

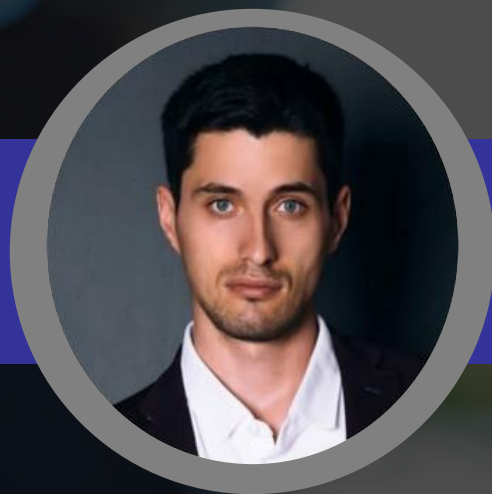
ビジネスルールマネジメントシステムの適用

2023年2月1日

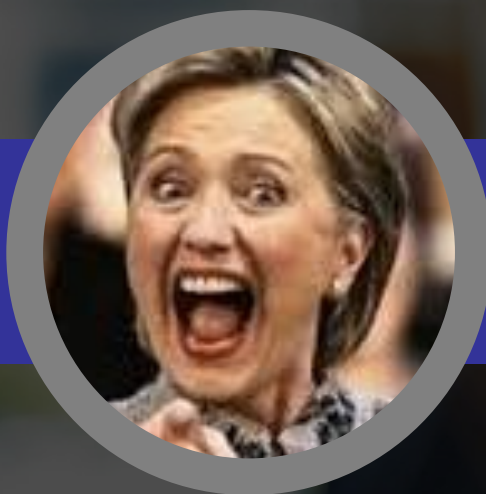


チーム マトリクス

チームとして、保険契約管理システムのための保険商品分析 および
ビジネスルールマネジメントシステム（BRMS）の適用を担当。



溝部 陽平
メンバー



五十島 萌
メンバー

写真はイメージです。



内田 聡
監修者

目次の内容

ビジネスルールマネジメント システムの適用



1.概要説明

- (1) ビジネスルールマネジメントシステムの概要

2.使用例

- (1) ビジネスルールマネジメントシステムの使用例

3.パラメータ取得の処理の流れ

- (1) 登場人物
- (2) キャラクター説明
- (3) 処理の流れ(i)
- (4) 処理の流れ(ii)
- (5) 処理の流れ(iii)

4.パラメータ登録の処理の流れ

- (1) 登場人物
- (2) キャラクター説明
- (3) 処理の流れ(i)
- (4) 処理の流れ(ii)

5.設計ポイントと運用ルール

- (1) パラメータ定義の管理体系
- (2) パラメータ定義の管理例
- (3) 設計ポイント
- (4) 運用ルール

1-(1) ビジネスルールマネジメントシステムの概要

概要

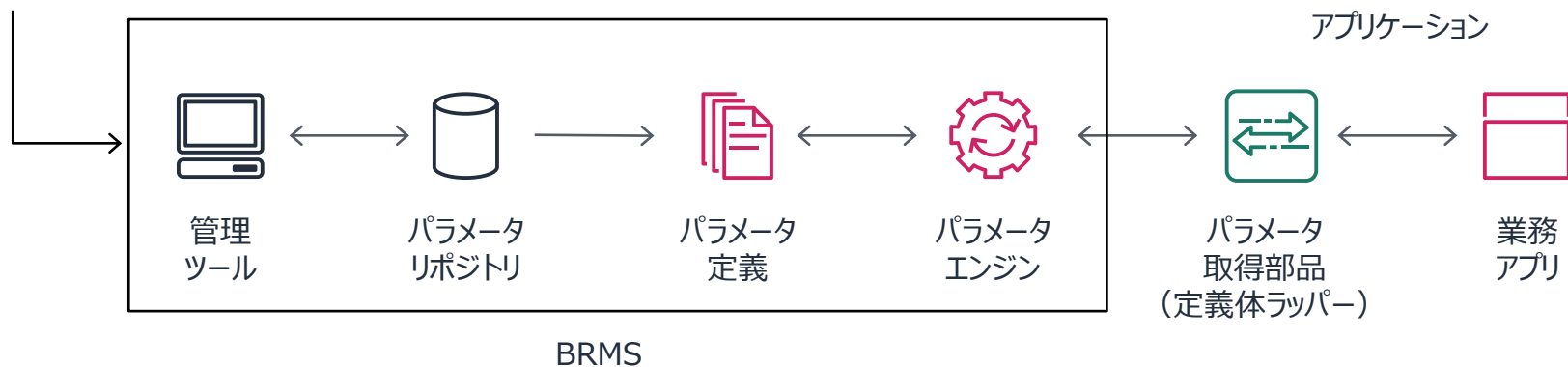
プログラム内の「ビジネスルール」と呼ばれるパラメータを外的に管理するシステム。
業務ロジックから定数値・処理パターン・条件を分離し、業務プログラム修正なしの機能変更を実現。



ユーザ



パラメータ表



利点

- ・テスト量の最小化。開発期間、コストの短縮が可能。
- ・パラメータを通じて、ユーザ・システムの部門間の認識共有を促進。

2-(1) ビジネスルールマネジメントシステムの使用例

ここでは、仮想の使用方法を通じて、BRMSの利用方法を説明する。

弁当会社のほっとMは、各店舗でより細かな情報連携による業績アップを狙って、ポータルアプリを導入することになった。



ほっとMの店舗情報をまとめると・・・

会社名	ほっとM	
部署	西日本	東日本
店舗	新大阪	札幌
	福岡	東京

西日本の店舗では-google
東日本の店舗ではヤホ-
の検索ブラウザを使用

札幌
フォントサイズ大

ポータルアプリで利用するブラウザの設定要望が各店舗で異なる。

3-(1) パラメータ取得の登場人物

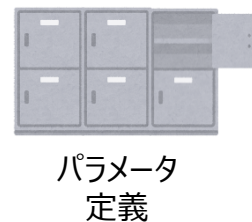
登場人物

以降のスライドでは、説明をわかりやすくするために、AからBのキャラクターに変換して説明を行う。

A



B



3-(2) パラメータ取得のキャラクター説明

キャラクターの説明



業務アプリ

取得した情報をもとに処理を行う人。
パラメータ取得部品に取得したい情報を問い合わせる。



パラメータ
取得部品

業務アプリとパラメータエンジンのブリッジ役。
業務アプリから取得した情報を翻訳し、パラメータエンジンへと渡す。



パラメータ
エンジン

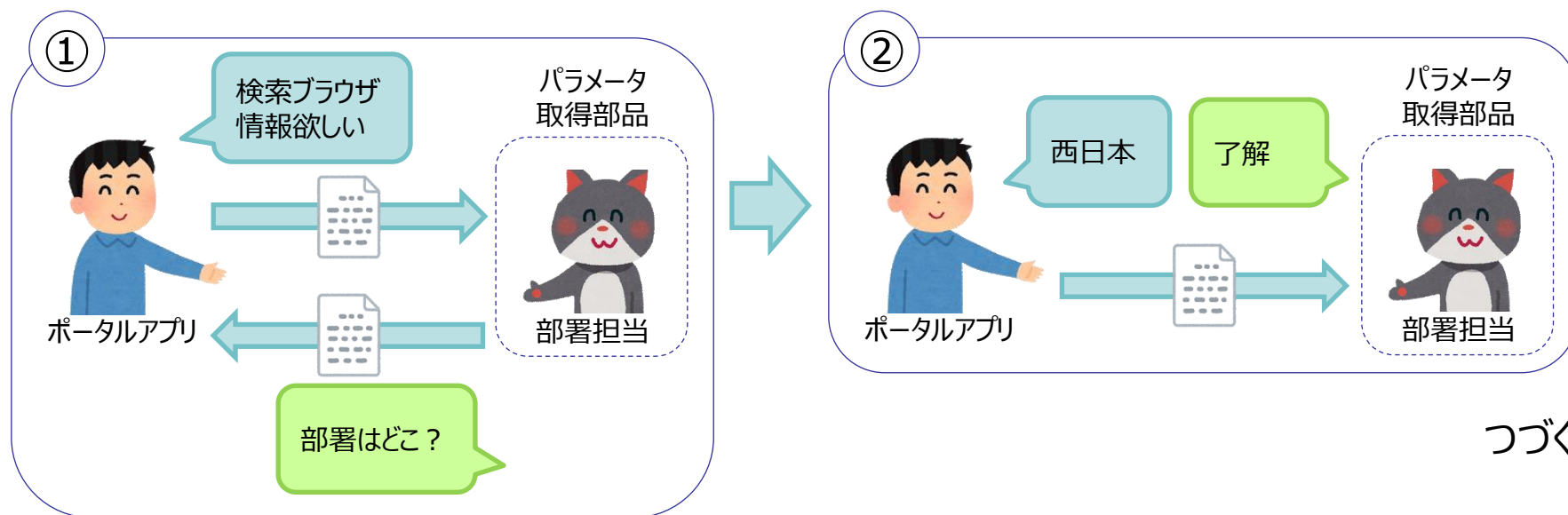
パラメータ定義から情報を取得する機械。
取得後、パラメータ取得部品へと情報を渡す。

3-(3) パラメータ取得の流れ(i)

流れの説明

ポータルアプリで利用するブラウザの設定情報の管理をBRMSを用いて行う。

パラメータ取得部品に「西日本」の「検索ブラウザ」について問い合わせる。
その際、探すために必要な情報をパラメータ取得部品へ渡す。

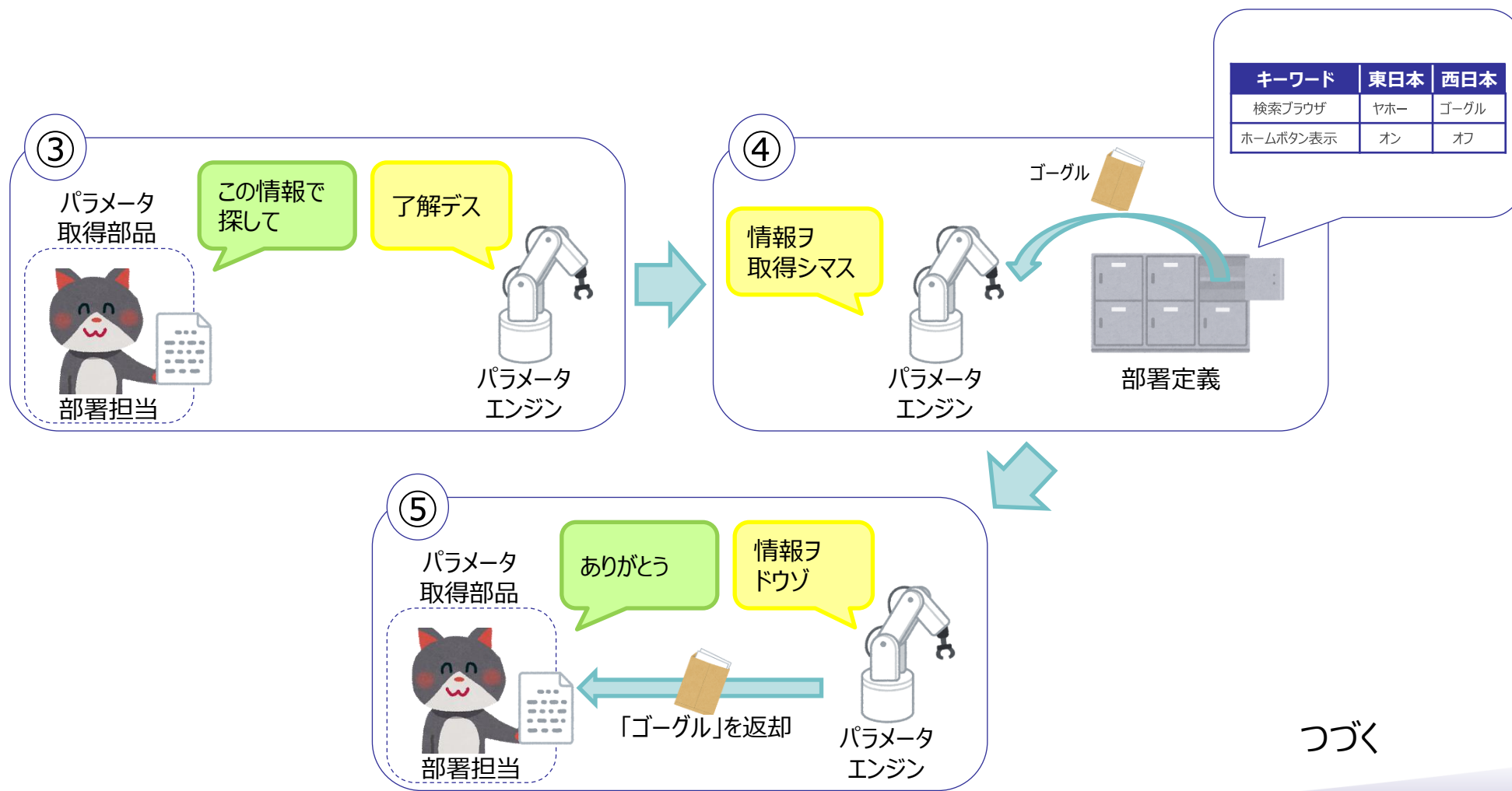


つづく

3-(4) パラメータ取得の流れ(ii)

流れの説明

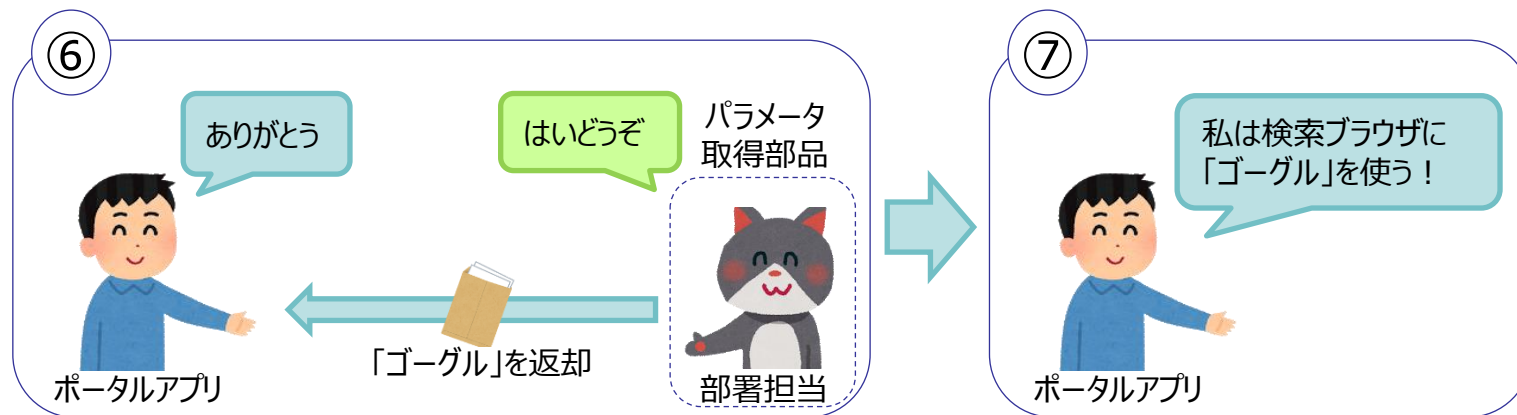
パラメータエンジンは、パラメータ取得部品から受け取った情報をもとに「-google」という情報を取得。



3-(5) パラメータ取得の流れ(iii)

流れの説明

ポータルアプリは、パラメータ取得部品を経由して取得した情報から、ブラウザの設定を行う。



おわり

4-(1) パラメータ登録の登場人物

登場人物

以降のスライドでは、説明をわかりやすくするために、CからDのキャラクターに変換して説明を行う。

C



ユーザ



管理
ツール



パラメータ
リポジトリ



パラメータ
定義



D



ユーザ



管理ツール



パラメータ
リポジトリ



パラメータ
定義

4-(2) パラメータ登録のキャラクター説明

キャラクターの説明



ユーザ パラメータ表

パラメータ表をもとに、パラメータ定義の値を登録・変更する人。
パラメータ定義をまとめたパラメータ表をもとに登録・変更を行う。



管理ツール

ユーザがパラメータ定義の値を登録・変更しやすくするツール。



パラメータリポジトリ

パラメータ定義を格納する倉庫。



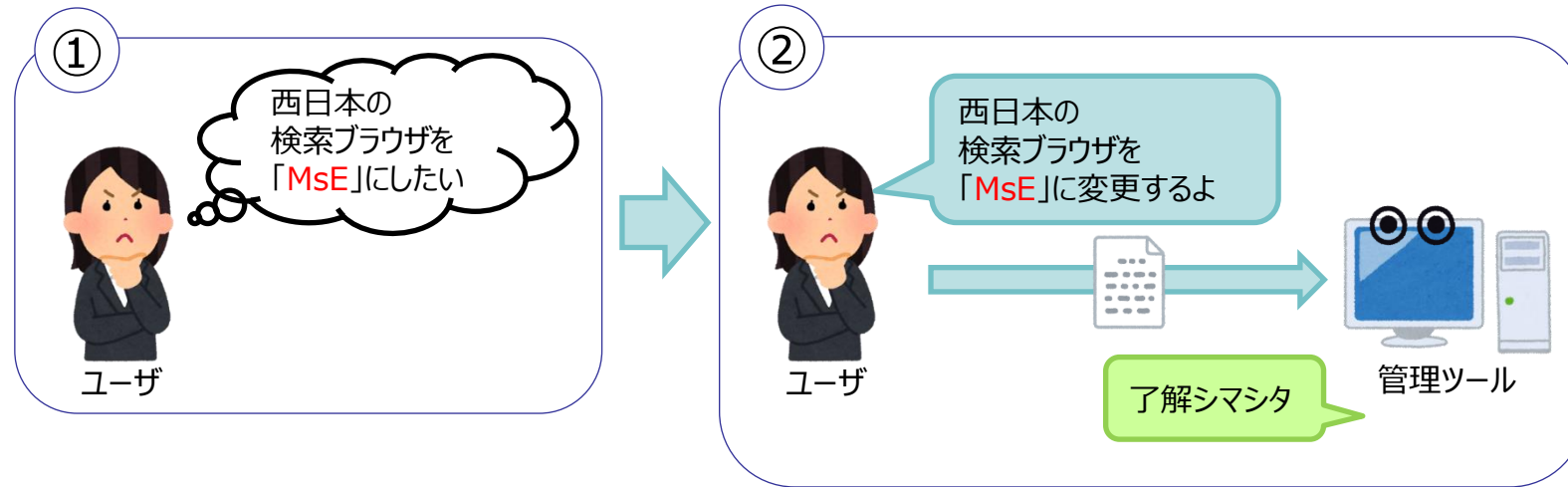
パラメータ定義

パラメータの値を格納するロッカー。

4-(3) パラメータ登録の処理の流れ(i)

流れの説明

ユーザが管理ツールを使用してパラメータ定義の値を登録や更新を行う。

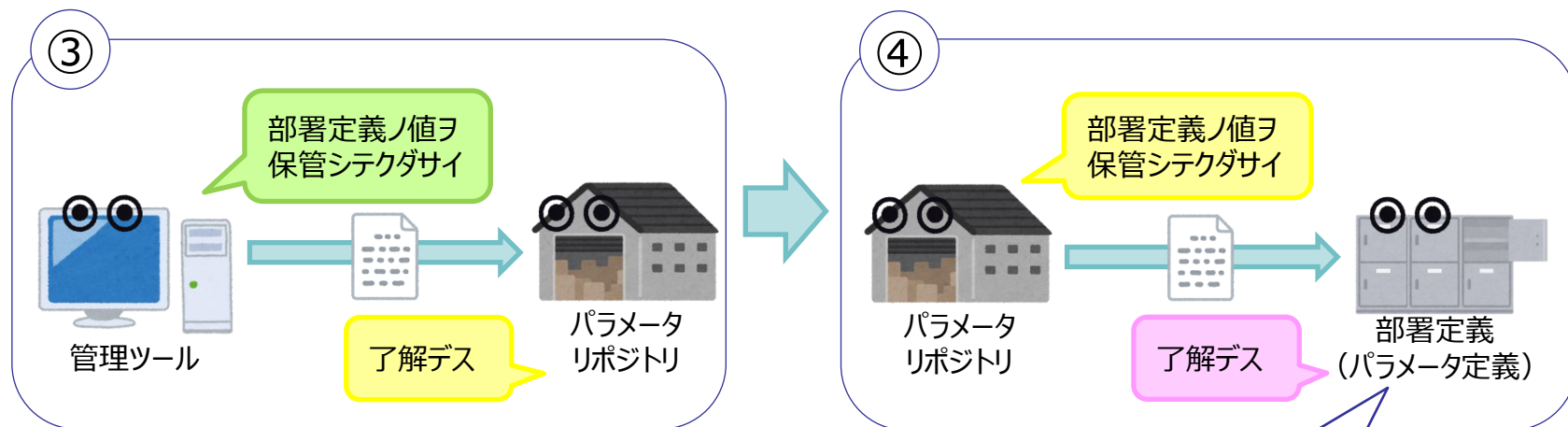


つづく

4-(4) パラメータ登録の流れ(ii)

流れの説明

ユーザが登録・変更した値は、パラメータリポジトリ内のパラメータ定義に保管される。



おわり

キーワード	東日本	西日本
検索ブラウザ	ヤホー	MsE
ホームボタン表示	オン	オフ

5-(1) パラメータ定義の管理体系

パラメータ定義の説明

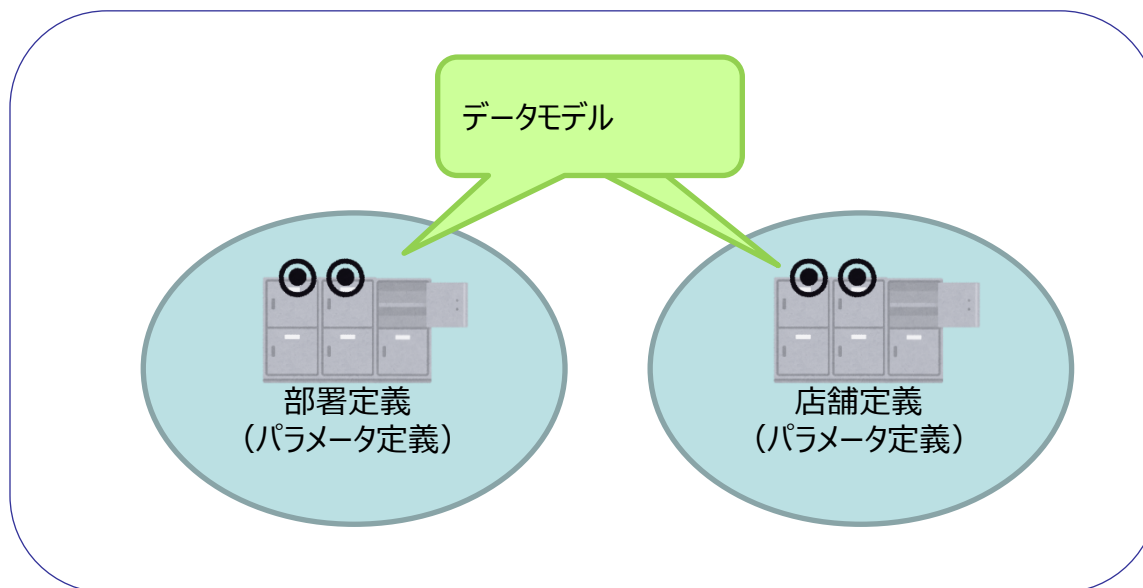


パラメータ定義

パラメータ定義の管理体系は細分化されており、大きい順にデータモデル、グループ、キーワードに分類して管理している。それぞれの説明は以下の通り。

➤ データモデル

パラメータ定義の値を管理する器、集合体。使用用途に応じて管理する単位に分ける。

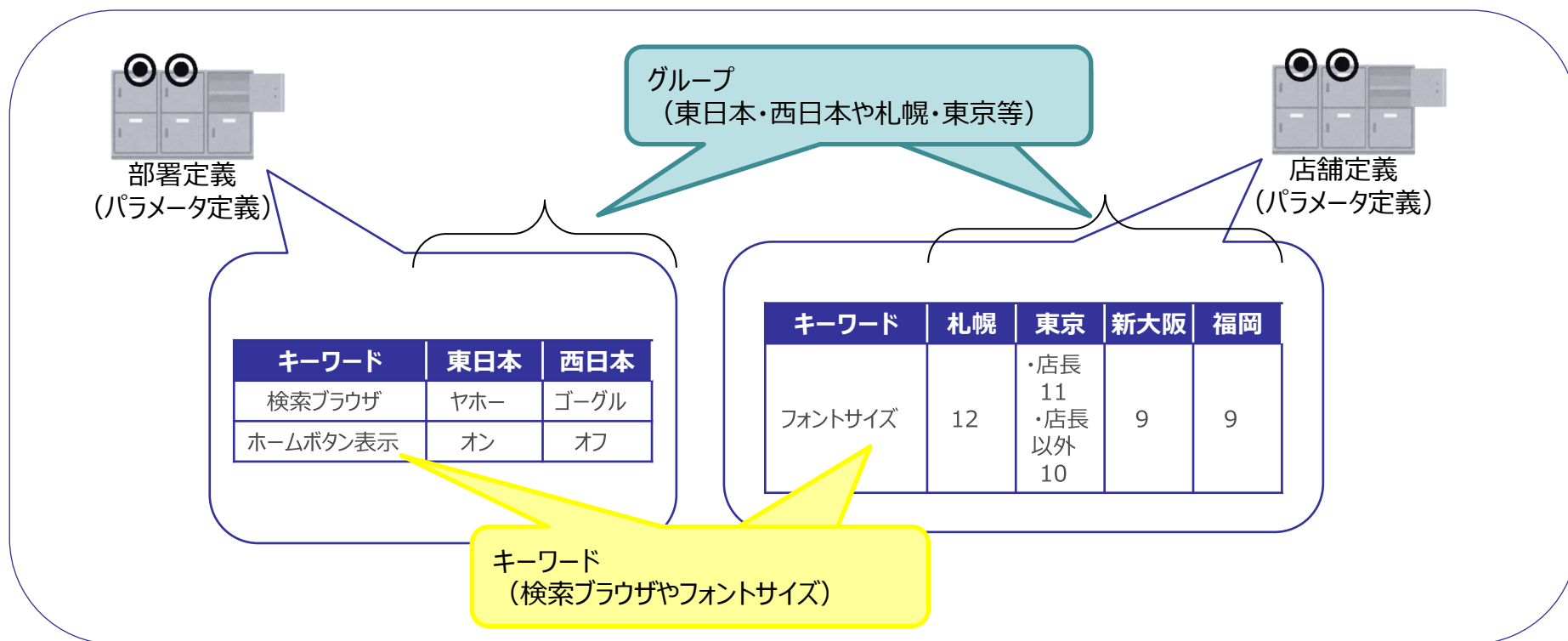


つづく

5-(1) パラメータ定義の管理体系

パラメータ定義の説明

- グループ
データモデルのパラメータ定義の値をグループ化したもの。
- キーワード
データモデルのパラメータ定義の値を管理する情報。



5-(2) パラメータ定義の管理例

ポータルアプリの管理例

今回のポータルアプリの例では、
データモデルは「会社定義」と「部署定義」と「店舗定義」に分割し管理する。

データモデル	グループ	キーワード
会社定義	「会社」単位	「会社名」や「本社住所等」の情報を管理
部署定義	「西日本と東日本」の 部署単位に分かれる	「検索ブラウザ」等の情報を管理
店舗定義	「新大阪、札幌、福岡、東京」 の店舗単位に分かれる	「フォントサイズ」の情報を管理



5-(2) パラメータ定義の管理例

ポータルアプリの管理イメージ

会社定義

キーワード	会社
会社名称	ほっとM
本社住所	福岡県〇〇市〇〇区 〇丁目〇-〇

西日本の店舗ではGoogle、
東日本の店舗ではYahoo-
の検索ブラウザを使用

札幌
フォントサイズ大

部署定義

キーワード	東日本	西日本
検索ブラウザ	Yahoo	Google
ホームボタン表示	オン(true)	オフ(false)

店舗定義

キーワード	札幌	東京	新大阪	福岡
フォントサイズ	12	・店長 11 ・店長以外 10	9	9

次ページ以降は「設計ポイント」と「運用ルール」を説明していく。



5-(3) 設計ポイント

【設計ポイント①】

➤ 部署または店舗分岐がある処理は必ずパラメータ定義の返却値から判断すること。

<パラメータ定義を**用いない**設計>

```
if (「部署コード」が"01" (東日本) の場合) {  
    ホームボタン表示処理へ  
}  
  
if (「店舗コード」が"001" (札幌) の場合) {  
    フォントサイズ ← 12  
} else if (「店舗コード」が"002" (東京) の場合) {  
    if (「店長コード」が"1" (店長) の場合) {  
        フォントサイズ ← 11  
    } else {  
        フォントサイズ ← 10  
    }  
} else {  
    フォントサイズ ← 9  
}
```

<パラメータ定義を**用いた**設計>

```
部署定義から「ホームボタン表示」を取得する。  
if (「ホームボタン表示」がオン(true)の場合) {  
    ホームボタン表示処理へ  
}  
  
フォントサイズ ← 店舗定義から「フォントサイズ」を取得する。
```

次ページは実際のプログラム(参考)となる

5-(3) 設計ポイント

実際のプログラム（参考）

<パラメータ定義を用いないプログラム>

```
if ( bushoCd.equals("01" (東日本) ) ) {  
    ホームボタン表示処理へ  
}  
  
if ( tenpoCd.equals("001" (札幌) ) ) {  
    fontSize ← 12  
} else if ( tenpoCd.equals("002" (東京) ) ) {  
    if ( tenchoCd.equals("1" (店長) ) ) {  
        fontSize ← 11  
    } else {  
        fontSize ← 10  
    }  
} else {  
    fontSize ← 9  
}
```

<パラメータ定義を用いたプログラム>

```
homeBottun = busho.getParameter(ホームボタン表示);  
if( homeBottun ) {  
    ホームボタン表示処理へ  
}  
  
fontSize ← tempo.getParameter(フォントサイズ);
```

例えば、新たな店舗(名古屋)のフォントサイズ(12)を追加したい場合、プログラムの修正は不要。
パラメータを追加するだけで開発可能となる。



5-(3) 設計ポイント

【設計ポイント②】

➤ 条件分岐や階層を深くしない設計にする。

定義値に条件式を登録することが可能。

条件分岐や階層が深いのであれば、保守し難いため、データモデルを分ける。

部署定義

キーワード	東日本	西日本
検索ブラウザ	ヤホー	グーグル
ホームボタン表示	オン	オフ
フォントサイズ	・札幌 12 ・東京 11 ・店長 11 ・店長以外 10 その他 9	9



部署定義

キーワード	東日本	西日本
検索ブラウザ	ヤホー	グーグル
ホームボタン表示	オン	オフ

店舗定義

キーワード	札幌	東京	新大阪	福岡
フォントサイズ	12	・店長 11 ・店長以外 10	9	9

フォントサイズについて、部署定義に条件式で登録することは可能。
店舗毎に値が変わるのであれば、店舗定義でフォントサイズを管理したほうが保守しやすい。



5-(4) 運用ルール

【運用ルール①】

- キーワードはカテゴリ化する。(カテゴリーに分類する)
- キーワード名称は、適切な名称。キーワード説明は、正しく記載する。

部署定義

キーワード名称	キーワード英字名	キーワード説明	カテゴリー1	カテゴリー2	東日本	西日本
検索ブラウザ	SELECT_BROWSE R	ポータルアプリで利用するブラウザの種類を管理する。	ポータル	ブラウザ	ヤホー	グーグル
ホームボタン表示	DISPLAY_HOME_B OTTUN	ポータルアプリのホーム画面にボタンを表示するかどうかを管理する。	ポータル	ホーム画面	オン	オフ

正しく記載する

カテゴリーに分類する

キーワードの名称と説明を正しく記載しないと何のキーワードかわからなくなる。
キーワード情報をきちんと整理しておく、どれを変更したらよいか一目でわかる。



5-(4) 運用ルール

【運用ルール②】

- ▶ 共通キーワードと個別キーワードを使い分ける。

大規模案件など、開発者が多い、プログラムの量が多い場合には、開発部署等で個別のキーワードを用意して持ち主を明確にする。今回の使用例では、顧客管理用・店舗管理用のキーワードなど。

店舗定義

キーワード名称	キーワード英字名	札幌店	東京店	新大阪店	福岡店
フォントサイズ	FONT_SIZE	12	・店長 11 ・店長以外 10	9	9
(店舗管理用) おすすめ弁当	RECOMMENDED_BENTO	天丼弁当	焼肉弁当	唐揚げ弁当	チキン南蛮弁当
(店舗管理用) おすすめお茶	RECOMMENDED_TEA	おーいえいお茶	烏龍茶	四十六茶	五右衛門茶

個別キーワード

全て「共通キーワード」の場合、変更するたびに使用箇所の影響分析が必要。開発部署の要件でかわる可能性があるものは「個別キーワード」で管理したほうが、影響分析の負荷を下げる。



5-(4) 運用ルール

【運用ルール③】

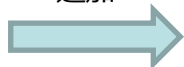
- 類似キーワードが増えないように統制する。不要なキーワードを増やさない。

設計者(Aさん)が、「検索ブラウザ」のキーワードがあるのに対して「初期ブラウザ」のキーワードを追加した場合、似たようにキーワードがどんどん増え、他の設計者がどちらを使用するか混乱を招く原因となる。

Aさん

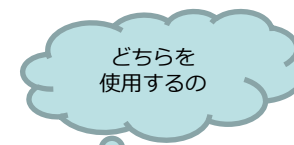


追加



部署定義

キーワード名称	東日本	西日本
検索ブラウザ	ヤホー	グーグル
初期ブラウザ	ヤホー	グーグル



Bさん



統制者

追加・修正する場合は統制者側でチェックする。
使用方法があっているかどうか、同様の申請が既にされていないかどうか。

統制者は人間が1個ずつ内容を確認するので、大変。責任重大。

ここがうまくできないとプロジェクトが失敗する。

最初は、整理が大変であるが、1開発が完了すると理解者が増えて、負荷が軽減されていく。