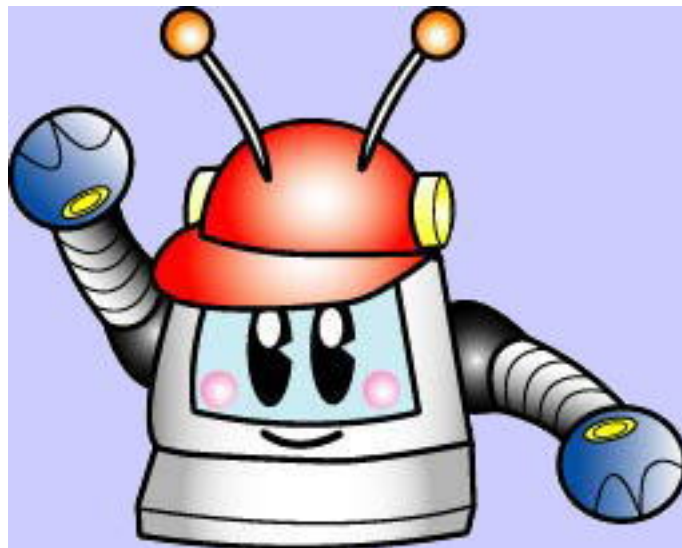


Webロボジェイピー®Struts版について



株式会社ユーサイドシステム

目次

1 Web ロボとは.....	3
2 Web ロボの技術的立場と役割.....	3
3 Web ロボの性能.....	4
4 Web ロボ利用のメリット.....	7
5 実行環境条件.....	8
6 生成部品.....	8
7 プログラムパターン.....	13
8 信頼性について.....	15
9 差込機能.....	16

1 Web ロボとは

Web ロボは、30にのぼる業務用システムアプリケーション開発に必要なロジックパターンの定義プログラムを用意し、定義プログラムが要求する項目について登録することにより、Java のStrutsフレームワークに対応する Web システムのソースを自動生成するシステムです。

実行に必要なすべての部品を生成し、ほぼそのままでも実行することが可能です。

これまでの、各種ツール、フレームワークに例を見ない、画期的でまさに Web ロボのネーミングにふさわしい、Web 開発のロボットと言っても過言ではありません。最低の定義で、最高の自動生成を、「定義＝プログラム」の理想を追求しております。

プログラム生成自動化の歴史は今にはじまったものではなく、これまで多くのトライがありましたが、多くは失敗に終わり、日本のソフト業界においてはジェネレーター、自動化という言葉はマイナスのイメージでとらえられ、そのトラウマにとらわれており、結局あいかわらずの手組みでの開発が主流になっております。

弊社は、物作りがロボット化によって日本の高賃金を克服しているように、ソフト産業においても、オフショアに現象されている高コストをソフト開発のロボット化によって克服すべきであるとの思いで、これまでのトラウマにあえて挑戦しております。

2 Web ロボの技術的立場と役割

この十年のコンピュータの技術的潮流の変化は、極めて激しく、コンピュータに携わるものにとっては戸惑いを隠せません。

Web ロボを開発してきた私どももこの間、こうした変化にどのように対応していくべきなのか、どの技術を選択して新たなこれからのシステム開発に関わっていくべきかに悩んでまいりました。パソコン及びインターネットの普及を前提とする時、これからのシステム開発のインフラ技術は Web システムであろうと結論しております。

さらに、Java をベースとしたオープン系か、. NETをベースにした WINDOWS 系でいくかの選択肢がありますが、より技術的なフリーハンドを担保するためには、Java をベースにすべきものと結論いたしました。

その結論の結果として、必然的にグローバルスタンダードの地位を得つつある、Strutsを基本フレームワークとして選択いたしました。

次に作表技術とリッチクライアント技術の選択という課題がありますが、作表はほぼPDFが一般的なものとなりつつあり、Web ロボも迷わず PDFLIB を利用しての PDF での作表を選択しております。

最後に残った大きな問題として、リッチクライアントをどのように実現すべきかがあります。

私どもの考え方として、Web システム構築に際しては極めて多くの技術を駆使しなくてはなりません。出来れば言語的には Java に限定できないかと考えております。

オープン系の問題点とし、GUI 技術が WINDOWS に遅れをとっており、今後この分野でより技術開発をすべきことと、リッチクライアントを GUI の課題ととらえるべきとの観点からも、アプレットでリッチクライアントを実現すべきと考えております。

最近 Ajax が取り上げられておりますが、リッチクライアントについてはこれまでも、いろいろ取り上げられてきましたが、まだ確立したものになっておらないのが現状です。

アプレットの問題点としてスピードがありますが、これまでのコンピュータシステムの歴史を振り返れば、スピード、容量の問題の解決こそがコンピュータの歴史であり、スピードの問題も早晚解決するものと思われ、リッチクライアントを GUI ととらえる限り 私どもはアプレットを選択すべきとの

立場であります。

Java のバージョンアップでも、Swingのパフォーマンスの改善がテーマにもなっており、スピードアップが期待されます。

これからも、ますます新たな技術的な提案がなされ、それらの見極めに多くの時間を割かざるを得ない宿命をコンピューター業界は背負っておりますが、しっかりした技術ベースを前提にしていきたいと考えております。なぜなら、システムは技術のはやりすたりに合せて作り変えるものではないのですから。

基本的には、安定的で使いやすく、コスト負担が少なく、システム保守がしやすいということがユーザーの要求条件であり、それに応えることが Web ロボの役割と認識しております。

もうひとつの問題として、日本において自動生成の是非についての議論があります。いわく、自動化によってプログラム技術のレベルが低下する。

いわく、現場作業的なプログラム技術はより賃金の安いオフショアにまかせ、日本では上流工程のシステム設計に力点をおくべきである。

後者に対しては、ソフト産業の技術は、設計技術とプログラム技術双方が必須であり、プログラム技術を失えば、必ず技術的空洞化を起こすことは明らかです。

前者に対しては、定型的なプログラムは自動的に作成すべきで、それによってより効率よく、安定した品質のソフト開発ができるのであり、生産性をあげ付加価値をたかめることが、ソフト産業にも求められております。又、極めてハードルが高い JavaWeb システム技術を自動生成されたソースより技術習得することによりそのハードルが下がり、より革新的なことに人的な資源を投入することが可能になります。

3 Web ロボの性能

自動化プログラムの性能評価をする場合以下の要素があげられるものと思います。

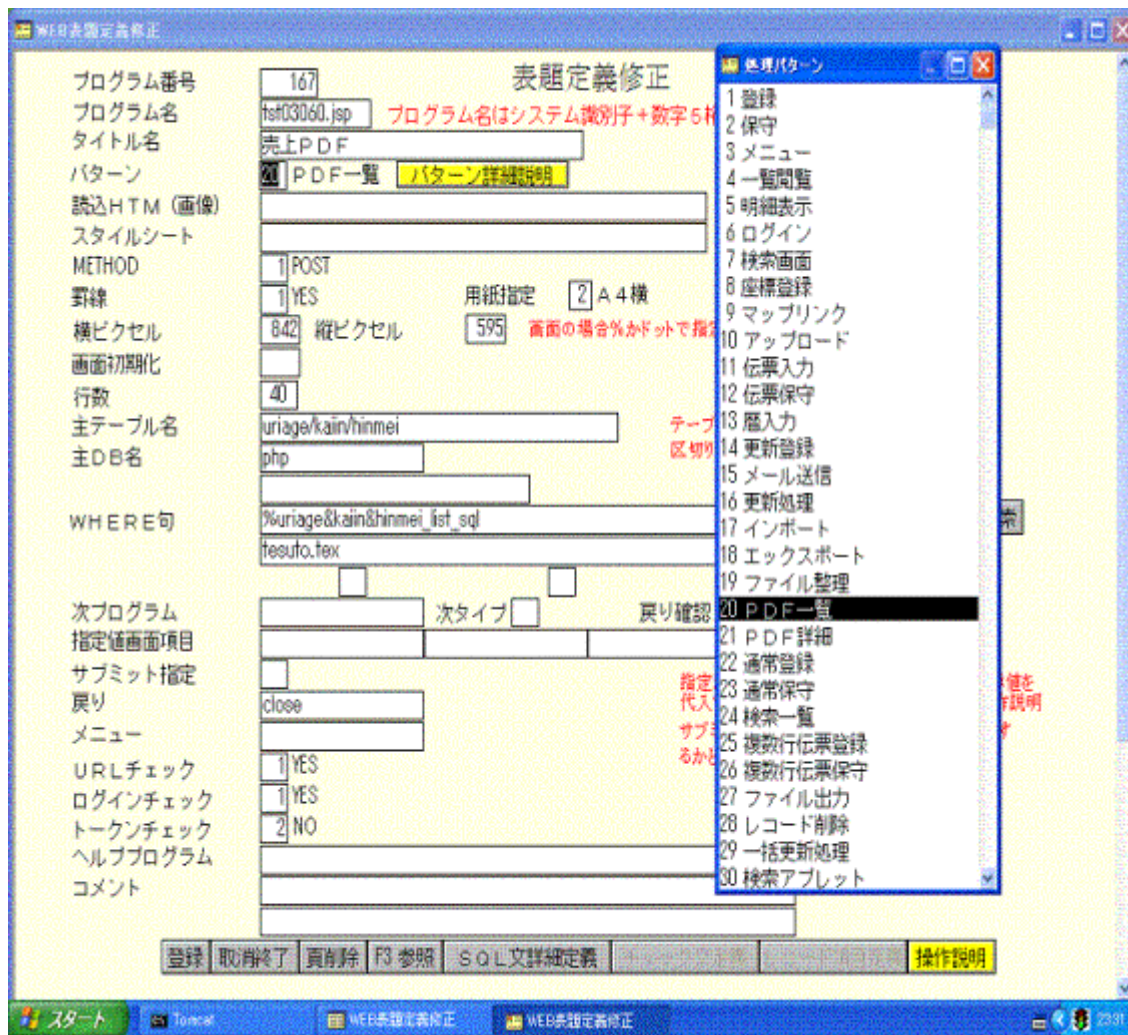
1. 簡便性

いかに簡単な定義で、いかに完成品に近いプログラムを生成するかこの点ですが、下記のような PDF による売上一覧表の例でご説明します。

プログラム名	タイトル名	プログラムパターン	ページ行数	主テーブル名	SQL 関数	次プログラム等
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	3	3	3	3	3
4	4	4	4	4	4	4
5	5	5	5	5	5	5
6	6	6	6	6	6	6
7	7	7	7	7	7	7
8	8	8	8	8	8	8
9	9	9	9	9	9	9
10	10	10	10	10	10	10
11	11	11	11	11	11	11
12	12	12	12	12	12	12
13	13	13	13	13	13	13
14	14	14	14	14	14	14
15	15	15	15	15	15	15
16	16	16	16	16	16	16
17	17	17	17	17	17	17
18	18	18	18	18	18	18
19	19	19	19	19	19	19
20	20	20	20	20	20	20
21	21	21	21	21	21	21
22	22	22	22	22	22	22
23	23	23	23	23	23	23
24	24	24	24	24	24	24
25	25	25	25	25	25	25
26	26	26	26	26	26	26
27	27	27	27	27	27	27
28	28	28	28	28	28	28
29	29	29	29	29	29	29
30	30	30	30	30	30	30
31	31	31	31	31	31	31
32	32	32	32	32	32	32
33	33	33	33	33	33	33
34	34	34	34	34	34	34
35	35	35	35	35	35	35
36	36	36	36	36	36	36
37	37	37	37	37	37	37
38	38	38	38	38	38	38
39	39	39	39	39	39	39
40	40	40	40	40	40	40
41	41	41	41	41	41	41
42	42	42	42	42	42	42
43	43	43	43	43	43	43
44	44	44	44	44	44	44
45	45	45	45	45	45	45
46	46	46	46	46	46	46
47	47	47	47	47	47	47
48	48	48	48	48	48	48
49	49	49	49	49	49	49
50	50	50	50	50	50	50
51	51	51	51	51	51	51
52	52	52	52	52	52	52
53	53	53	53	53	53	53
54	54	54	54	54	54	54
55	55	55	55	55	55	55
56	56	56	56	56	56	56
57	57	57	57	57	57	57
58	58	58	58	58	58	58
59	59	59	59	59	59	59
60	60	60	60	60	60	60
61	61	61	61	61	61	61
62	62	62	62	62	62	62
63	63	63	63	63	63	63
64	64	64	64	64	64	64
65	65	65	65	65	65	65
66	66	66	66	66	66	66
67	67	67	67	67	67	67
68	68	68	68	68	68	68
69	69	69	69	69	69	69
70	70	70	70	70	70	70
71	71	71	71	71	71	71
72	72	72	72	72	72	72
73	73	73	73	73	73	73
74	74	74	74	74	74	74
75	75	75	75	75	75	75
76	76	76	76	76	76	76
77	77	77	77	77	77	77
78	78	78	78	78	78	78
79	79	79	79	79	79	79
80	80	80	80	80	80	80
81	81	81	81	81	81	81
82	82	82	82	82	82	82
83	83	83	83	83	83	83
84	84	84	84	84	84	84
85	85	85	85	85	85	85
86	86	86	86	86	86	86
87	87	87	87	87	87	87
88	88	88	88	88	88	88
89	89	89	89	89	89	89
90	90	90	90	90	90	90
91	91	91	91	91	91	91
92	92	92	92	92	92	92
93	93	93	93	93	93	93
94	94	94	94	94	94	94
95	95	95	95	95	95	95
96	96	96	96	96	96	96
97	97	97	97	97	97	97
98	98	98	98	98	98	98
99	99	99	99	99	99	99
100	100	100	100	100	100	100

上記プログラムのプログラム表題定義では、下図のようにプログラム名、タイトル名、プログラムパターン、ページ行数、主テーブル名、SQL 関数、次プログラム等のプログラム全体

の定義をします。



プログラム詳細定義では、表題部の出力項目、明細部の出力項目とその行列番号、合計計算項目、キーブレイク項目を定義しています。
たったこれだけの定義で、上記のPDFでの罫線入りで、合計入りの売上げ一覧表プログラムが作成されます。他パターンでもほぼ同様の定義です。

PDF一覧定義登録

項目定義登録 (PDF一覧)

プログラム番号 主テーブル名

項目番号	項目区分名	行番号	タイプ	計打出	キー番号	テーブル項目名(1)	テーブル項目名(2)	テーブル項目名(3)	項目数	出力項目	出力項目区分
1	1表題部	101	7出力			torihikisaki	torihikisaki	取引先	4.0		
2	1表題部	102	7出力			kainmei	kainmei	取引先名	40.0	1文字	
3	2明細部	101	7出力		1	torihikisaki	torihikisaki	取引先	4.0		
4	2明細部	102	7出力			denpyono	denpyono	伝票番号	4.0		
6	2明細部	103	7出力			hinmei	hinmei	品名	30.0	1文字	
7	2明細部	104	7出力			suuryou	suuryou	数量	4.1		
8	2明細部	105	7出力	1		tanka	tanka	単価	3.1		
9	2明細部	106	7出力	1			kingaku	金額	7.0		
5	2明細部	202	7出力			gyou	gyou	行番号	2.0		

登録 取消終了 前頁 次頁 行挿入 行削除 F3参照 操作説明

画面に表示する項目のタイトルを入力します。

スタート Tomcat WCP項目定義登録 PDF一覧定義登録 22:35

2. 柔軟性及び自動化率

簡便性と自動化率は本来、矛盾するものですが、Web ロボではパターン化と関数の差込機能によりこの矛盾を解決しております。

小部品を関数として定義し、それを各所で差し込みすることにより、又利用者が用意した独自クラスのインポート機能により柔軟性を担保し、高い自動化率を獲得することを可能にしています。

又、画面設計においては、画面作成ツールで作成されたHTMLは、Strutsのタグを差し込むことによって、Web ロボに簡単に取り込み出来ます。

3. 汎用性

Web ロボは、オープンな開発環境である Java 言語で、Strutsといグローバルスタンダードなフレームワークに対応したプログラムソースを生成します。

Java はオープンと言いながら、独自のフレームワークでは他のフレームワークとは共存できず、汎用性を失いかねません。Web ロボがStrutsを選択した理由がここにあります。

又、万一別なフレームワークであっても、Web ロボはソースの生成であり、大半の生成 Java クラスはそのままで利用することが可能です。

データベースもMy SQL、Postgres、Oracle、SQL サーバーに対応しております。

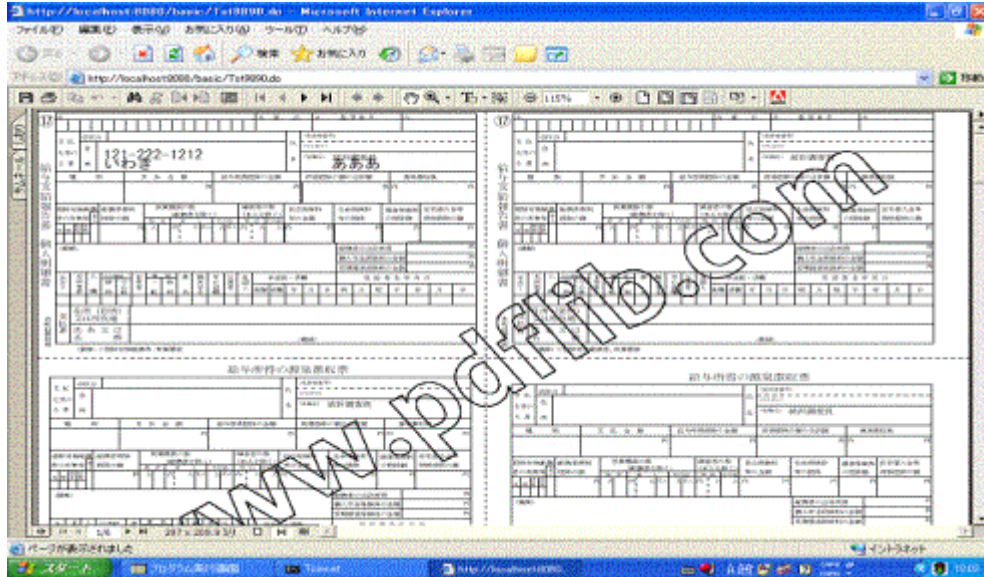
4. 信頼性

Web ロボは、例外処理を考慮したプログラム生成、SQL インジェクションに対応したプログ

ラム、権限管理機構、PDF のパスワード機能を利用した文書管理機構等、信頼性に配慮したシステムを提供することが可能です。

5. 難易度

合計処理機能を持った PDF、下記の複雑な PDF 帳票、コード検索のウインドウが開くアプレット、伝票形式でのアプレットプログラムなど難易度の高いプログラムを作成することが可能です。



4 Web ロボ利用のメリット

1. 最低一人は全技術についての知識が要求されますが、その指導のもとで Web ロボの定義ルールを理解することにより、コンピュータについての知識がない方をも即戦力化することが出来ます。
一方、COBOL等の言語に堪能でシステム経験が豊富な年齢の高い世代が、Web、Java等の新技術への対応ができずにおりますが、Web ロボはその世代の再活用を可能にし、若い世代の業務知識、経験の不足を補う大きな役割を果たすことが出来るようになります。又、極めて簡単な定義作業で、プログラム作成が可能ですので、大変高い生産性が獲得できます。
2. 一から Java、Struts、JSP、Java Script 等々の技術習得は極めて困難ですが、Web ロボにより生成されたプログラムにより、それらを理解する方が容易で近道となります。
Web ロボでは、定義用プログラム、自動生成プログラム及びアプレットの基本プログラムを除いて、すべてのソースを見ることが出来ます。
Web ロボは教育ツールとも言えます。
3. ますます複雑化するプログラム開発ですが、手組みをする場合はどのようなルールを作っても部品構成の統一化をはかることは困難となり、全体の体系及び個別プログラムの解析は厄介なものとなります。
Web ロボではすべての部品を自動生成しますので、それらの部品構成は Web ロボの生成ルールに則りますので同じものになり、又同じプログラムになります。
従って、システム及びプログラムの管理は極めて容易になります。
4. Web プログラム開発は、セキュリティー対策についても多くの配慮が要求されますが、手組

みによる場合、個人の能力知識に格差があり、すべてのプログラムに細かな配慮をすることは、至難の業と言わなければなりません。

Web ロボが用意するこれらの配慮が、すべてのプログラムに自動的に施されます。

5. システムの開発には必ず仕様の変更が伴います。

手組みの場合、それまで作られた部品の見直しと変更が要求され、そのために要する工数は極めて大きなものにならざるを得ません。

しかしながら、完全な仕様を詰めるためには、多くの時間を要するだけでなく、不可能に近いものがあります。又、下流工程を待たせることになります。

Web ロボでは、仕様の変更になった部分の定義を変えるだけの作業ですみますので、仕様変更による負担は軽微なものになります。

場合によっては、プロトタイプを作りながら仕様の決定をしていくことも可能です。

5 実行環境条件

PDF ツール	PDFLIB 正規版
データベース	PostgreSQL/MySQL/Oracle/SQL サーバー等
Web サーバー	Tomcat
フレームワーク	Struts
Java 環境	Java J2SDKEE

6 生成部品

Web ロボは、テーブル定義、プログラム表題定義、プログラム詳細項目定義、関数定義より、下記 I 図のように Struts に対応した実行に必要な部品をすべて生成します。

30 を超えるロジックパターンを用意し、そのパターンに必要な要素を定義するプログラムを用意することにより、簡単に厄介な Web システムを自動作成します。

① テーブル定義よりは

テーブル生成 SQL、テーブルビーン、テーブルアクセスのメソッドクラス、標準 SQL クラス、バリデータ xml、テーブルアクションクラス(アプレット 用)

② プログラム表題定義よりは

マッピング xml

③ プログラム詳細項目定義よりは

フォーム xml

④ %関数定義よりは

定義 SQL クラス、複合テーブルビーン、複合テーブルアクセスメソッドクラスが自動生成されます。

更に、テーブル定義、プログラム表題定義、プログラム詳細項目定義、関数定義より、アクションクラスプログラム編集クラス、JSP プログラム、アプレット プログラム(アプレットパターンの場合)が自動生成され、実行に必要なすべての部品が用意されます。

1. JSP パターンでのテーブルアクセス関係部品構成

Strutsも、一般的なMVCといわれる部品構成の手法をとっておりますが、Strutsに対応しても、部品構成の考え方は統一がとれません。従って、部品化が進めばすすむほど、又システムの規模が大きくなればなるほど、システムの把握は困難になってまいります。

Web ロボにおいては、下記 II 図のように極力部品の機能を明確にして、仮に生成部品に変更が生ずる場合、局所化するための部品構成を取っております。

クラスの部品構成としては、アクションクラス、編集プログラムクラス、SQL 編集クラステーブルアクセスメソッドクラス、テーブルビーンという階層クラス で構成されます。

2. アプレットパターンでのテーブルアクセス関係部品構成

JSP パターンの部品構成とアプレットでの部品構成は異なります。

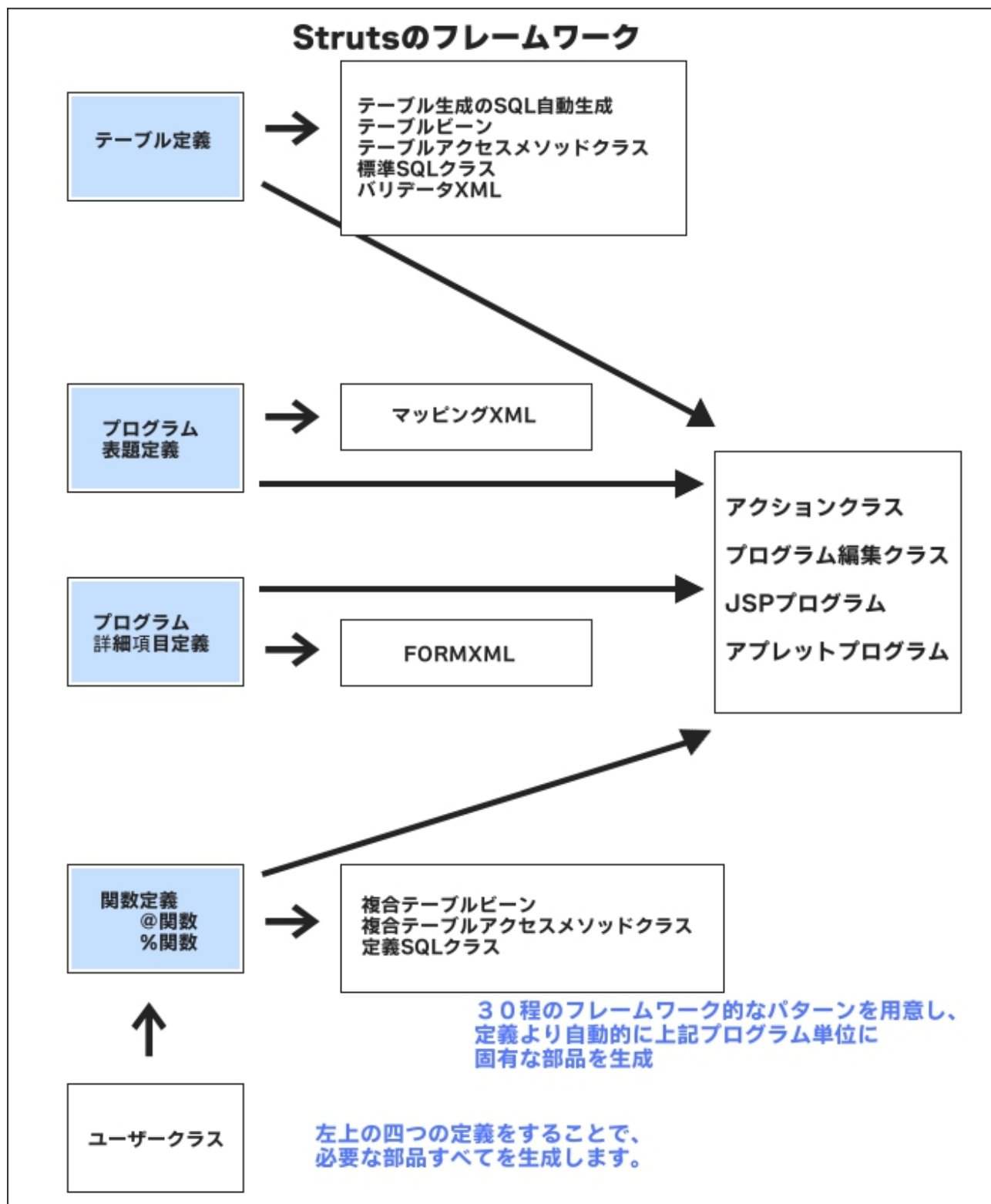
Web ロボでは、アプレットで必要になる下記 iii 図のようなテーブルアクセス関係部品をすべて自動生成します。

JSP との大きな違いは、アプレットにおいては編集プログラムクラス、SQL クラスを介さず、アプレットプログラムの中で直接 SQL 文を編集して、テーブルアクションクラスを起動して、テーブルアクセスメソッドクラスよりテーブルビーンという流れでの処理になります。

尚、アプレットプログラムの呼び出しは、アクションクラスを介して JSP プログラムを呼び出し JSP プログラムより、アプレットを呼び出しという流れになります。

従って、アプレットプログラムにおいても、アクションクラス、JSP プログラムが生成されます。

i 図: Web ロボ部品生成概念図



JSPテーブルアクセス関係部品構成

Webロボでは、テーブルビーンクラス、テーブルアクセスクラス、標準SQLクラスがテーブル定義より自動生成されます。クライアントからデータベースへのアクセスの流は、

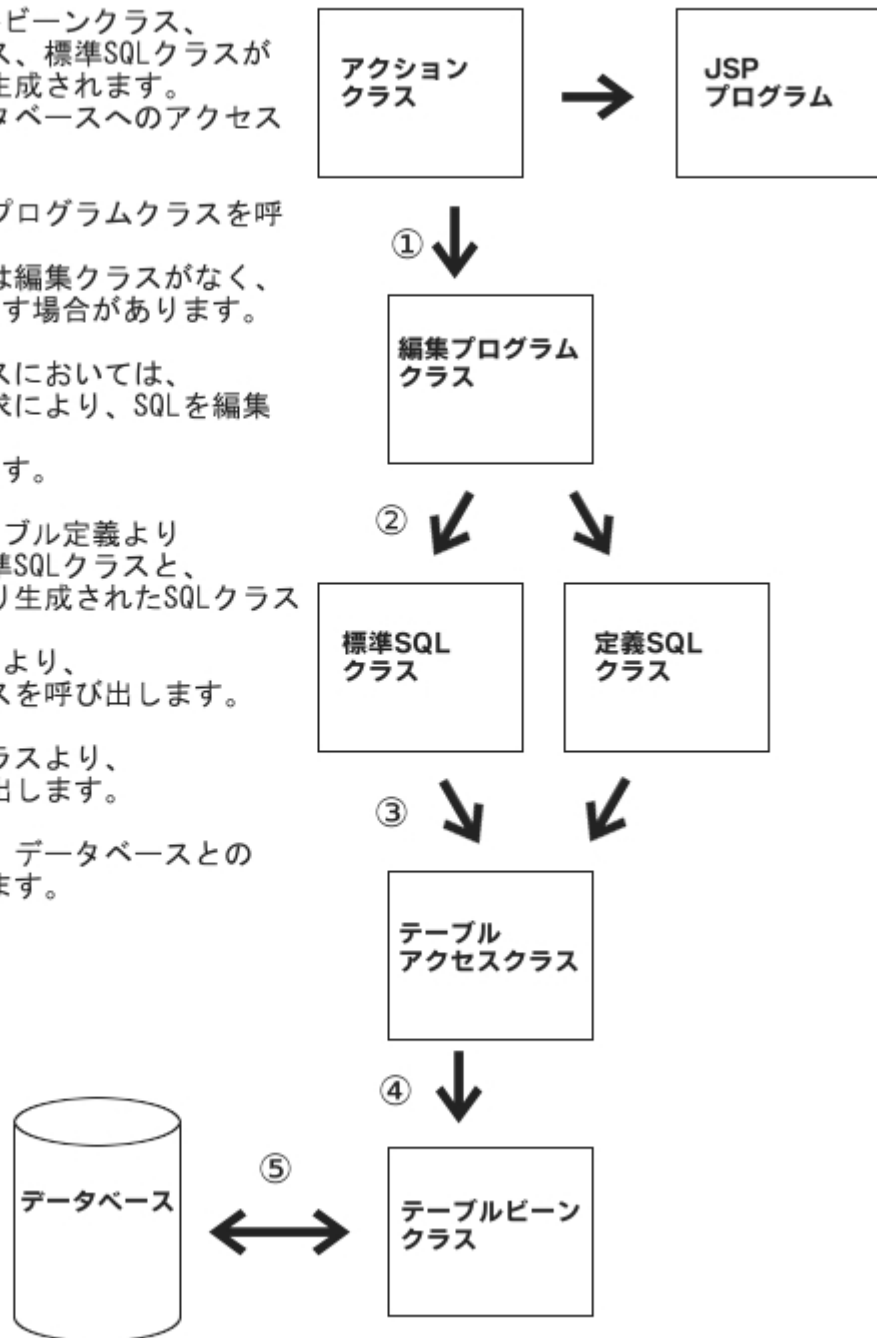
①アクションより編集プログラムクラスを呼び出します。
尚、パターンによっては編集クラスがなく、直接SQLクラスを呼び出す場合があります。

②編集プログラムクラスにおいては、アクションクラスの要求により、SQLを編集し、SQLクラスを呼び出します。

③SQLクラスには、テーブル定義より自動的に生成される標準SQLクラスと、関数で定義した関数より生成されたSQLクラスの二つがあり、SQLクラスより、テーブルアクセスクラスを呼び出します。

④テーブルアクセスクラスより、テーブルビーンを呼び出します。

⑤テーブルビーンより、データベースとのデータのやり取りをします。



アプレットタイプ関係部品構成

アプレットタイププログラムでは、
テーブルビークラス、
テーブルアクセスクラス、
テーブルアクションクラス
がテーブル定義より自動生成されます。
クライアントからデータベースへのアクセスの流れ
は、

⑥クライアントにおいて、SQL定義が生成される。
httpプロトコルにより、
サーバーのテーブルアクションを起動する。

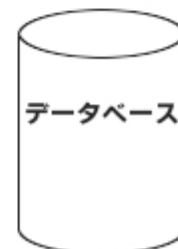
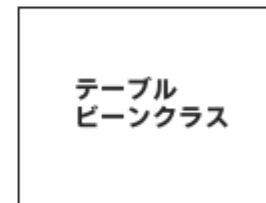
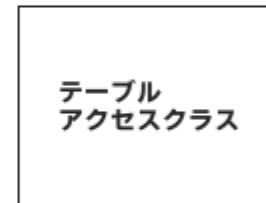
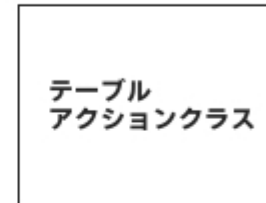
⑦テーブルのアクションクラスは、
テーブルアクセスクラスの
アプレットSQLメソッドを呼び出す。

⑧テーブルアクセスクラスより、
テーブルビークラスを呼び出す。

⑨テーブルビークラスより、データベースとの
やり取りをする。

⑩テーブルアクセスクラスのアプレットメソッドが、
取り込んだデータを項目の区切り文字に挿入し、
httpプロトコルでクライアントにデータを送信する。

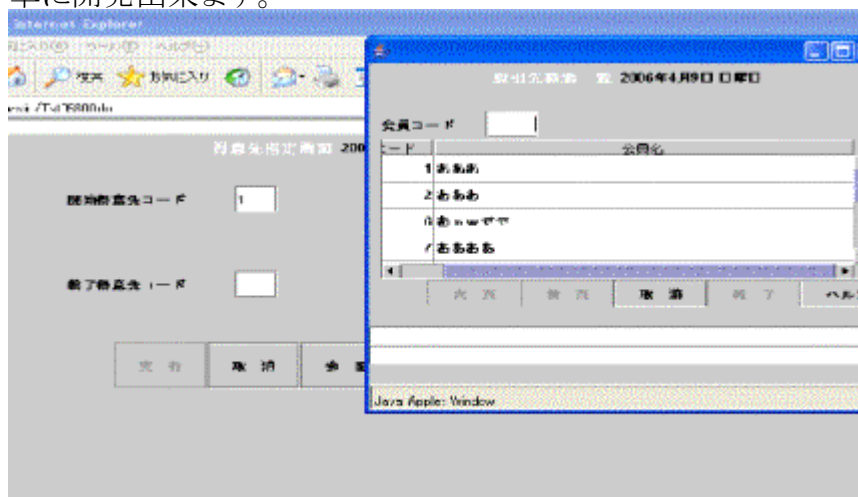
⑪クライアントにおいて、送信されたHTMLデータを、
Javaプログラムで読み出しできるように編集し、
処理をする。



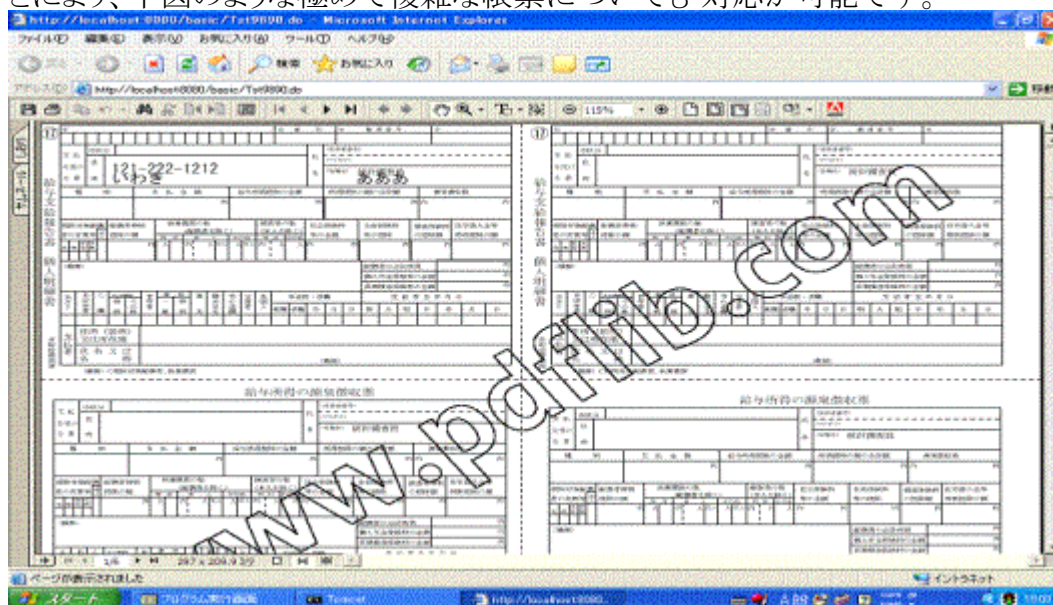
7 プログラムパターン

パターン及び定義の特徴として、以下のような点をあげることができます。

1. 出力はPDFLIBなるライブラリーを利用して、罫線の自動生成を行います。
2. 一覧表においては、合計項目及びキープレイク項目の指定により、簡単に合計処理ができます。
3. アプレットプログラムが極めて簡単な定義で、自動生成されます。
特に、アプレットの場合は伝票形式での複雑な入力、ウインドウを開いてのコード検索が簡単に開発出来ます。



4. アプレットプログラムの場合、項目前の条件により項目をスキップしたり、入力後にテーブルを読んでマスター項目を表示したり、項目チェックをすることが可能です。
5. MRDBの帳票定義機能を利用して帳票設計し、その座標値をWebロボに取り込みすることにより、下図のような極めて複雑な帳票についても対応が可能です。



パターン名	区分	パターン内容
登録	JSP	通常の日データ登録パターン。
保守	JSP	通常の修正、削除をするパターン。
メニュー	JSP	メニュー画面のパターン。
一覧閲覧	JSP	一覧表形式でのデータ閲覧パターン。
明細表示	JSP	明細形式でのデータ表示パターン。
ログイン	JSP	ID、パスワードの入力によりログインするパターン。
検索画面	JSP	検索条件を入力する画面のパターン。
座標登録	JSP	座標をマスターに登録するパターン。
マップリンク	JSP	座標をクリックして、その値を検索するパターン。
アップロード	JSP	写真、画像等をサーバーにアップロードするパターン。
Jtable 入力	Applet	Jテーブル形式(表形式)での登録パターン。
Jtable 保守	Applet	Jテーブル形式(表形式)での修正、削除パターン。
暦入力	JSP	カレンダー形式での入力をするためのパターン。
予定登録	JSP	予定表の登録パターン。
メール送信	JSP	メールの送信するパターン。
データ更新	JSP	データの更新処理をするパターン。
インポート	JSP	外部のテキストデータをデータベースにインポートするパターン。
エクスポート	JSP	データベースより外部テキストデータにエクスポートするパターン。
ファイル整理	JSP	データレコードのないファイル削除パターン。
PDF 一覧	JSP	PDF での一覧作表パターン。
PDF 詳細	JSP	PDF でのレコード詳細作表パターン。
Applet 通常登録	Applet	アプレットでの、通常での登録パターン。
Applet 通常保守	Applet	アプレットでの、通常での保守パターン。
Applet 一覧	Applet	アプレットでの、一覧表示パターン。
Applet 伝票登録	Applet	アプレットでの、1レコード複数行での伝票形式登録パターン。
Applet 伝票保守	Applet	アプレットでの、1レコード複数行での伝票形式保守パターン。
ファイル出力	JSP	ファイルへのレコード出力パターン。
レコード削除	JSP	レコードを削除するパターン。
一括更新処理	JSP	個別定義した更新系プログラムの一括処理パターン。
Applet 検索画面	Applet	アプレットでの検索画面タイプ。
MRDB 詳細 PDF	JSP	MRDB の詳細帳票定義利用パターン。

8 信頼性について

1. SQL インジェクション対策

JSP プログラムの場合

PreparedStatement 対応のプログラムを生成します。

アプレットプログラムの場合

roboX フォルダの `SQLInjectionEscape` クラスにおいて、' 及び `¥` などの SQL の特殊文字について `Escape` 処理するクラスを用意しています。

SQL 文の編集において文字項目の場合、下記のような処理をします。

```
sb.append(", "+SQLInjectionEscape(get$varh("jyusyo"))+"");
```

尚、使用するデータベースによってエスケープする文字が変わりますので、適宜、`SQLInjectionEscape` クラスを手直しします。

SQL インジェクションについては、警察庁のホームページをぜひご覧下さい。

(http://www.cyberpolice.go.jp/server/rd_env/PDF/20060330_SQLInjection.PDF)

2. 例外処理機構

Web ロボにおいては、各クラスにおいて適切な例外処理をするようにここがけております。

例外が発生した場合、roboX フォルダの `SystemExceptionHandler` において処理内容を定義するようにしております。

適宜、このクラスを修正して適切な処理をするようにします。

最終的に、JSP ホルダの `Exception.JSP` によりクライアントへメッセージ表示をしております。

適宜、この JSP を修正して適切なメッセージを表示するようにします。

3. フォームへの入力チェック

テーブル定義より `Validation.xml` を生成しますので、Java Script での入力チェックをしますが、更にアクションクラスにおいてもテーブル定義で定義したチェック内容を自動的にチェック文として自動生成します。

チェック項目の詳細はテーブル定義を参照してください。

尚、要望によりチェック項目は順次増やしていきたいと考えております。

4. 認証及び権限管理機構

① URL チェック

表題定義において、URL チェックを YES に設定することにより、メニューを通さないアクセスを拒否することが出来ます。

② パスワードチェック

表題定義において、ログインチェックを YES に設定することにより、ログインせずに、アクセスすることを拒否することが出来ます。

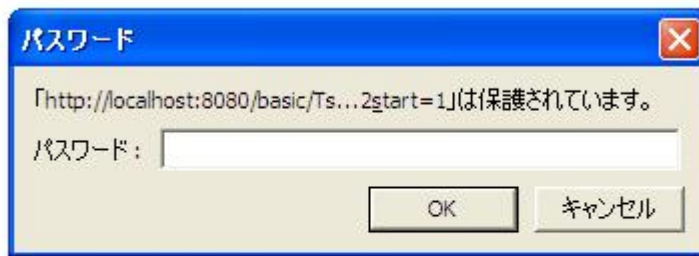
③ 権限管理テーブル

権限管理テーブルを利用することにより、権限ファイルへのプログラム名と権限 No の設定により、プログラム単位にアクセスの可否を設定することが出来ます。

④ PDF 帳票によるセキュリテシステム

PDF での作表の場合、権限ファイルへプログラム単位に、パスワード設定の有無と PDF の

パーミッションを設定することにより、下図のように自動的にパスワードと操作制限の設定することができます。



9 差込機能

以下の差込機能の利用により、極力プログラムの変更を回避することができます。

1. 関数による差込機構

@関数: クラス化しない Java 記述文を定義します。

これを、詳細項目定義の画面出力文、テーブル出力文アプレットの場合の入力前文、入力後文で使用します。

%関数: SQL 文を関数として定義し、共通部品化します。

JSP プログラムの表題定義の WHERE 句で使用します。

Applet プログラムにおいては、%関数を使用せずに表題定義及び詳細定義の WHERE 句ないし、表題定義の SQL 文詳細定義において、直接 SQL 文の編集をします。

%関数より、複合テーブルのテーブルビーン及び複合テーブルアクセスメソッドクラスを生成します。

2. Java クラスの登録によるimport文の生成

関数定義、画面出力文、テーブル出力文、入力前文、入力後文、SQL 文詳細定義、チェック文定義、レコード項目定義において、独自に用意したクラスを使用する場合、そのクラス名とimportフォルダ名を登録することにより、自動的にImport文を生成し、プログラムの変更を回避します。

3. SQL 文詳細定義

アプレットの場合、WHERE 句では定義しきれない SQL 文をここにおいて定義することができます。

4. チェック文定義

テーブル定義では出来ない相関チェック等の細かな入力チェックをここにおいて定義することができます。