ＣＳＳ統合開発環境(3)
－プラットフォーム－

4U－3

末宗　英利*　吉野　松樹**　髙舘　公人**
*日立ソフトウェアエンジニアリング(株)　**(株)日立製作所

１．はじめに

システム開発の設計からテストに至るまでを統合的に支援する、ＣＳＳ方式による分散開発環境を構築し[1]、その中で、ＣＡＳＥツールの共通基盤となる、プラットフォームを開発した。プラットフォームはツールの起動制御、仕様書／仕様書間の関連管理、ツール間のメッセージ通信を行う。リポジトリとツールの間に位置して、リポジトリ内の情報資源に、仕様書というセマンティックを付加してツールに提供する役目と、サーバマシン上のリポジトリに存在する仕様書をツールに受け渡すことで、ツールからは仕様書の存在場所を透過にする役目と、仕様書に対するアクセスの排他制御／機密保護の役目を果たしている。これによって、ツールの開発工数を削減し、利用者がプラットフォーム上にツールを構築することを容易にした。本報告では、プラットフォームの構成と、その特徴について述べる。

２．システム構成

プラットフォームは①メッセージサーバ②仕様書管理サーバ③排他サーバ④CASEマネージャー⑤組み込み支援からなる(図１)。

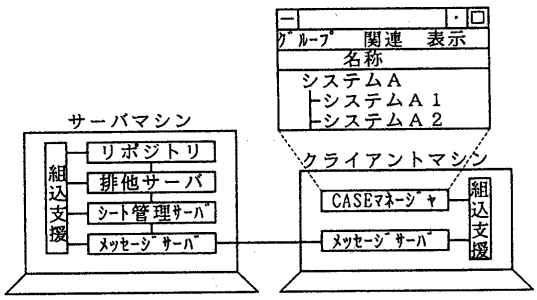

図１　システム構成図

①メッセージサーバは、ツールを起動し起動／終了を通知するイベントを発生させる機能と、ツール間の情報交換を行う機能を有する。②仕様書管理サーバは、仕様書とそのグループの操作機能、仕様書間の関連の操作機能、仕様書間の関連に基づく変更通知機能を有する。③排他サーバは、仕様書を含む資源の使用要求、開放要求を受付け、資源の使用状況を管理する。④CASEマネジャーは、仕様書の操作、ツールの起動等のユーザインタフェースを提供する。⑤組み込み支援は、ツールの組み込みを行い、ツールの起動条件等の組み込み情報を管理する。

３．オープン化

組み込み支援が提供する形式でツールの起動条件を記述したツールは、プラットフォームから起動することが可能である。ツールは、仕様書のクラスに対するメソッドとしてプラットフォームに登録される。ツールの組み込みに際して、新たな仕様書のクラスが必要なときは、仕様書のクラスを登録する。ツールは、ロードプログラムと、ロードプログラムを呼び出すシェルプログラムの２つを提供する。利用者が、ツールの起動前、起動後に必ず実行したい処理がある場合には、シェルプログラムを書き換えることで実現可能とした。

さらに、プラットフォームの機能を２章で述べた５つのサブシステムに分割することによって、利用者が独自のCASEマネジャーをプラットフォーム上に構築することも可能とした。

４．統合化

ツールの統合には、ユーザインタフェース(UI)統合、データ統合、制御統合、プロセス統合の４つがある[2]。

(1)ＵＩ統合：ツールの外観、動作、メタファが似ている。
(2)データ統合：情報の一部が変更されても、全体として整合性が保たれている。
(3)制御統合：ツール間で機能が共有できる。
(4)プロセス統合：定義したプロセスを支援するために、ツールが効果的に相互作用する。

プラットフォームはこの中のデータ／制御／ＵＩ統合の役目を担う。ＵＩ統合においては、全ツールがMotifを用い、規定したスタイルガイドを守ると共に、プラットフォームは共通ＵＩルーチンを提供した。プロセス統合はプラットフォーム上の作業誘導システムが実現した[3]。

４．１　データ統合

データ統合は、リポジトリの役目でもあり、プラットフォームでは、非永続的なデータと、データを仕様書として見た時の整合性維持機能を提供している。

An Integrated Software Development Environment on Client Server System(3) -Platform-;
Hidetoshi SUEMUNE*, Matsuki YOSHINO**, Masato TAKADACHI**,
*Hitachi Software Engineering Co., Ltd., **Hitachi, Ltd.

非永続的なデータの整合性を維持するために、ツール間の通信機能を提供している。本機能をツールが利用することによって、あるツールのデータを別のツールへ画面上でカット＆ペーストしたり、あるツールのデータを変更した際に、同時に起動してる別のツールの関連するデータを画面上で自動的に変更することが可能になる。

仕様書間の整合性を維持するために、仕様書間を関連付ける。利用者は、仕様書を変更した時に、関連する仕様書に変更通知を送付することができる。変更通知には、変更した仕様書名、利用者が記述した変更内容、変更日付が記される。変更通知を受信した利用者は、担当した仕様書を修正し、変更通知に対する対策状況を登録する。管理者は、対処状況を監視することができる(図２)。

仕様書間の関連は、ある仕様書から別な仕様書を自動生成した際、ある仕様書から別な仕様書を呼び出した際に、自動的に付けられる。

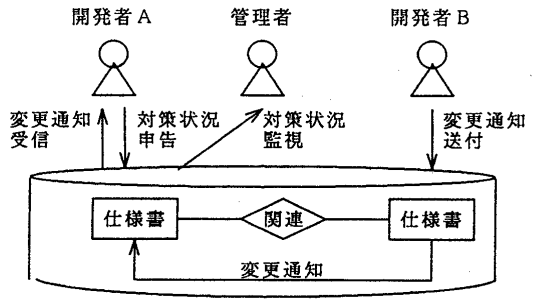

図２　変更通知受発信

４．２　制御統合

ツール間でイベントを通信し合う機能を実現した。これによって、あるツールから別なツールを起動することが可能になり、仕様書を編集中に関連する別な仕様書を編集可能とした。さらに、仕様書を削除した結果を、別のワークステーションの仕様書階層の表示結果に反映することができる。

５．分散環境

仕様書管理サーバとメッセージサーバは、ツールを実行しているワークステーションと異なるワークステーション上の仕様書やツールに対する操作／通信を可能とする。

５．１　仕様書管理

リポジトリは、情報をＥＲモデルで管理し、情報の意味には関知しない。プラットフォームはツールが扱う仕様書をリポジトリで管理する役目と、サーバワークステーション上のリポジトリに存在する仕様書をツールに受け渡すことで、ユーザからは仕様書の存在場所を透過にする役目を果たしている。各ツールは仕様書管理サーバに接続することで、仕様書を操作できる。

５．２　メッセージ通信

通信には、複数のプログラムで共有する資源を管理するサーバ（例：リポジトリ）に対して、サーバの持つ機能を呼び出すリクエストと、プログラム内で発生する事象を不特定多数のプログラムに報告するイベントの２種類がある。各ツールは、メッセージサーバに接続し、受信したいリクエスト、イベントをメッセージサーバに登録することでリクエスト、イベントを受信できる。これによって、リモートプロシジャコールの機能と合わせて用いることにより、プログラム自動生成など処理時間を要するバッチ的なツール機能を、処理速度の速いサーバワークステーション上で実行することが可能になる。

５．３．排他制御

ある利用者がファイルの編集中に書き込みロックを設定すると、他の利用者はファイルの内容を参照することもできなくなってしまう。ロックを設定しないと、２人が同時に同じファイルに編集可能となってしまい、先に格納した編集結果が失われてしまう。２人同時の編集を禁止し、ある人が編集中のファイルを参照することを可能とするために、以下に示す４種類の排他モードを設けて、仕様書のアクセスを制御した。

	使用状態			
使用要求	READ	排他READ	WRITE	排他WRITE
READ	○	○	○	×
排他READ	○	○	×	×
WRITE	○	×	×	×
排他WRITE	×	×	×	×

仕様書の編集中にＷＲＩＴＥモードを設定することにより、同時編集は不可能だが、ＲＥＡＤモードで参照することはできる。さらに、排他ＷＲＩＴＥモードを設定することで、仕様書の参照も防ぐことができる。

６．おわりに

ツールの共通基盤となるプラットフォームについて構成要素と特徴について述べた。環境内にツールを組み込むことを容易化することでオープン化し、データ／制御／ＵＩ統合を実現することでツール間の統合度を高めた。ユーザから仕様書や他のツールの存在場所を透過にすることで、ＣＳＳ環境でのツールの稼働を可能にした。

８．参考文献

[1] 田村他：ＣＳＳ統合開発環境(1)－概要－，第４５回情報処理学会全国大会論文集

[2] Ian Thomas他：Definition of Tool Integration for Environments, IEEE Software Mar.1992 pp.29-35

[3] 吉川他：ＣＳＳ統合開発環境(1)－作業誘導－，第４５回情報処理学会全国大会論文集