

ソフトウェア関連発明特許に係る判例紹介  
～被告プログラムは原告特許の技術的範囲に属しないとされた裁判例～

平成29年（ワ）第31706号

原告：株式会社パッセルインテグレーション

被告：ソフトバンクロボティクス株式会社

2019年 5月21日

執筆者 弁理士 田中 伸次

## 1. 概要

本件は、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」とする特許第3754438号の特許権（以下「本件特許権」といい、この特許を「本件特許」という。また、本件特許の願書に添付した明細書及び図面を「本件明細書等」という。）を有する原告が、被告が「Choregraphe」という名称のプログラム（以下「被告プログラム」という。）を提供することは、本件特許権を侵害する旨を主張して、損害賠償金及び遅延損害金の支払いを求めたが、裁判所は、被告プログラムは本件特許の技術的範囲に属しないと判断した、裁判例である。

## 2. 特許請求の範囲の記載

### 1) 本件発明

本件発明は以下のとおりである。以下の記載は判決文からの引用である。

- 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、
- 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

I 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。

## 2) 本件発明の意義

本件発明は、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において、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る。

```

<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>

```

} スクリプト

図1：本件明細書1の図9

### 3) 経過

本件発明に係る特許出願（特願2004-235768号）の経過は、以下のとおりである。

平成16年 8月13日 出願，審査請求、早期審査の申し出  
平成17年 2月14日 拒絶理由通知  
平成17年 4月14日 意見書，補正書提出  
平成17年11月30日 特許査定  
平成17年12月22日 登録

### 3. 被告プログラム

被告プログラムは、「ボックスをつなげていく形でPepperのソフトウェアを実装することができる開発環境」であり、コンピュータ上で、Pepperで稼働するソフトウェアを開発するためのプログラムである。

被告プログラムを起動し、新規プロジェクト（ロボットを動かすアプリの作成に必要なファイルの集まりをいう。）を作成すると、behavior.xar が作成される。behavior.xar は、テキストエディタ等で読むことができるファイルである。

被告従業員による解説書では、behavior.xar は「ビヘイビア」と呼ばれている。

「ビヘイビア」とは振る舞いを定義する単位であり、複数の「ボックス」（例えば「Say ボックス」等のビヘイビアの機能単位）から構成される。

「ボックス」は、「アプリの一連の振る舞い（フロー）を定義するための基本構成要素」と定義され、フローダイアグラム上にボックスを追加すると、behavior.xar に<Box>タグに含まれるデータ（「ボックスデータ」という。）が追加され、<Box>タグには、<Parameter>タグに含まれる値や、<Script>タグに含まれる Python コード（「演算用スクリプト」という。）が存在する。

例えば、「Set Language」ボックスであれば、ロボットが話す言語を規定する演算用スクリプト、「Say」ボックスであればロボットに発音させたい言葉を含む変数や、そのテキストを発音するように命令する演算用スクリプトである。

被告プログラムにおいては、各ボックスにボックス識別情報とともに含まれるこれらの変数、演算用スクリプトその他の情報が、まとめて behavior.xar に集積され、記憶されている。

被告プログラムは、この「ボックス」同士を結合することによって、様々な「ビヘイビア」を作成し、それを（「プロジェクト」として）集積することによって、ロボットの多くの動作を規定したアプリの開発を可能にするものである（別紙3-1 被告プログラム説明書より、抜粋）。

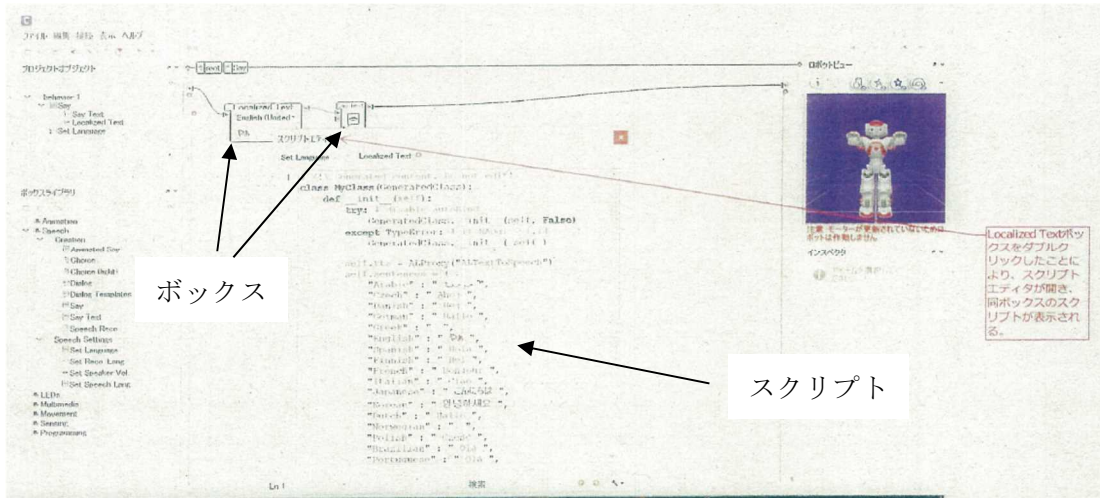


図1：別紙3-1に添付の図14

#### 4. 争点

争点は、以下のとおりである。

- 被告プログラムは、本件発明の技術的範囲に属するか（争点1）
- 本件特許は特許無効審判により無効とされるべきものか（争点2）
- 損害の発生の有無及びその額（争点3）

裁判所は、争点1を構成する複数争点のうち、

##### 争点1-2

被告プログラムは、「ノード識別情報」（構成要件B）、「親ノード識別情報」（構成要件E、G）、「ルートノード」（構成要件E）、「木構造」、「木構造表示ステップ」（構成要件G）を充足するか

##### 争点1-4

被告プログラムは、「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」（構成要件G）を充足するかについて、判断した。

#### 5. 裁判所の判断

##### 1) 争点1-2について

##### ア 「木構造」及び「親ノード」の意義

裁判所は、本願発明における「木構造」及び「親ノード」の意義を以下のように認定した。

「木構造」は、ノードを表示するラベルとラベル間を接続する結合線であるリードから構成される図として表現される表示に関する概念であって、基本となる

要素、すなわちルートから複数の要素に枝分かれをした階層構造を意味し、閉路を含まないものと解するのが一般的かつ合理的である。

「親ノード」は、あるノードに対して、当該ノードが属する階層内において当該ノードに直近して先行するノードを意味すると解するのが自然かつ合理的である。

#### イ 被告プログラム

裁判所は、被告プログラムを以下のように、認定した。

被告プログラムを起動して本件ロボットの振る舞いを構築する際に用いるボックスは、ボックスとして囲まれている範囲内に「Say」などとしてボックスの名前が表示され、また、左側に、シグナルやデータを受け取るための入力コネクタが設けられ、右側に、ボックスからのデータや終了を示すシグナルを出力するための出力コネクタが設けられており、これらの、ボックスの名前、入力コネクタ又は出力コネクタは、ボックスの構成要素である。入力コネクタ又は出力コネクタには、複数の種類があるが、「onStart」と呼ばれる入力コネクタは、ここにシグナルが送られると、ボックスが開始状態になることを意味するコネクタであり（以下、この入力コネクタを「onStartコネクタ」という。）、「onStopped」と呼ばれる出力コネクタは、ここからシグナルが送られた場合、ボックスが停止したことを意味するコネクタである（以下、この出力コネクタを「onStoppedコネクタ」という。）。

（中略）

本件behavior.xarで記述されるSayボックスは、別紙6の図1のとおり位置付けられているところ、SayボックスのonStartコネクタより後のフローダイアグラムは、別紙6の図2のとおりであり、リードで接続される順に、①SayボックスのonStartコネクタ、②LocalizedTextボックス、③SayTextボックス、④SayボックスのonStoppedコネクタとなっている。

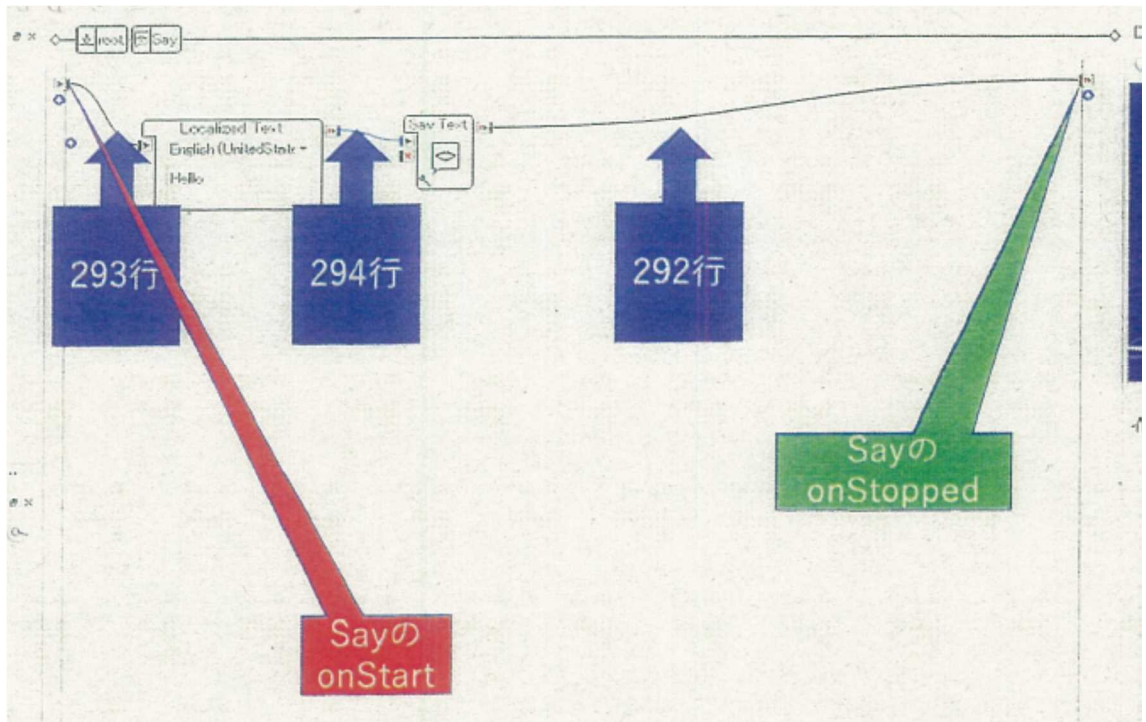


図2：別紙6の図2

#### ウ 検討

以上を踏まえて、裁判所は構成要件充足性を以下のように判断した。

Sayボックスについて、SayボックスのonStartコネクタから出発して、SayボックスのonStoppedコネクタに接続されているのであり、SayボックスのonStartコネクタ及びonStoppedコネクタは、いずれも、Sayボックスの構成要素である以上、Sayボックスのフローダイアグラムにおけるボックスの接続関係は、Sayボックスから出発してSayボックスに戻る閉路として表示されていることになり、木構造であるとはいえない。

(中略)

そうであれば、被告プログラムは、「木構造」を有しているとはいえず、したがって、「木構造表示ステップ」(構成要件G)を充足しないというべきである。

#### 2) 争点1-4について

ア 「ノードデータテーブル表示ステップ」及び「ノード変数データ」の意義について  
裁判所は、本願発明における「ノードデータテーブル表示ステップ」及び「ノード変数データ」の意義を以下のように認定した。

「ノードデータテーブル」とは、「ノードデータ」の一覧表であり、上位ノード

変数データ、自ノード変数データ及び代入用スクリプトを同時に表示するものと解するのが一般的かつ自然である。

「ノード変数データ」は、変数の値を意味すると解するのが自然かつ合理的である。

#### イ 被告プログラムの認定、及び構成要件充足性の判断

裁判所は、被告プログラムの認定、及び構成要件充足性の判断について、以下のよう述べた。

まず、原告は、被告プログラムのフローダイアグラム画面上のインスペクタに表示された入力コネクタの名称が「上位ノード変数データ」に当たると主張しているところ、入力コネクタの名称は変数の値ではないから、「上位ノード変数データ」に当たると認めることはできない。よって、被告プログラムは、「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」を充足しない。

### 6. 結論

裁判所は、被告プログラムは、「木構造表示ステップ」（構成要件G）、並びに、「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」（構成要件G）を充足しないから、本件発明の技術的範囲に属すると認めることはできず、原告の請求を棄却した。

### 7. 考察

構造化データをその構造が理解しやすいように画面表示する点において、本件特許と被告プログラムとは類似の構成である。しかし、本件特許が扱うデータは文書データを想定しており、被告プログラムはロボットの制御プログラムであった。

そのため、本件特許で扱うデータは木構造で表示され、かつ、ノードで示されるルートは閉路を含まないと認定された。一方、被告プログラムが扱う制御プログラムはオブジェクト指向プログラミングと推察され、ボックスで示されるオブジェクトのメソッドが、ノードを用いて表示している。一見すると本件特許と同様であるが、論理的には閉路を形成しているため、被告プログラムは本件特許の技術的範囲に含まないと、裁判所は判断した。

特許請求の範囲の記載は、権利範囲を広く確保するために上位概念で記載し、それをサポートするために、下位概念としての実施の形態を種々記載する。下位概念は無数にあり、記載できる実施の形態には限界がある。

本件明細書に記載の実施の形態では、構造化データとして文書データが前提であり、他の構造化データ、被告プログラムのようなオブジェクト指向プログラムの構造化データ

は考慮されていない。

したがって、本件のような判断結果に帰着するのは必然と考える。明細書作成時には、請求項に記載された発明を広くカバーするように可能な限り、種々の実施の形態を記載すべことを改め確認したい。また、権利行使時には、特許の技術的範囲の技術的意義の解釈には明細書の記載が参酌されうることを念頭に置いた上で、イ号は技術的範囲に含まれるか否かを判断すべきであろう。

以上