

(別紙 3 - 1)

被告プログラム説明書

被告プログラムの構成は、下記のとおりである。

5 第 1 被告プログラムの画面等の説明

- 図 1 被告プログラムを起動し、新規プロジェクトを開いた直後の画面
- 図 2 図 1 の状態のbehavior.xar
- 図 3 図 1 の状態から、Set Languageボックスをフローダイアグラム画面上に配置したときの画面
- 10 図 4 図 3 の状態のbehavior.xarの内容
- 図 5 図 3 の状態から、フローダイアグラム左端のコネクタとSet Languageボックスの入力コネクタとを結合させたときの画面
- 図 6 図 5 の状態のbehavior.xarの内容
- 図 7 図 5 の状態から、Sayボックスをフローダイアグラム画面上に配置したときの画面
- 15 図 8 図 7 の状態のbehavior.xarの内容
- 図 9 図 7 の状態から、Set Languageボックスの出力コネクタとSayボックスの入力コネクタとを結合させたときの画面
- 図 1 0 図 9 の状態のbehavior.xarの内容
- 20 図 1 1 図 9 の状態から、Sayボックスの出力コネクタとフローダイアグラム画面右端のコネクタとを結合させたときの画面
- 図 1 2 図 1 1 の状態のbehavior.xarの内容
- 図 1 3 Set Languageボックスの変数をEnglishに設定したときの画面
- 図 1 4 Localized Textボックスをダブルクリックしたときの画面
- 25 図 1 5 Say Textボックスを選択し、同ボックスのインスペクタを表示した画面

- 5
- 図 1 6 Sayボックスを選択し、同ボックスのインスペクタを表示した画面
- 図 1 7 図 1 6 のSpeedの値に関するスライダーを使ってSpeedの値を80にした画面
- 図 1 8 Say TextボックスのSpeedに関する編集ウィンドウズと「親からの継承」チェックボックスを表示した画面
- 図 1 9 図 1 8 の「親からの継承」チェックボックスにチェックを入れた状態のSay Textボックスのインスペクタを表示した画面
- 図 2 0 図 1 8 の「親からの継承」チェックボックスのチェックを外した状態のSay Textボックスのインスペクタを表示した画面
- 10 図 2 1 behavior.xar内のSayボックスとSet Languageボックスの結合情報におけるボックス識別情報のidの値をbehavior.xar内にボックス識別情報として存在しない値に変更したのち、フローダイアグラムを表示した画面
- 図 2 2 Sayボックス、Localized Textボックス及びSay Textボックスを直列につないだときの画面
- 15 図 2 3 「ボックスに変換」機能を使い、Localized Textボックス及びSay Textボックスを、downPageボックスに変換した画面
- 図 2 4 図 2 3 でdownPageボックスをダブルクリックした画面
- 図 2 5 図 2 3 でdownPageボックスに変数karyoku（値：北海道発電所）、変数suiryoku（値：黒部発電所）を設定した状態で、downPageボックス左下のスパナの画像をクリックした画面
- 20 図 2 6 図 2 5 からdownPageボックスをダブルクリックし、表示されるSay Textボックスのスク립トエディタを開いた状態で、スク립トエディタのウィンドウ上で右クリックして表示されるメニューから「関数を挿入」、「親からの変数を取得」を選択した画面
- 25

第2 被告プログラムの内容

被告プログラムは、Windows、MacOS及びLinux対応版があり、それぞれのOSが
登録されたコンピュータ上で作動するプログラムである。

5 1 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。

10 2 behavior.xarの説明

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。

15 「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
20 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。

25 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。

5

3 情報の記憶及び表示

(1) 情報の表示

ア ビヘイビアは、フローダイアグラム上で、ボックスとボックス間の結合線で表現することによって、ビジュアルに表示される。

10 イ フローダイアグラム画面上のボックスを選択すると、インスペクタが更新され、「一般」「変数を設定」「入力/出力/パラメータ」等の項目が表示される（図15）。

15 「一般」には、当該ボックスの名前、当該ボックスの概要、タイプ、使用言語（コンピュータ言語）、フローダイアグラム画面上の当該ボックスの画像が表示される。

「変数を設定」には、当該ボックスに設定されている変数の名称、変数の内容が表示される。

「入力/出力/パラメータ」には、当該ボックスに付属する出力コネクタ、入力コネクタの説明、タイプ、性質が表示される。

20 ウ 「インスペクタ」には、(a)当該ボックスが有している入力・出力コネクタの名称及び(b)当該ボックスの変数名が表示され、インスペクタ上のアイコンをクリックすると、当該ボックスのコネクタや変数の追加、名称やタイプ等の編集、削除をすることができる。

25 さらに、インスペクタに表示された入力コネクタの名称は、スクリプトエディタにより表示されるスクリプトにも含まれている。

エ ボックス内左下のパラメータボタンをクリックすると、当該ボックスの変

数が表示され、これを設定することができる。

オ また、ボックスを右クリックして「スクリプトを編集」という項目をクリックすると、「スクリプトエディタ」が開き、当該ボックスに対応するスクリプトが表示される（図14）。

5

(2) フローダイアグラム画面上の操作とbehavior.xarの更新

ア フローダイアグラム画面上にボックスが設置されると、behavior.xar内に当該ボックスに含まれる変数、演算用スクリプトが加わり、内容が更新される。

10 フローダイアグラム画面上でボックスのコネクタ同士を結合すると、behavior.xar内に、ボックス間の結合関係に関する<Link>タグ内の情報が加わり、内容が更新される。

図1～図12は、フローダイアグラム画面上の操作が行われるたびにbehavior.xarが更新される様子を示したものである。

15 イ behavior.xar内においては、フローダイアグラムごとに、各ボックスにidが割り振られ、これによって各ボックスを識別することができる（以下、各ボックスに割り振られた「id」を「ボックス識別情報」という。）。

なお、ボックス識別情報は編集可能であり、番号以外を設定することもできる。

20 ウ 本件behavior.xar（別紙3-2）には、「Box name=" Set Language" id=" 2"」（18行目）、「Box name=" Say" id=" 1"」（92行目）等と表示されているように、各ボックスにはボックス識別情報が振られている。なお、同種類のボックスを作成した場合でも、それぞれ別個のボックス識別情報が付与される。

25 エ また、ボックスに付属する入力コネクタと出力コネクタをクリックしてドラッグすると線が伸び、別のコネクタ上でドロップすると両コネクタを結

合することができる。

オ 被告プログラムにおいては、あるボックスの入力コネクタに結合されている出力コネクタ及び当該出力コネクタを有しているボックスの結合情報が〈Link〉タグを付けて behavior.xar 上に記載されることになる。例として、
5 本件 behavior.xar（別紙 3-2）には、「Link
inputowner=" 1" indexofinput=" 2" outputowner=" 2" indexoutput=" 3
"」（301行目）という一節がある。

カ このうち、inputowner とは、情報をインプットされる側のボックス識別
10 情報を指しており、indexofinput は当該ボックスの入力コネクタに付され
ている識別情報を指す。また、outputowner とは、情報をアウトプットす
る側のボックス識別情報を指しており、indexoutput はボックスの出力コ
ネクタに付されている識別情報を指す。

キ つまり、本件 behavior.xar の上記一節は、「Set Language」ボックス（ボ
15 ックス識別情報は 2，別紙 3-2・18 行目）の onReady コネクタ（コネク
タ識別情報は 3，別紙 3-2・65 行目）から、「Say」ボックス（ボックス
識別情報は 1，別紙 3-2・92 行目）の onStart コネクタ（コネクタ識別
情報は 2，別紙 3-2・116 行目）に向けて、結合線がつけられているこ
とを示している。

ク ボックスの結合情報におけるボックス識別情報の id の値を behavior.xar
20 内にボックス識別情報として存在しない値に変更すると、結合線自体がな
くなってしまふ（図 21）。

4 behavior.xarの実行

ア 被告プログラムにおいて、アプリの再生が行われる場合、結合されたボ
25 ックスが出力コネクタのある結合元のボックスから、入力コネクタのある結
合先のボックスへという処理順序により処理される。

イ 被告プログラムにおいては、アプリをコンピュータ上で再生することによって、バーチャルロボットがどのような挙動をするかをロボットビュー画面（各画面の右上にあるロボット画像が表示されたウィンドウ）上で表示させ、アプリの動作を確認することができる。

5 この際、behavior.xar に含まれる演算用スクリプトを含むプログラムコード全体をコンピュータがインタプリタ方式で実行する。

ウ たとえば、「Set Language」ボックスにおいては、「変数を Set Language に設定」というウィンドウが開き、Language と書かれた右側にプルダウンメニューが表示される。

10 エ ここで、「Set Language」ボックスの言語を English と設定してビヘイビアを再生すると、「Set Language」ボックスのボックスデータである

「English」を変数として保存し、「Localized Text」ボックスのボックスデータである演算用スクリプトが当該ロボットに設定された変数

15 「English」を関数の戻り値として受け取り変数の呼出し値、「Localized Text」ボックスのボックスデータである「Hello」という単語を、「Say Text」ボックスに出力し、ロボットは「Hello」と発言する。

オ 「Set Language」ボックスの言語を Japanese と設定してビヘイビアを再生すると、「Localized Text」ボックスのボックスデータである「Japanese」を変数として保存、「Localized Text」ボックスのボックスデータである演算用スクリプトが当該ロボットに設定された変数「Japanese」
20 を関数の戻り値変数の呼出し値として受け取り、「Localized Text」ボックスのボックスデータである「こんにちは」という単語を「Say Text」ボックスに出力し、ロボットは「こんにちは」と発言する。

カ このように、ボックスデータである演算用スクリプトを含んだプログラム
25 コード全体を実行すると、ロボットが動作し、「Hello」や「こんにちは」などと発音する。「Hello」や「こんにちは」は、behavior.xar 内のデータ

である（別紙 3 - 2・194～215 行）。

5 Sayボックス及びSay Textボックスにおける変数Speedの更新

5 ア Say ボックスのインスペクタから Speed の値をスライダーを使って変更すると、Say ボックスの<Parameter>タグ内の<Speed>タグ内の value が変更される（該当箇所は、別紙 3 - 2・120 行）。

10 イ Say Text ボックスの Speed について編集画面を見ると、「親から継承する」というチェックボックスがある。チェックを入れると、behavior.xar 内の Say Text ボックスの<Parameter>タグ内の<Speed>タグ内の inherits_from_parents という変数が 1 になる（該当箇所は、別紙 3 - 2・178 行）。チェックが入っている状態で、Say Text ボックスのインスペクタに変数 Speed に関する情報を表示すると、Say ボックスの<Parameter>タグ内の<Speed>タグ内の value と同じ値が Say Text ボックスのインスペクタにおいて表示される。

15 このとき、スライダーSpeed の値に関するスライダーは、灰色に反転しており、変更できない。

 また、このとき、behavior.xar 内の Say Text ボックス内の<Parameter>タグ内の<Speed>タグ内の value は変更されない。

20 ウ 上記「親から継承する」チェックボックスのチェックを外すと、behavior.xar 内の Say Text ボックスの<Parameter>タグ内の<Speed>タグ内の inherits_from_parents という変数が 0 になる（別紙 3 - 2・178 行）。そして、Say Text ボックスのインスペクタから Say Text ボックスの Speed の値を変更することができるようになる。Speed の値を変更すると、behavior.xar 内の Say Text ボックス内の<Parameter>タグ内の<Speed>タグ内の value も変更される。

25

6 本件behavior.xar におけるSay Textボックスにおける変数sentenceに関する演算

ア Say Text ボックスのスクリプトには、次の記述がある（別紙3-2・153～156行）。

5

```
sentence = "¥RSPD="+ str( self.getParameter("Speed (%)") ) + "¥ "  
sentence += "¥VCT="+ str( self.getParameter("Voice shaping (%)") ) +  
"¥ "  
sentence += str(p)  
10 sentence += "¥RST¥ "
```

イ すなわち、Say Text ボックスの自ノード変数である sentence について、153行で、①" ¥RSPD=" という文字列、②Speed の値を文字列にしたもの、③" ¥" という文字列が sentence に代入される。str()は、Python の命令
15 文であり、()内のデータを文字列として認識するものである。呼び出した Speed の値が 80 である場合、str により" 80" という文字列が認識される。なお、このとき、Speed は、「親から継承」チェックボックスにチェックを入れている場合、上記5のとおり、直系上位ボックスである Say ボックスから承継される。

20

ウ 次に、154行で、①" ¥VCT=" という文字列、②Voice shaping の値を文字列にしたもの、③" ¥" という文字列が sentence に追加的に代入される。なお、このとき、Voice shaping も、Speed と同様に Say ボックスから継承することが可能である。

25

エ 155行で、p のデータを文字列にしたものが sentence に追加的に代入される。p は、150行に記載されている Say Text ボックスの入力コネクタ" onInput_onStart()" という関数に定義されたものであり、当該入力コ

ネクタに対応する出力ネクタである Localized Text ボックスの出力ネクタ onStopped(Self, sentences[sDefaultLang])]] (別紙 3 - 2・219 行) に対応するものである。

具体的には, Hello やこんにちはといった, 挨拶のテキストが p に代入される。

オ 156 行で, " ¥RST¥" という文字列が sentence に追加される。

カ 以上のスクリプトにより, sentence は,

「¥RSPD= [適用される Speed の値] ¥VCT= [適用される Voiceshaping の値]
¥ [p の内容を文字列にしたもの] ¥RST¥」

となる。

7 「ボックスに変換」機能

ア Set Language ボックス, Localized Text ボックス及び Say Text ボックスを直列につなぐ (【図 2 2】)。

このとき, ロボアプリは, Set Language ボックスと Say ボックスを直列でつないだ場合と同様に動作する。

イ 上記アの状態における behavior.xar において, 各ボックスに割り振られた id は, 次のとおりとなる。

| Set Language | Localized Text | Say Text |
|--------------|----------------|----------|
| id="2" | id="5" | id="3" |

ウ 上記アの状態における behavior.xar において, ボックス間の結合情報は次のとおりとなる。なお, 結合情報は, behavior.xar においては, root ボ

ックスのタグ内に作成される。

【rootボックスタグ内の結合情報】

```
<Link inputowner="2" indexofinput="2" outputowner="0" indexofoutput="2" />  
5 <Link inputowner="3" indexofinput="2" outputowner="5" indexofoutput="3" />  
<Link inputowner="5" indexofinput="2" outputowner="2" indexofoutput="3" />  
<Link inputowner="0" indexofinput="4" outputowner="3" indexofoutput="4" />
```

ここで、outputownerが出力側のボックスを示しており、inputownerが入
10 力側のボックスを示している。つまり、上記の結合情報は、idが①0から2
へ【1行目】，②2から5へ【3行目】，③5から3へ【2行目】，④3か
ら0へ【4行目】と結合していることを表している。

エ Localized Text ボックス及び Say Text ボックスを選択し、右クリックす
15 ると、メニューが表示され、その中に「ボックスに変換」という選択肢が
現れる。

「ボックスに変換」を左クリックすると、Localized Text ボックス及び
Say Text ボックスをひとまとめでしたボックスが作成される（新たに作成
されたボックスに「downPage」という名称を付けた）。

20 これを画面に表示したのが【図23】である。そして、【図23】から
downPage ボックスをダブルクリックした画面が【図24】である。

オ 上記エの状態における behavior.xar において、downPage ボックスに割り
振られた id は1である。

| Set Language | Localized Text | Say Text | downPage |
|--------------|----------------|----------|----------|
| id="2" | id="5" | id="3" | id="1" |

カ 上記エの状態における behavior.xar において、ボックス間の結合情報は次のとおり、root ボックスのタグ内の結合情報が変更され、downPage ボックスのタグ内に結合情報が新たに作成される。

5

【rootボックスタグ内の結合情報】

```
<Link inputowner="2" indexofinput="2" outputowner="0" indexofoutput="2" />
<Link inputowner="0" indexofinput="4" outputowner="1" indexofoutput="3" />
<Link inputowner="1" indexofinput="2" outputowner="2" indexofoutput="3" />
```

10

【downPageボックスタグ内の結合情報】

```
<Link inputowner="3" indexofinput="2" outputowner="5" indexofoutput="3" />
<Link inputowner="5" indexofinput="2" outputowner="0" indexofoutput="2" />
<Link inputowner="0" indexofinput="3" outputowner="3" indexofoutput="4" />
```

15

つまり、上記の結合情報は、rootボックスタグ内については、idが①0から2へ【1行目】、②2から1【3行目】、③1から0へ【2行目】と結合していることを表している。

そして、downPageボックスタグ内については、①0から5へ【2行目】、②5から3へ【1行目】、③3から0へ【3行目】と結合していることを表している。

20

8 「親からの変数を取得」機能

ア 被告プログラムでは、「ボックスに変換」機能を用いて複数のボックスをまとめた1つのボックスを作成する場合に、新たに作成されたボックスを、まとめられたボックスから見て、「親」と呼ぶ。

25

イ 上記7で扱ったロボアプリにおいて、downPage ボックスの編集ボックスから、downPage ボックスの変数を追加することができる。(変数名は、karyoku と suiryoku とした。【図25】)

ウ 次に、downPage ボックス内の Say Text ボックスのスク립トエディタを開き、スク립ト上で右クリックをすると、メニューが現れる。

メニュー内には、「関数を挿入」という選択肢が表示され、これにマウスポインタを合わせると、「親からの変数を取得」という選択肢が表示される。

「親からの変数を取得」にマウスポインタを合わせると、「親」であるdownPage ボックスで設定した変数 (karyoku, suiryoku) が表示される

(【図26】)。

karyoku をクリックすると、スク립トエディタのカーソル位置に、次の記載が追加される。

```
self.getParameter("karyoku")
```

この機能により、downPage ボックスの変数である karyoku を使ったスク립トを、Say Text ボックス内に書き込むことができる。

9 被告プログラムにおけるFDボックスと内壁コネクタ

behavior.xarを被告プログラムで表示し、Sayボックスを展開すると、フローダイアグラムの両端にコネクタがある(以下「内壁コネクタ」という。)

Sayボックスのように、ボックス内にフローダイアグラムを持つボックスを、フローダイアグラムボックス(以下「FDボックス」という)と呼ぶ。複数のボックスをまとめてFDボックスとすることにより、処理の流れが把握しやすくなり、部品化・共有化が促進され、メンテナンス性を高めることにつながるとされている。

FDボックスは、当該ボックスタグ内に

```
<Timeline enable="0">
  <BehaviorLayer name="behavior_layer1">
    <BehaviorKeyframe name="keyframe1" index="1">
      <Diagram >
        [ここに、FDボックス内に含まれるボックス等の情報が書かれる]
      <Diagram />
    </BehaviorKeyframe>
  </BehaviorLayer>
</Timeline>
```

の記述を含むボックスである。この記述により、FDボックス内にフローダイア
グラムが形成され、そこにボックスを配置することが可能となる。Sayボックス
については、別紙3-2の121行目から124行目及び295行目から298
行目が該当する。

FDボックス内のフローダイアグラムにおける内壁コネクタは、左端がFDボック
スの入力ボックスの入力コネクタと、右端がFDボックスの出力コネクタと同一の
コネクタとなり、FDボックスの入出力コネクタの名称を変更すると、FDボックス
内の内壁コネクタの名称も同様に変更される。

上記追加に関して、FDボックスの入出力コネクタの名称を変更すると、FDボッ
クス内の内壁コネクタの名称も同様に変更される点は、証拠（甲21）のとおりで
ある。この点に関して、「変換後のボックスの入力の種類は、（先頭の）変換元
ボックスの入力の種類となり、出力の種類は、（最後の）変換元ボックスの出力
の種類になります。」との説明がある。つまり、ボックスに変換機能を使ってFD
ボックスを作成する場合、最初の変換元ボックスの入力コネクタの種類が、FDボ

ックスの入力コネクタ（左端内壁コネクタ）に引き継がれ、最後の変換元ボックスの出力コネクタの種類が、FDボックスの出力コネクタ（右端内壁コネクタ）に引き継がれることになる。

5 Link情報において、内壁コネクタのidは、0とされる。これは、FDボックスを複数作っても変わらない。すなわち、同一のフローダイアグラム内でid=0は、内壁コネクタを指すことになる。また、idの識別性も同一フローダイアグラム内で判断される。

behavior.xarにおいて、rootボックスもFDボックスとして記述されている（別紙3-2・14行目～17行目及び304行目～307行目）。