3L-8

複数ユーザシステムの並行開発におけるモジュール管理手法について

磯貝恭子　西山みどり　坂上安春　清兼幸雄　廣瀬正明　井上進

富士通株式会社

1．はじめに

本稿は，電子交換ソフトウェアの並行開発を支援するモジュール管理手法とその支援システムについて述べる。

電子交換システムは，大規模リアルタイムシステムのひとつであり，そのシステムファイルは比較的，密に結合したモジュールにより構成されている。そこで従来より，モジュール構成管理や統一的な版数管理を支援する開発環境の構築を行ってきた。(1)

情報化社会の進展に伴い通信サービスは一層拡大傾向にあり，顧客ニーズの多様化に迅速に対応していくことが不可欠である。このため，顧客毎に要求される機能を同時に並行開発する支援環境の構築が急務となっている。

2．並行開発におけるモジュール管理と問題点

交換ソフトウェアの開発は，国外／国内／私企業網などの複数の顧客（ユーザ）に対してタイムリに各サービスを提供するために，開発フェーズを分割し各ユーザ毎のシステムを並行して開発している（図1）。このため各フェーズ毎のモジュール開発管理作業と各ユーザ毎のシステムファイルの作成作業が複雑になっている。

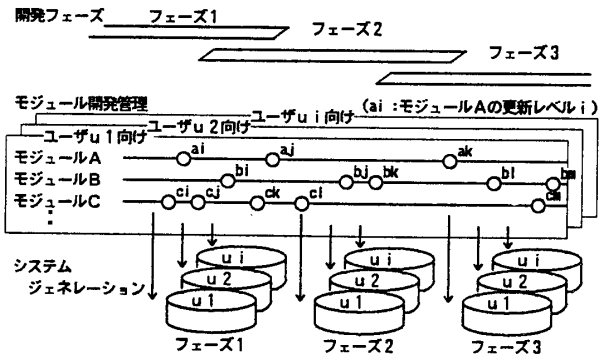

図1　並行開発の形態

(1)　モジュール開発管理

フェーズ毎に機能が追加されるのに伴なって，各モジュールが変更される。ある時点でバグの検出や機能変更が発生すると，それぞれのフェーズに対応したレベルのモジュールに修正が必要となる。また，各ユーザ毎に異なった機能の開発をしなければならない場合，元のモジュールをコピーし，別モジュールとしてそれぞれ機能追加が行われる（これをモジュール分岐と呼ぶことにする）。

モジュール分岐の問題は，変更／修正手続きが二重管理となることである。また，分岐したあとで元のひとつのモジュールに再統合することは，時間が経過するほど（変更・修正箇所が多くなるために）困難である。したがって，モジュール分岐をなるべく避けるために部品化を図り，共通モジュールと個別モジュールに細分化して整理することが行われる。しかし，細分化によりモジュール数や種類が増えるのでモジュールの構成管理が一層複雑になる。

(2)　システムジェネレーション

各ユーザ毎に機能の異なるモジュールをすべて盛り込み，方式最大の標準システムファイルを全ユーザに提供することは，現状の実装メモリ量を考慮すると現実的ではない。このため，いくつかの標準システムを設定し，ユーザ毎に必要なモジュールだけを選択してシステムファイルを作成する必要がある。

また，交換システムは長期運用や非常時の無停止動作を保証する必要性からハードウェア条件やＯＳの制約が多くあり，ユーザ毎に行うシステムジェネレーションは非常に複雑で，少数のエキスパートにしか出来ない作業となっている。

3．解決へのアプローチと実現方法

3．1　基本的な考え方

並行開発に伴う問題点は，裏返すと同時期に単フェーズ，単ユーザの開発しか行わなければ発生しない問題である。そこで2.で示した問題点の解決へのアプローチとして，開発者にとってはあたかも目的のフェーズ／ユーザ専用に準備されたものであるかに見える環境の構築を目指した。以下に具体的な実現方法を示す。

3．2　フェーズ毎のモジュール管理手法

(1)　モジュール状態管理

モジュール開発管理において以下の３つの状態の概念を取り入れる。

・機能版数

本来同一モジュールでありながら，モジュール分岐などにより分割されたモジュールの版数を示す。

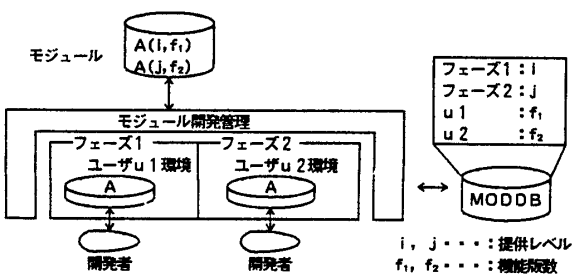

図2　モジュール開発管理

"A METHOD OF MODULE MANAGEMENT IN PARALLEL DEVELOPMENT FOR TELECOMMUNICATIONS SOFTWARE",
Kyoko ISOGAI, *Midori* NISHIYAMA, *Yasuharu* SAKAUE, *Yukio* KIYOKANE, *Masaaki* HIROSE, and *Susumu* INOUE, FUJITSU LIMITED

・提供レベル

該フェーズ内で，モジュールが他のモジュールと結合して問題ない状態に達した時の更新レベルを示す。

・凍結

フェーズにおけるモジュールの開発完了を示す。

各モジュールは，機能版数とフェーズ毎の提供レベルで表し，すべてモジュール管理データベース（ＭＯＤＤＢ）で管理する。この管理手法により，各フェーズのあるユーザ向けの開発においては，見掛け上，モジュールは一つのものとして扱うことができるため，開発者はエディットやコンパイル時に分岐したモジュールである事を意識する必要はない（図２）。

(2) 変更管理

モジュールにバグが検出された場合，凍結モジュールや分岐先のモジュールにも同様のバグが存在する可能性がある。障害を含んでいる可能性のある分岐モジュールに通知するだけでなく，図３に示す様に障害修正管理システム[2]と連携し，実際に修正が施されたモジュールとその修正内容の管理を行うことができるようにした。

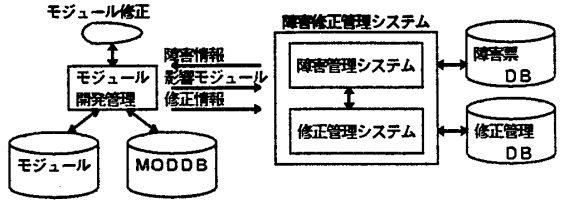

図３　障害修正管理システムとの連携

３．３　管理者のためのモジュール管理機能の提供

モジュール管理データベースは，機能版数，提供レベルの他に，開発凍結状況，モジュールの属するサービス／プロトコル等の情報を管理している。

また，モジュールの登録・変更は，すべて管理者の承認した管理番号（モジュール登録票番号，障害票番号など）を入力しなければできないようになっており，これらの情報はＭＯＤＤＢに自動的に収集される。

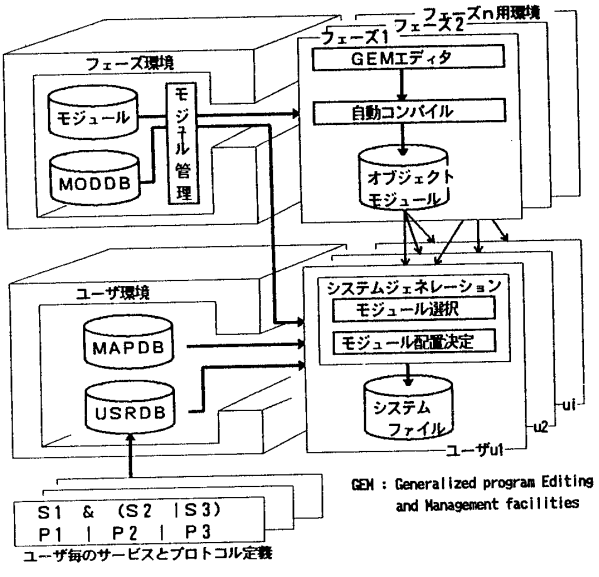

図４　システムジェネレーションの自動化

管理者はＭＯＤＤＢから得られる情報を使って以下に示す作業を行うことができる。

・モジュール開発進捗状況把握

（フェーズ毎の進捗，サービス／プロトコル毎の進捗，など）

・モジュール分岐管理

（モジュール分岐状況，分岐後のモジュール凍結／廃棄可否の判断，など）

３．４　システムジェネレーションの自動化

サービス／プロトコル／ハードウェア条件等によって選択が必要なモジュールは，各モジュール毎に属性を持たせ，ユーザのシステム要件の定義情報（ＵＳＲＤＢ）と照らし合わせることにより自動選択する。この後，メモリ割付け規則（ＭＡＰＤＢ）に従ってメモリ上の配置を決めてシステムファイルを生成する（図４）。ＭＡＰＤＢのルールは約５０種あり，ハードウェア条件などの割り付け規則と経験的な規則とから構成されている。システムジェネレーションの自動化により，ユーザ毎の要求条件に合わせてカスタマイズすることが容易となった。

４．並行開発環境の構成

図５に並行開発支援環境のシステム構成の概要を示す

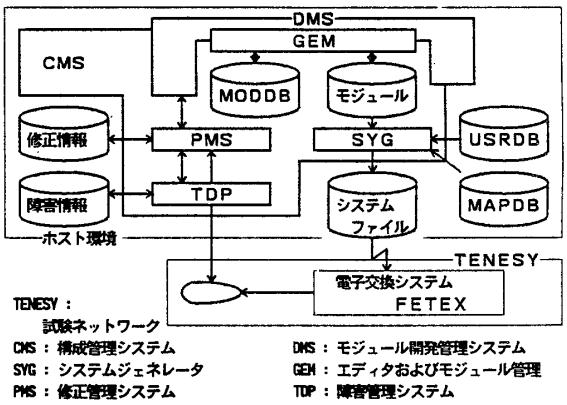

図５　並行開発支援環境のシステム構成

５．おわりに

障害修正，あるいは機能追加のないシステムは，既に機能的に寿命が尽きたシステムであると言える。このことはシステムが”アクティブ”である限り，機能追加が繰り返し行われるということであり，複数システムの同時並行開発と機能追加を前提とした開発支援環境は益々重要であると考える。

現在は，機能追加がモジュール構成に及ぼす変化や修正モジュールの影響範囲を評価する支援システムを開発中であり，さらに信頼性の高いソフトウェア開発のための支援環境構築を目指す予定である。

(参考文献)

(1) 浜，他「交換ソフトウェアにおける版数管理の一手法」，昭和62年電子通信学会全国大会

(2) 山田，他「大規模ソフトウェアの開発保守支援のための付加価値情報システムの要件」，昭和60年後期情報処理学会

(3) 児玉，他「電子交換ソフトウェア開発における試験支援ネットワーク」，1985年 4月交換研究会

(4) Rochkind, J., " The Source Code Control System ", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, December, 1975