
映像配信システムの開発事例

2012年2月10日

株式会社 **OKIソフトウェア**
情報ソリューション事業部
関西支社システム第一部 原田



本日の内容

1. OKIソフトウェアの紹介
2. インド発注への取り組み
3. 発注システムの説明
4. 発注内容の説明
5. 発注品質
6. UML活用の説明
7. UML適用のメリット・デメリット
8. 今後の取組について

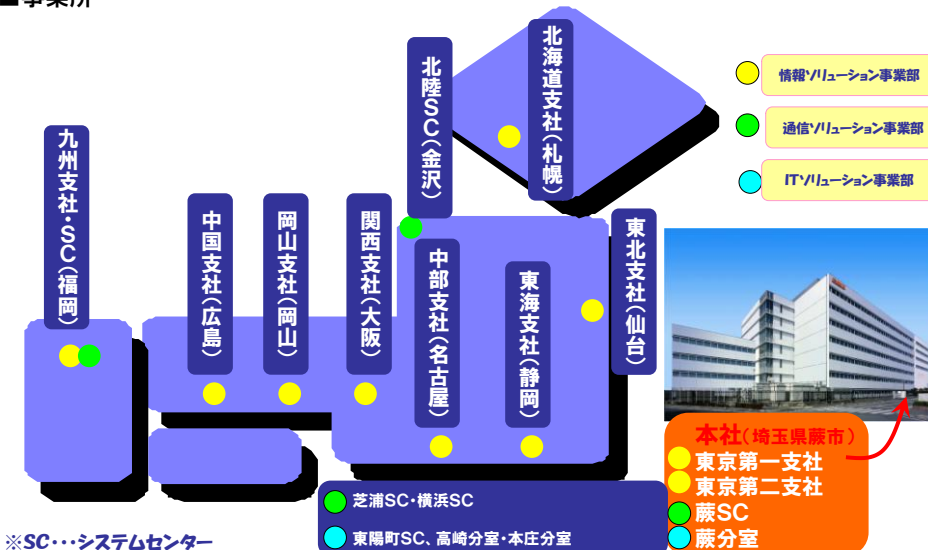
1. OKIソフトウェアの紹介

(<http://www.oki-osk.jp/>)





■会社概要

| | |
|------|--|
| 社名 | 株式会社OKIソフトウェア |
| 設立 | 1977年4月1日（沖通信システム、沖インフォテックとの合併：2010年10月1日） |
| 資本金 | 4億円（沖電気工業株式会社100%出資） |
| 事業内容 | <input type="checkbox"/> ソフトウェア／組込ソフトウェア開発・設計・製造・保守 <input type="checkbox"/> システム構築サービス <input type="checkbox"/> SI/ソリューションサービス(CTI・CRM、SCM、NMS他) <input type="checkbox"/> コンサルティング(システムソリューション、インフラ構築支援システム、ネットワークシステム、各種設計支援) <input type="checkbox"/> アウトソーシング(システム運用管理) <input type="checkbox"/> 情報機器販売 |
| 従業員数 | 1,202名(2011年3月31日時点) |
| 売上高 | 273億円(2010年度) |
| 認定 | OISO9001：1997年12月認定取得 Oプライバシーマーク：2005年12月認定取得 OISO27001：2007年12月認定取得 OISO14001：2003年3月認定取得(蕨地区のみ) |





■事業所



■OKIソフトウェア事業内容

| | | | | |
|------------|--|--|--|--|
| 沖電気事業 | 通信ソリューション事業 キャリア系NWシステム NGN, VoIP, 映像配信 エンタープライズ系NWシステム IPテレフォニー, eおと, CTI 交通・防災システム 防災無線, 航空管制 プリンティングソリューション カラーノンインパクトプリンタ | 情報ソリューション事業 金融システム 営業店, 事務系中, ATM 社会情報システム 人事給与, EA, バックオフィス 旅客運輸システム 予約, 発券, チェックイン, Web 交通システム ETC, VICS, 道路情報 ディフェンスシステム 水中音響(ソナー), 空中音響 | ATM-BankIT  | |
| | ITシステム事業 ERPソリューション MCFrame, Syteline B2B・Webソリューション Web-EDI, EAI | | | |
| eソリューション事業 | ミドルウェアソリューション Situator ネットワーク監視ソリューション NetSentinel, IP-VQMS | コールセンターソリューション CRM, CTI, IVR, 音声 組込みソフト開発 携帯, カーナビ, 地図情報 | 流通システム 受発注, カード決済/ポイント 自治体システム 総合窓口, 財務, 催告 | 社内IT 販売・会計ソリューション 販売・物流, 財務会計支援 経営・営業ソリューション 予算管理, 顧客管理 システム運用・保守 ITIL, 仮想環境, シェアード |
| | DISCOVERY 01  | enjoy.CRM II  | CItage  | |

■OKI事業開発例(旅客運輸)

| | | |
|--|--|---|
| ■駅利用システム | ■空港利用システム | |
|  | | |
| 顧客操作型指定券券売機  | 国際線用自動チェックイン機  | 国内空港 チェックインカウンタ  |

2. インド発注の取り組み

■発注ロードマップ

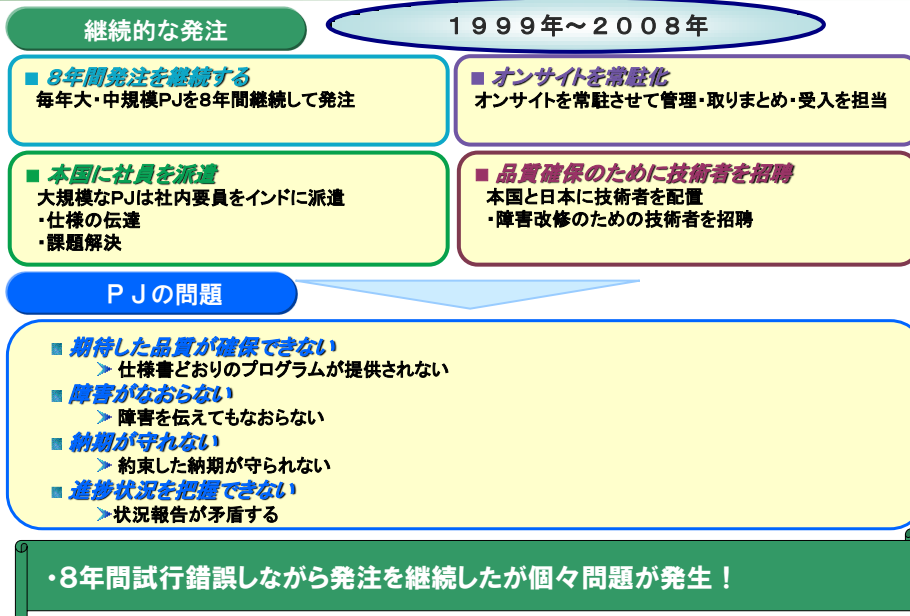
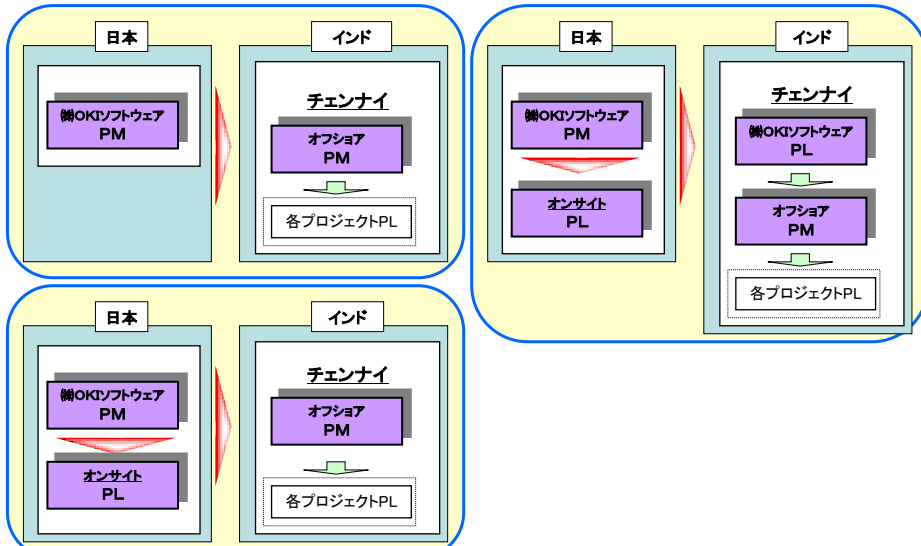
| | 99年度 | 00年度 | 01年度 | 02年度 | 03年度 | 04年度 | 05年度 | 06年度 | 07年度 | 08年度 | 09年度~ |
|---------------------|------|------|------|------|------|------|------|------|------|------|-------|
| 旅行業向け 物品発注システム | | → | | | | | | | | | |
| 航空会社向け ロビー端末システム | | | → | | | | | | | | |
| 航空会社向け WEB照会システム | | | | → | | | | | | | |
| 旅行業向け 発券端末システム | | | | | → | | | | | | |
| 航空会社向け 監視システム | | | | | | → | | | | | |
| 鉄道業向け 窓口端末システム | | | | | | | → | | | | |
| 旅行業向け 出張手配システム | | | | | | | | → | | | |
| 旅行業向け 代理店システム | | | | | | | | | | | → |



会社概要

| | |
|-------|--|
| 商号 | Nihon Technology Private Limited (NTPL) ISO・9001-2008 を取得済 |
| 初期出資額 | 150万ドル |
| 会長 | M.R.Ranganathan |
| CEO | Chellappa Sriram |
| 従業員数 | 100名 |
| 事業内容 | ソフトウェア開発、情報処理サービス、エンベデッド関連、 ソフトウェア商品等の開発・販売 |
| 本社 | インド、タミルナードゥ州、チェンナイ市 〒600 020 アダヤール ガンディナガー #7 2番クレセント パーク ロード TEL +91-44-43998000 (代表) |
| 東京営業所 | 〒107-0052東京都港区赤坂9-1-7赤坂RHビル 526 TEL : 03-6440-9121 FAX : 03- 6440-9122 |
| 大阪営業所 | 〒530-0043大阪市北区天満 2-2-21 ヒロビル6階 TEL 06-4801-8255 FAX 06-4801-8233 |

■ 2008年までの発注体制



発注側の問題

日本の協力会社でも継続しなければならない仕様書作成

- 仕様書の矛盾・不明点多
 - ・国内発注先は自分で設計しているため、不備を自力で補完できる。
しかし、海外の場合、誤った解釈をする場合が多い。
- 仕様書の書き方の展開不備
 - ・仕様書作成者のレベルにて左右され統一規格(書く内容等)にて記述されていない。

(特記事項)

- ・完全な仕様書は最後まで完成せず行間は説明にてフォローすればよいと考えている。
- ・抜けが多すぎてプログラムの処理抜けとなる。

仕様変更履歴無し仕様変更

- 仕様変更履歴無しで、更新された仕様書を送った場合、全ページ再英訳となり、相当の時間を必要とする。
- 分割にて国内・海外でI/Fがある場合等、変更内容をリアルタイムに伝達できていない。

国内スケジュールの遅延

- 発注工程と受入以降の作業線表を綿密にすりあわせていない。

受注側の問題

管理者の能力の問題

- 単に報告を取りまとめるだけの管理者(ほとんど実態を知らず)
- 計画能力無し(遅れに対して日程を詰めるだけの再計画)

作業要員コントロール不足

- 指示した後、細めなモニタ・コントロールを行わない。
 - ・各担当者が自分の裁量で作業を進める。

(特記事項)

- ・個々の技術者は、品質に対する考えが甘く、コントロール無しでは、品質の向上は期待できない。
- ・個々の技術者には、指摘されたら改修すればいいという意識が強い
- ・分業によりコーダーとテスターが異なる
- ・労務管理を行わない

各担当者の詳細な作業状況を把握できていない

- 担当から相談があれば、その解決のアクションを起こすが、特に相談がなければ、問題無しとみなす。
(各担当者ができるといえば、それはできるものとみなされる)

障害管理の不備

- 障害発生・改修の水平展開がない。各担当は、他人の担当部分に無関心
- 障害の発生→クローズまでが管理されていない

実施対策

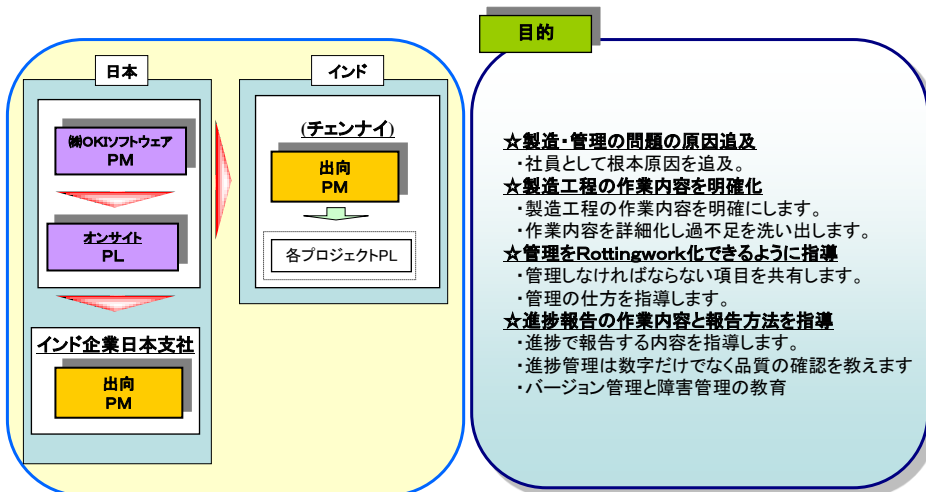
発注側対策

- **提示仕様の内容に対する責任は、発注元にあることを徹底**
→説明で補足する意識の廃絶
- **納入仕様書・システム仕様書の精度向上**
→国内の開発においても協力会社を入れ替えても製造できる仕様書の作成
- **仕様変更の管理と伝達方法の確立**
→仕様変更については時期・伝達の方法を検討し管理徹底する。

受注側対策

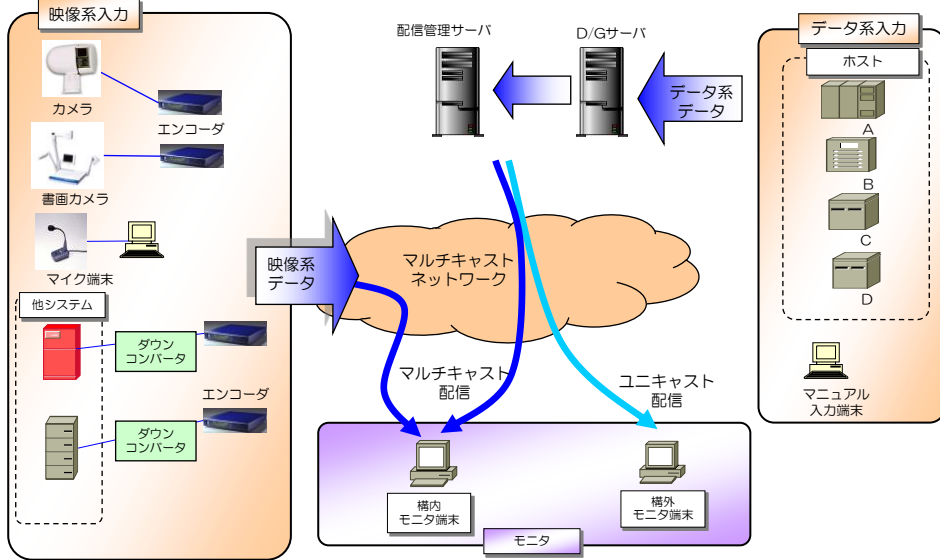
- **Managerの育成**
→社内のDM・PLクラスを教育しワンランクアップする
- **開発計画のプロジェクト管理を徹底**
→国内と同じ管理を実施する
- **報告に際しては数字だけで管理しない**
→進捗管理の徹底と報告形式の確立
- **分業の禁止**
→プログラム・モジュールの責任の明確化
- **障害管理・バージョン管理の徹底**
→抜けなく正確に実施

■2009年からの発注体制



3. 発注システムの説明

■システム構成



■特色

ツール・技法

- ◆ **UML**による設計
→ガイドラインの作成と展開
- ◆ **ClearCase**によるバージョン管理
→人的ミスの削減
- ◆ **eclipse**を使用したコーディング
→コーディング規約のツール組み込みによりコーディングミスの防止

システムの特徴

- ◆ **配信機能**
→Java Message Service (JMS)を使用しPush型配信を利用した効率的な配信
- ◆ **モニタリング機能**
→ブラウザを利用した映像系とデータ系の配信情報の融合
- ◆ **MIG通信管理機能**
→配信データ量の増大に対する対策
- ◆ **VisualCast^{SS}**
→映像配信

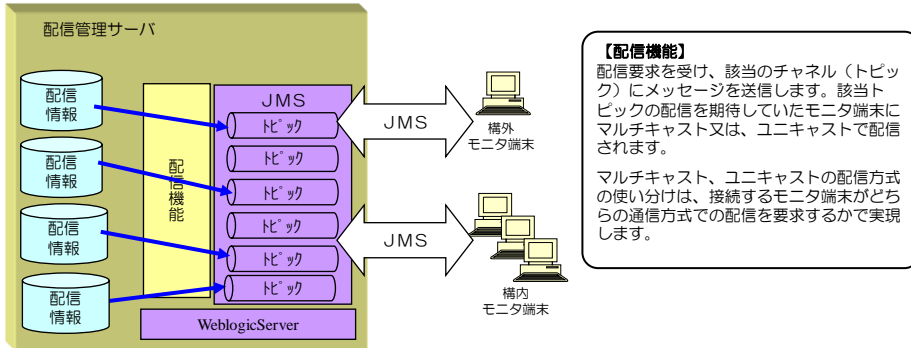
■配信機能

構内モニタ端末はIPマルチキャスト、構外モニタ端末はユニキャストでのPush型配信を実現します。マルチキャスト配信はJava Message Service (JMS)を使用し効率的な配信と致します。

【配信方式の考え方】

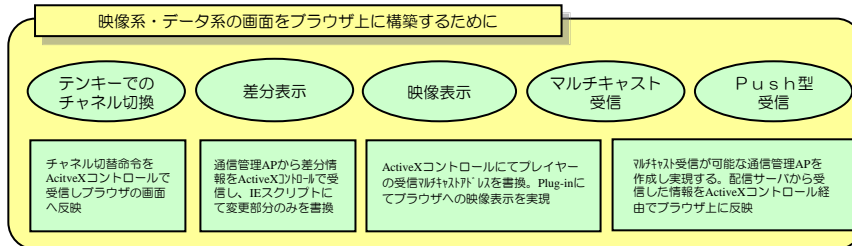
同一チャンネルを多数のモニタ端末でモニタする形態である本システムでは、マルチキャスト配信がサーバ負荷、ネットワーク負荷の観点から有効と考えます。従って、空港内のモニタ端末に対しては、マルチキャスト配信をご提案します。

また、マルチキャスト通信が不可のネットワークに接続されている空港外モニタ端末へは、ユニキャスト方式で配信します。

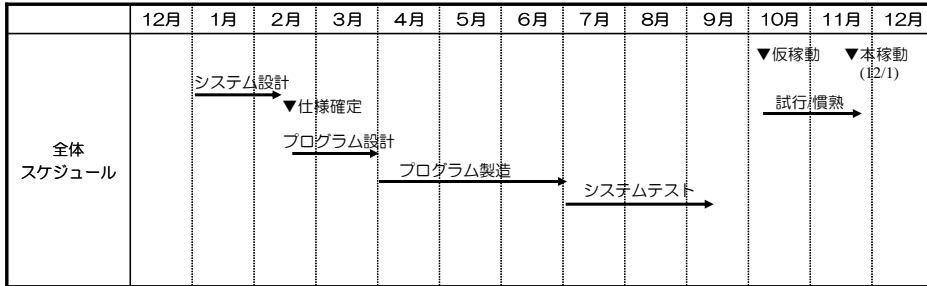


■モニタリング機能

異なるソリューションである映像系とデータ系の配信情報を違和感のない操作性でのモニタリングを実現するために、映像系・データ系ともにブラウザにて表示します。

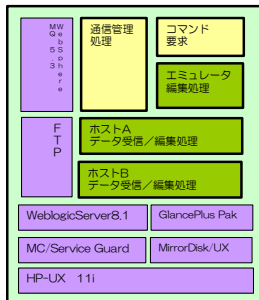


■スケジュール

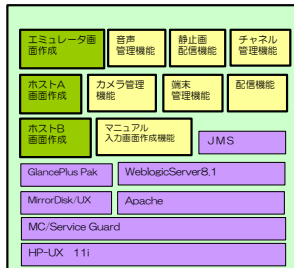


4. 発注内容の説明

【D/Gサーバ】



【配信管理サーバ】



| No | 処理名 | 概要 |
|----|----------------|---|
| 1 | エミュレータ編集処理 | ・受信したデータストリームを解析し、全画面及び差分情報を取り込み配信サーバに通知する。 ・最新データを保持する。 |
| 2 | ホストAデータ受信/解析処理 | ・30秒毎にFTPで送信されるメイクアップ割付ファイルを基に、画面単位に分割し、配信管理サーバに通知する。 ・最新データを保持する。 |
| 3 | ホストBデータ受信/解析処理 | ・1分毎にFTPで送信される情報ファイルを基に、画面単位で分割し、配信管理サーバに通知する。 ・最新データを保持する。 |
| 4 | エミュレータ画面作成機能 | ・D/Gサーバからエミュレータの全画面情報及び、差分画面情報を受信する。 ・全画面情報の場合、HTMLで画面情報を作成し、XML形式で配信機能に通知する。 ・差分情報を受信した場合、一次的に蓄積し、2秒間隔で差分画面情報を行単位でHTML変換し、XML形式で配信機能に通知する。 ・30秒間隔に全画面データをモニタ端末へ送信しデータ補正を行う。 |
| 5 | ホストA画面作成機能 | D/GサーバからホストA画面情報を受信し、HTMLで画面を作成し、XML形式で配信機能に通知する。モニタ端末へ30秒間隔で送信する。データは、全画面分データとする。 |
| 6 | ホストB画面作成機能 | D/GサーバからホストB画面情報を受信し、HTMLで画面を作成し、XML形式で配信機能に通知する。モニタ端末へ1分間隔で送信する。データは、全画面データとする。 |

D/Gサーバ・配信管理サーバ・モニタ端末の開発の全体の1/3を発注

5. 発注品質の説明

品質

全体として品質は良かった。

障害区分において、仕様ミスと設計ミスが全体の約40%を占めている。

これは、PD完了後にエラーコードの体系を変更したことによるものであり内部設計に誤りがあったわけではない。

また、ロジック抜けやリターンコードの判定誤り／漏れのCDミスでも約40%の割合となっている。

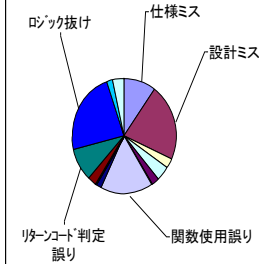
傾向としては例外処理や定義ファイル異常時の処理に障害が集中していた。

その他のCDミスの要因としては、HP-UXの仕様の考慮漏れがあった。

UMLを使用したプログラム仕様書の品質としては満足できる品質を確保できた。

例外処理(エクセプション)の記述はアクティビティ図で記載したが、コーダの技術力の問題で障害が多かった。

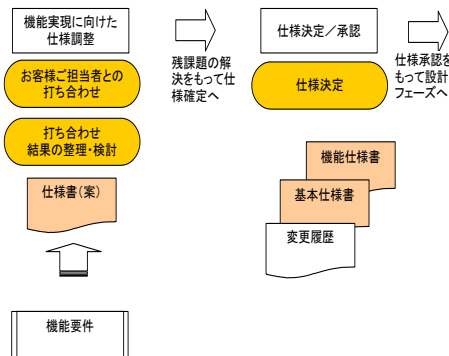
障害分布



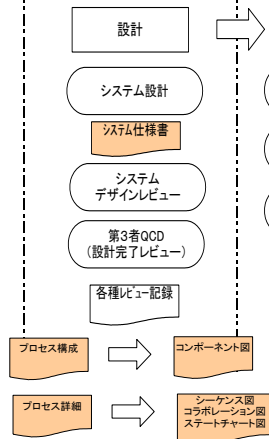
- 仕様ミス
- 設計ミス
- ループ処理誤り
- データ宣言誤り
- 変数使用誤り
- 配列処理誤り
- 算術演算誤り
- 関数使用誤り
- 入出力誤り
- 文字列操作誤り
- 終了処理誤り
- I/Fフォーマット誤り
- I/Fシーケンス誤り
- リターンコード判定漏れ
- リターンコード判定誤り
- ロジック抜け
- データ設定ミス
- データ参照ミス

6. UML活用の説明

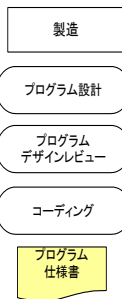
【仕様確定PHASE】



【設計PHASE】



【製造PHASE】



UML作成時の方針

機能仕様書からのユースケース作成が一番難しく、適切なユースケース図を作成することにより、より良いシステム設計を行うことができる。

ユースケース：システムの利用者(外部システムを含む)をアクターとして、各アクターごとにシステムがどのような機能やサービスを提供するべきかを識別したもの。

ユースケースを活用したUML作成手順を下記に示す。

- ①該当する制御の機能仕様書よりユースケース図を作成する。
→UMLを作成するにあたり、ユースケース図が無ければ設計が不可能といえる。
Roseでは各ユースケースごとにUML図を作成するとRoseのフォルダ構成が機能と対応し、ドキュメント作成も容易である。
(4. Rose利用方法について参照)
- ②①で作成したユースケース図より1ユースケース単位で概要・シナリオ又はイベントフローを作成する。
 - (a) 概要…一つのユースケースに対するユースケースの説明
 - (b) シナリオ…一つのユースケースの流れを具体的に記述した文章。
シナリオ内で記述する名称・金額・数量などは実際の値を使用し、現実処理が行われたように記述する。
 - (c) イベントフロー…一つのユースケースにおける全ての流れを、汎用的に記述する。
 - ・事前条件…ユースケースを開始する為の制約や条件
 - ・事後条件…ユースケースが終了した後の制約や条件
 - ・基本フロー…イベントフローの最も一般的な流れ
 - ・代替フロー…基本フローより頻度が少ないが正常な流れ
 - ・例外フロー…正常終了しない流れ
- ③②にて作成したシナリオ及びイベントフローよりユースケース単位でクラス図を作成する。
- ④②にて作成したシナリオ及びイベントフローと③にて作成したクラスを利用してユースケースごとにシーケンス図及びコラボレーション図を作成する。
→シーケンス図はシナリオ及びイベントフローの全てのパターンを網羅する必要がある。
モデリング時にパターンを少なく考慮するか、コラボレーション図を利用する。

UML作成時の注意事項(From UML推進チーム)
 ・UMLをSD時より利用するとシステム全体のユースケース図よりブレイクダウンした設計をおこなうことができる。(推奨)
 ・設計者間でコミュニケーションを行い、データ中心にシステム設計を行う必要がある。
 ・共通データを扱う箇所に関してはRoseファイルを統一して設計を行うとデータClassのずれを防ぐことができる。
 →Roseのフォルダ構成を明確にし、役割分担をする必要がある。

プログラム仕様書 (UML表記法) ガイドライン

本仕様書に、UML (Rational Rose使用) を用いてプログラム仕様書を作成する際のガイドラインを記す。

1. ドキュメントの様式について

- ①ドキュメントは制御単位で作成する。
- ②表紙、及び変更履歴はSWSにて指定されたものを付加すること。
- ③下記の目次内で指定しているUML図・記述項目は必須項目とする。
- ④SoDaにてドキュメントを出力する場合は「4. Roseの利用方法について」に従うこと。

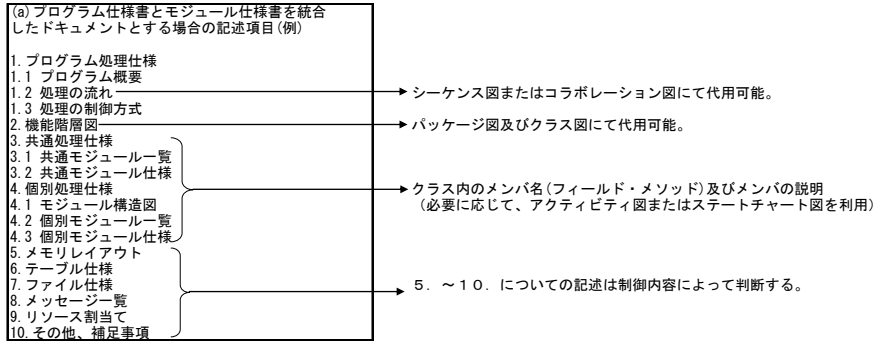
| UML版目次 | |
|--|--|
| 1. プログラム処理仕様 | |
| 1.1 概要 | |
| 1.2 ユースケース | → ①制御全体のユースケース図 * ①はSoDa標準テンプレート rup business use case model survey.docにて一括出力可能 |
| 2. 個別処理仕様 | |
| 2.1 ユースケース名(1) | → ②～⑤はユースケースごとに記述する事とする。 ②ユースケースの概要 ③ユースケースのシナリオ又はイベントフロー (但し、システム仕様書にて既に記載してある場合は参照場所を記述するのみでよい。記述内容は3. ②参照) ④クラス図 ⑤シーケンス図又はコラボレーション図 * ②④⑤はSoDa標準テンプレート rup use case realization report.docにて一括出力可能 |
| 2.2 ユースケース名(2) | |
| 2.3 ユースケース名(3) | |
| 3. モジュール仕様 | |
| 4. (エラーメッセージ一覧・その他補足事項) * 4以降は制御内容により記述が必要である場合に利用する。 | → ⑥全クラス内のメンバ名(フィールド・メソッド)及びメンバの説明 (但し、メソッドの処理が複雑で文章だけでは説明が不十分になる可能性がある場合は、必要に応じて、アクティビティ図またはステートチャート図を利用すること。) * ⑥はSoDa標準テンプレート classesattrspstable.docにて一括出力可能 |

2. 目的

- ①RationalRoseを利用する場合にプログラム仕様書としての最低限必要な記述項目を明確にする。
- ②プログラム仕様書として記載内容に不足がない事をチェックする。
- ③UMLの知識があるものであれば、仕様書の内容を理解できること。
- ④Roseにて入力した内容をSoDaWordを利用したドキュメント作成を目標とする。
- ⑤Roseにて入力した項目をSoDaで出力するだけでは、十分な仕様書を作成することはできないのでSoDaにて出力可能な項目と出力不可である項目を本仕様書に明記する。

* 従来 (SWS) のプログラム仕様書の記述内容をUML図にて記述する場合に代用可能な項目を下記に示す。

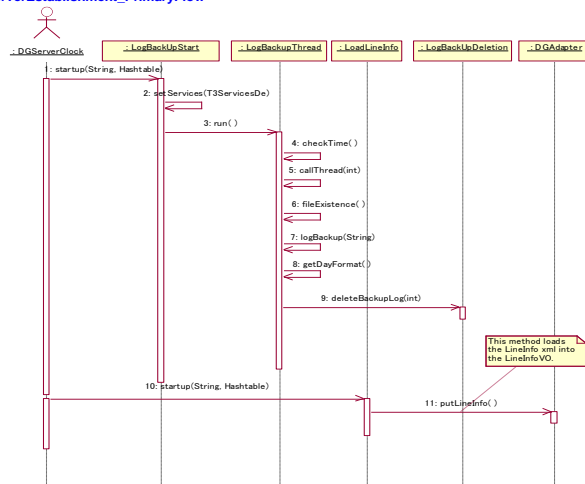
↓ SWS プログラム仕様書、モジュール仕様書作成の手引きより



■ サンプル

1. Interaction Diagrams

1.1 DGServerEstablishment_PrimaryFlow



7. UML適用のメリット・デメリット

メリット

★開発に携わる人たちが共通の言葉を持ち、コミュニケーションができる

・日本語・英語・タミール語などの自然言語が持つあいまいさや冗長性を排除できる。

★設計の再利用が非常にやりやすい

・UMLは環境に非依存なモデリング言語のため、作成した設計モデルは、いろいろな言語や環境で利用できる。

★造るシステムにしたがってダイアグラム選択しわかりやすい仕様書を作れる

・すべてのダイアグラムを使うのではなく、システムの特性・要件に従って機能を実装する時に必要なダイアグラムを選択して使用できる(柔軟性がある)

デメリット

★要求定義では活用率が低い

・まだまだ仕様の決定権を持っている人には浸透していない。
・UMLを使用できる技術者が少ない。

★どうしても曖昧な部分ができてしまう。

・アクティビティ図、ユースケース図で、動的情報は抑えられるが、クラス図は静的情報なのでクラスの中に、そのメソッドが所属する理由が表現できないためにクラス図が、どうしてもこの形になるのか？という部分が曖昧になる。
・何秒以内に結果が返ってこなければならぬ、というような仕様の記述手段がない。
・画面遷移は、UMLで書くとうわかりにくくなる。
・データベースの設計はE-R図を使った方がいい場合もある。

★既存システムの改修には向かない

・レガシーな既存の仕様書をすべてUMLにはできない。

8. 今後の取り組みについて

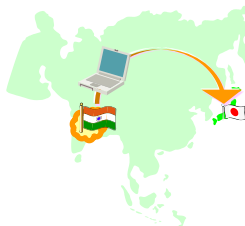
■ 生産性向上のためのインド発注

日本テクノロジー活用の利点

- コスト削減
- 先端技術者不足の解消
- 欧米顧客とのビジネスで増われたノウハウの活用

インドオフショアの強み

- 組込みソフト
- 通信制御ソフト
- ERP、CRMのインプリメンテーション
- 企業向けソフトのマイグレーション



■ 業務アプリケーションを開発する為には

仕様書の精度向上が不可欠！

施策1

日本語で書く部分の改善
※リダービリティ
※英訳しやすい文章の書き方
⇒納入仕様書・システム仕様書

施策2

システム仕様書の改善
※UML利用にて仕様書精度向上
※UMLで表現できない部分の書式の制定

施策3

UML技術者の育成
※50%から80%へ

発注
施策

