、複数のノードを互いに関係づけてなる階層構造を有して各ノードには詳細な知識が定義されるデータ構造のデータベース(4)と、該データベース(4)内のデータを前記知識記述言語に変換して知識ベース(1)に記述するデータ変換装置(3)と、データベース(4)内に知識の階層構造を定義するための階層構造定義装置(5)と、データベース(4)内に定義された階層構造の各ノードに対して詳細な知識を定義する
15 ための詳細知識定義装置(6)とを具えている。

5 【0009】

【作用】階層構造定義装置(5)及び詳細知識定義装置(6)は、例えばキーボード、マウス等の入力操作によってデータ入力が可能であり、階層構造定義装置(5)のデータ入力操作によって、データベース(4)内に知識の階層構造のみが定義され、
20 階層構造定義装置(5)のデータ入力操作によって、データベース(4)内に定義された階層構造の各ノードに対して詳細な知識が定義される。

【0010】この際、データベース(4)のデータ構造は、知識ベース(1)の知識記述言語に比べて、人間の言語に近い形で表現することが可能であるため、知識所有者自身が階層構造定義装置(5)及び詳細知識定義装置(6)を操作して、自己
25 の知識をデータベース(4)に容易に登録することが出来る。

【0011】データベース(4)内に階層構造にて登録された知識は、データ変換

装置(3)により知識ベース(1)の知識記述言語に変換され、知識ベース(1)に記述される。

【0012】

【発明の効果】本発明に係る知識ベース構築システムは、知識所有者自身が知識ベース構築者となって操作することが出来るから、従来の知識ベース構築者による知識獲得プロセスは不要となる。又、知識の体系化は自動的に行なわれ、その結果が全てデータベースに登録されるから、知識の統一的な保存、管理が可能である。従って、知識ベースの知識を変更する必要が生じた場合にも、データベースを修正する処理だけで済む。

10 6 【0013】

【実施例】以下、本発明の一実施例につき、図面に沿って詳述する。図1に示す如く知識ベース構築システム(2)は、知識ベース(1)と知識ベース構築者(8)の間に介在して、ユーザインターフェースの役割を果すものである。

【0014】知識ベース構築システム(2)は、複数のノードを互いに関係づけてなる階層構造を有して各ノードには詳細な知識が定義されるデータ構造のデータベース(4)と、該データベース(4)内のデータと前記知識記述言語の間の双方向の変換を行なうデータ変換装置(3)と、データベース(4)内に知識の階層構造を定義するための階層構造定義装置(5)と、データベース(4)内に定義された階層構造の各ノードに対して詳細な知識を定義するための詳細知識定義装置(6)と、詳細知識の定義の際に他のノード或いは他の知識ベースの知識を参照するための知識参照装置(7)とから構成され、階層構造定義装置(5)、詳細知識定義装置(6)及び知識参照装置(7)は夫々、知識ベース構築者(8)によるデータ入力操作が容易な構成となっている。

【0015】例えば、ある製品設計に関する知識ベースを作成する場合、先ず階層構造定義装置(5)の操作によって、その製品の部品構成ツリーが作成される。ここで、階層構造定義装置(5)による処理は、図2に示す如くノード名称の定義

(S 1)と階層構造の定義(S 2)から構成される。

【0016】即ち、ノード名称の定義においては、図示の如く製品を構成する複数の部品の夫々にノード名称(A~G)が付与される(S 1)。この段階では、各ノードは相互に関連なく独立しており、例えば図示するイメージにて画面表示される。

【0017】その後、階層構造の定義においては、例えば画面に表示された各ノードをマウスの操作等によってツリー状に連結することにより、図示の如く相互の親子関係が定義されることになる(S 2)。

【0018】この結果、図1のデータベース(4)内には、各ノードの親子関係、即ち自己の名称、自己の親ノードの名称及び子ノードの名称が記述される。

【0019】次に、部品構成ツリー上の各ノード、即ち部品や部位についての知識が詳細知識定義装置(6)によって定義される。この作業は階層構造の下位から上位へ向かってノード毎に行なわれ、各ノード毎に図3に示す処理が実行される。

【0020】先ず、1つのノードが選択され(S 3)、選択されたノードについて、必要な属性の定義(S 4)、属性の分類(S 5)、及び属性値の定義(S 6)が行なわれる。必要な属性の定義(S 4)の際には、部品の“高さ”、“幅”、“長さ”、“色”、“材質”等の属性名のみが定義され、これらの属性名が図示の如く各ノードに付加される。

【0021】属性の分類(S 5)においては、定義された属性の性質に応じて、入力属性と決定属性に分類される。ここで、入力属性は、自己の親ノードからその属性値を入力すべき属性(例えば“高さ”、“幅”、“長さ”)であり、決定属性は、自己のノードにて属性値を決定すべき属性(例えば“色”、“材質”)である。これによって、データベースには図示のイメージのデータ構造が形成される。

【0022】属性値の定義(S 6)では、決定属性に分類された属性に対して属性値(“黒”、“スチール”)の入力が行なわれる。ここで、属性値としては、数値に限らず、数式や条件分岐が入力可能である。

【0023】上記の処理(S3～S6)が最上位のノードに至るまで繰り返される。この結果、全てのノードについて、図3のステップS6に示す如きイメージのデータ構造が形成される。ここで、最上位のノードに入力属性として定義されている属性名が、知識ベース運用の際にデータを入力すべき項目であり、該データ入力によって知識ベースが動作を開始することになる。

【0024】上記の詳細知識定義装置(6)の操作の過程で、他の部品或いは他の製品の知識を参照する必要性が生じたときは、知識参照装置(7)を用いて、他のノード或いは他の知識ベースを参照することが可能である。詳細知識定義装置(6)は、他のノードの知識を検索してその結果を報知する処理と、他のノードの知識を提示し或いはその知識の流用を許可する処理の2つの機能を有している。

【0025】上記の処理によってデータベース(4)の登録が終了した後、その旨を知識ベース構築システム(2)にデータ入力すれば、データ変換装置(3)がデータベース(4)内のデータを自動的に所定の知識記述言語に変換して、知識ベース(1)に記述する。

【0026】その後、知識の変更に際しては、例えば、知識の階層構造を変更する場合には、階層構造定義装置(5)を用いてデータベース(4)の該当箇所を書き換えるだけで、知識ベース(1)の階層構造が変更されることになる。又、あるノードの知識を変更する場合は、詳細知識定義装置(6)を用いてデータベース(4)の該当箇所を書き換えるだけで、これが知識ベース(1)の知識に反映される。

【0027】例えば、図4に示す如く取っ手(12)を具えた内箱(11)と外箱(10)からなる引き出し(9)を設計する場合、階層構造定義装置(5)の操作によって、図5に示す5つのノード名称(“引き出し” “内箱” “外箱” “取っ手” “内箱本体”)が定義され、これらのノードに図示の如き階層構造が定義される。

【0028】その後、詳細知識定義装置(6)の操作によって、図6に示す如く各ノードに詳細知識が定義される。この作業は、下位のノードから上位のノードへ向かって進められる。例えば、“取っ手”は、予め規定された形状のものを用い

るので、全ての属性値を自己のノードにて決めることが出来る。従って、属性は決定属性のみから構成される。

【0029】一方、“内箱本体”は、その大きさが上位の部品の大きさから決定されるべきものであるから、“高さ”“幅”“長さ”は親ノードである“内箱”
5 に委ねられる。“外箱”についても同様である。

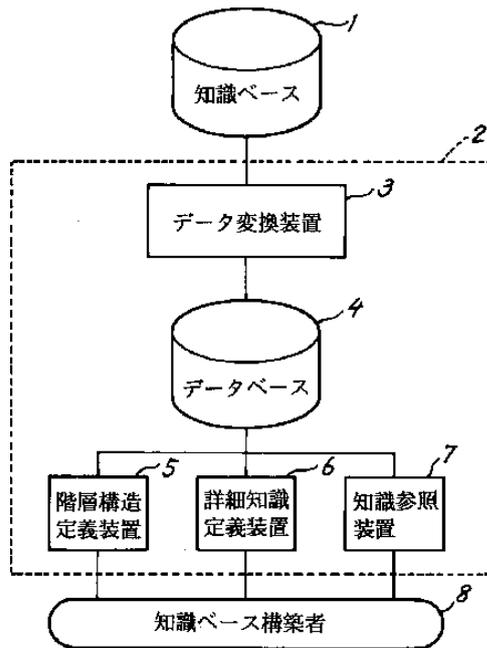
【0030】“内箱”のノードでは、“内箱本体”に入力すべき3つの属性(“高さ”“幅”“長さ”)を自己の大きさから決定する。又、“内箱”自身の大きさは、親ノードである“引き出し”にその決定を委ねる。

【0031】この様にして全てのデータ入力作業が終了すれば、データベース内
10 には、図6に示す如きイメージでデータ構造が登録され、最上位のノードである“引き出し”の“高さ”“幅”“長さ”が、知識ベースに対する入力データとなる。

【0032】上記知識ベース構築システムは、データベース(4)のデータ構造が知識ベースに比べて簡単であるため、知識記述言語についての知識や経験を有しない知識所有者であっても、階層構造定義装置(5)、詳細知識定義装置(6)及び
15 知識参照装置(7)を操作することによって、容易に知識ベース(1)を構築することが出来る。又、従来の如き知識獲得の作業が不要であるから、知識ベース構築のための処理が簡略化される。更に、知識の体系化と知識の記述の両プロセスが共に共通のデータベースを用いて行なわれるため、知識の追加や修正が従来に比
20 べて遥かに容易となる。

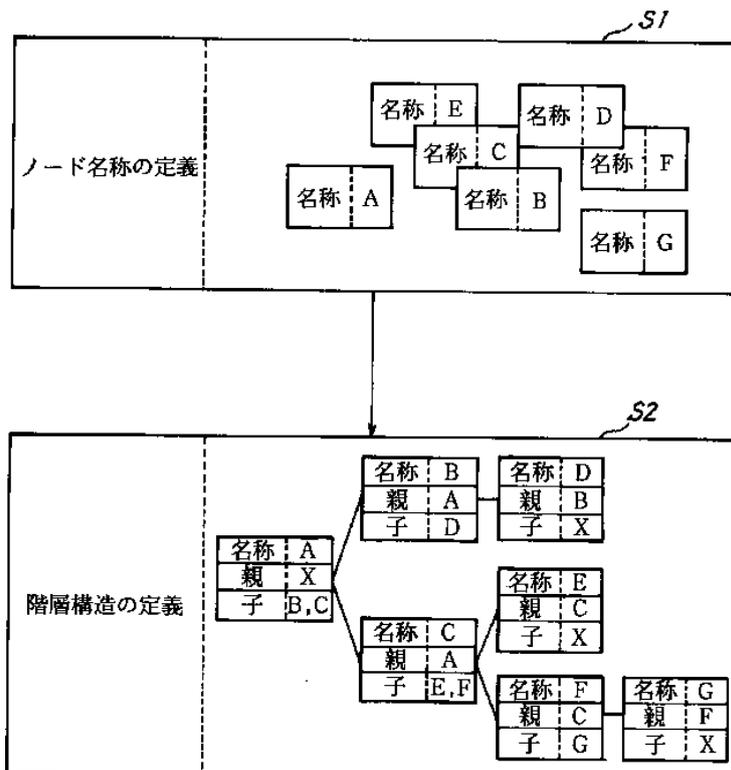
【0033】上記実施例の説明は、本発明を説明するためのものであって、特許請求の範囲に記載の発明を限定し、或は範囲を減縮する様に解すべきではない。又、本発明の各部構成は上記実施例に限らず、特許請求の範囲に記載の技術的範囲内で種々の変形が可能であることは勿論である。

【図1】

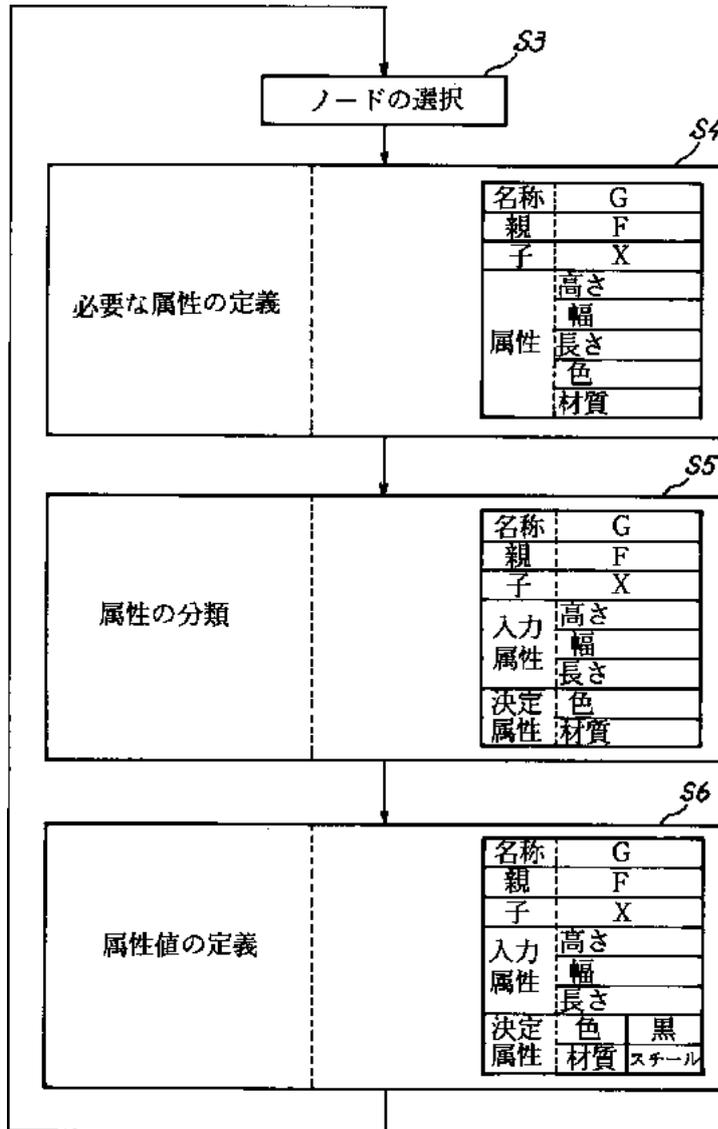


5

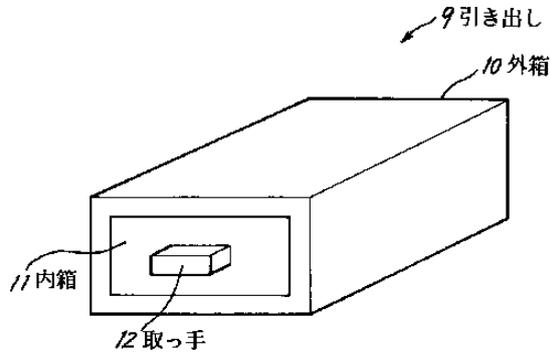
【図2】



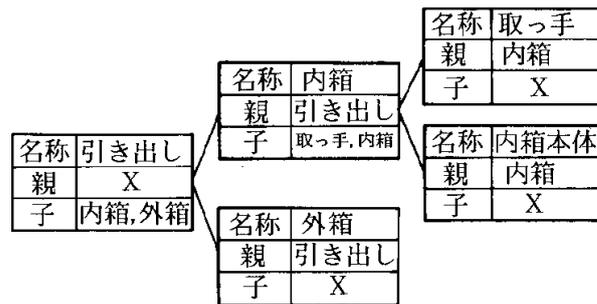
【図3】



【図4】



【図5】

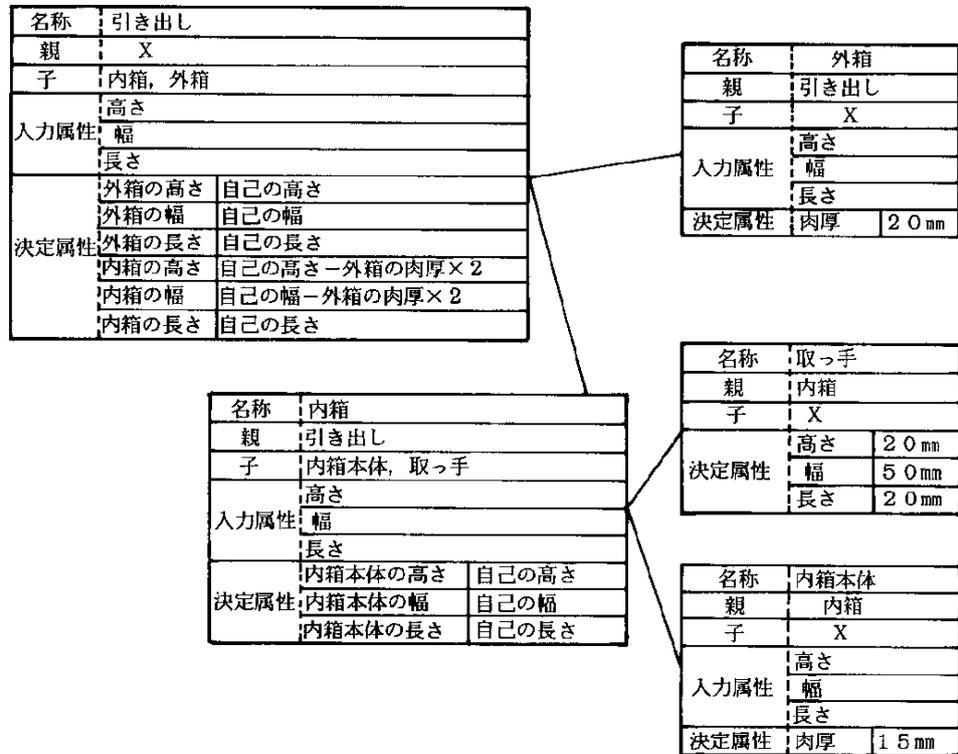


5

10

15

【図6】



【図7】

