ＣＳＳ統合開発環境(4)
−作業誘導−

4U−4

吉川　彰弘*　　高舘　公人*　　穂垣　博文**
*(株)日立製作所　**日立ソフトウェアエンジニアリング(株)

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た[1]。

分散開発においては複数の作業者がいかに協調して開発を進められるかが問題となる。我々は作業の手順を登録し、この手順に従って作業を誘導することにより、作業の標準化による品質向上や協調開発の支援などを行うシステムを開発した。本稿ではこのシステムの構成と作業の誘導方法、及びその実現法について述べる。

２．システム構成

本システムは①作業手順定義②作業誘導③スクリプト生成からなる(図１)。

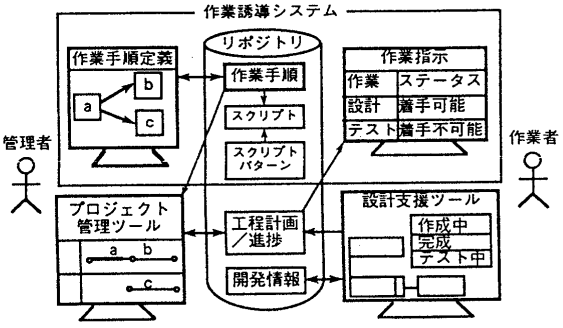

図１　作業誘導システムの構成

作業手順とは作業の内容や順序などをいう。手順で示される個々の作業に具体的な仕様書の名前や担当者、開発期間などを割当てる工程計画はプロジェクト管理ツール[2]で支援する。

作業誘導では作業手順の定義に従ってツールの起動などを行う。その実現法には①作業手順を参照しながらツール起動を制御する②作業誘導を行うスクリプトを自動生成しこのスクリプトを実行する、の２通りがある。①はツール起動の制御方法が固定的なので拡張性や自由度に欠ける。②は一旦スクリプトを生成するので①よりもシステム構成が複雑となる反面、生成されたスクリプトを修正できるという自由度がある。さらに本システムではスクリプトを生成するプログラムも変更可能としたので将来の拡張性にも富む。これらから②の方式を採用した。

３．作業手順定義

作業手順定義では以下を定義する。

(1) 作業の番号、名称、階層、順序
(2) 各作業で作成する仕様書の種別
(3) ステータスのパターン
(4) 作業誘導のパターンとパラメータ

この他、(5)作業の負荷(6)ワードプロセッサによる仕様書の様式(7)工程の見積りのための基準定数、なども定義するが[2]などで述べるのでここではこれ以上ふれない。

以下(1)〜(4)の定義内容と利用方法について説明する。

(1) 作業の階層と順序が誘導の基本的な枠組みとなる。

(2) 作業は仕様書の種別と多対１の対応関係にある。当該作業においてどの仕様書の種別を用いるかを定義する。

仕様書の種別は①専用の設計支援ツールを用いて編集する仕様書②汎用のワードプロセッサで編集する仕様書、の２種類がある。我々は百数十種類の仕様書様式を用意している。

①の仕様書の種別とツールとの対応はツールをシステムに組み込むときに記憶する。このためこの対応関係はＣＡＳＥプラットフォームが管理している。

作業とツールは仕様書の種別を介して間接的に対応している。この対応によりツールの自動起動ができる。

(3) 仕様書にはステータスという属性がある。例えば"作成中"や"完成"などの作業の状況を表す。ステータスの遷移は仕様書の種別によって異なる。例えばテスト項目表では"テスト項目作成中"→"テスト項目完成"→"テスト中"→"テスト完了"という遷移となる。このように仕様書に対して設定可能なステータスを仕様書の種別ごとに定めている。このステータスの組合せをステータスのパターンと呼び、仕様書の種別と対応づけて定義する。

(4) 作業誘導の一例をあげると、プログラム作成において①編集、②コンパイル、ここでエラーがあれば①に戻り、なければ③テストに進み、さらにここでバグがあれば①に戻り、なければ④次の作業、のように誘導が進む。

誘導は一通りではなく、いくつかのパターンを用意している。各作業についてどのパターンを用いるかを定義し、またパターンに与えるパラメータ(後述)を定義する。

An Integrated Software Development Environment on Client Server System(4) -Process Navigation-;
Akihiro YOSHIKAWA*, Masato TAKADACHI*, Hirofumi HOGAKI**,
*Hitachi, Ltd, **Hitachi Software Engineering Co., Ltd.

4．作業誘導

作業誘導は前述の作業手順に従って以下の支援を行う。

(1) 行うべき作業を表示し、ツールを自動起動する。

(2) 作業に対応する仕様書やプログラムを自動的にツールに読み込ませる。

(3) 関連する仕様書のステータスなどに基づいて作業状況を検知し、次に行うべき作業を表示する。

(4) プログラム編集後のコンパイルなどの定型的な作業は自動的に行う。

(5) 仕様書の編集中に、関連する仕様書を表示する。

以下、主に問題点や対処などについて順に説明する。

(1) 前述の作業とツールとの対応に基づいて起動を行う。

(2) 作業手順では作業と仕様書の種別との対応を定義するので、作業に対応する具体的な仕様書名は分からない。従って工程計画の際に作業と仕様書名との対応を記憶し、作業誘導の際にスクリプトの引数に与える。

(3) 複数の作業者による共同作業では、他の作業者の作業状況に応じた対処が必要である。例えば他の作業者に影響する部分は優先される必要がある。しかし他の作業者の状況の把握や対処の判断は、いずれも難しい。

本システムでは仕様書のステータスによって全作業の状況を検知できる。さらに作業の階層や順序、仕様書間の関連、ツールのリターン値等を勘案して対処を決定する。そのための論理を記述したスクリプトパターンを予め用意しているので、適切な誘導が実現可能となる。

(4) 例えばコーディング作業では編集、コンパイル、デバッグなどの一連の作業を繰り返す。複数のツールを連続的に起動することは従来から行われていたが、例えばコンパイルでエラーが発生したら直ちに編集に戻るなどの制御を設定するのが難しいので初級的な作業者は十分に使いこなせなかった。本システムではこれを作業手順として予め定義しておくので、すぐに利用できる。

(5) 仕様書間の関連を記憶することで、例えば関連仕様書の参照、仕様変更の時に関連仕様書に変更通知を送付、などができる。ここで、関連の登録方法が問題となる。作業者が明示的に指示する方法は操作が繁雑であり、誘導の観点からも初級者に向かない。次のようなとき自動的に関連を登録できる。(a)コンパイルなどの生成処理のとき（入力と出力との関連）(b)ある仕様書の編集画面から他の仕様書を呼びだしたとき。また仕様書の種別間の関連、仕様書の担当者が同じか、仕様書の名称が似ているかなども判断材料に利用できる。

5．スクリプトの生成

本システムでは前述のような作業誘導を行うプログラムを前述の作業手順から自動生成する。プログラムはコマンドスクリプトで記述される。作業の階層と順序に従って作業を順に進めてゆく部分とさらに詳細な誘導を行う部分とからなる。後者の誘導を行うには誘導のパターンを用意し、これにパラメータを与えて生成する(図２)。

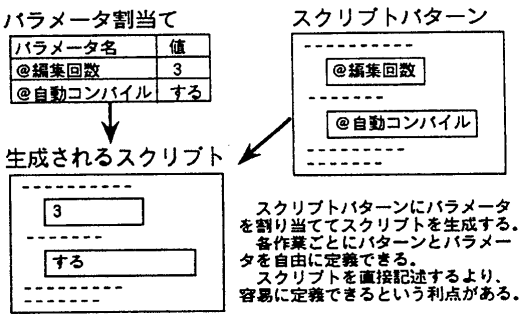

図２　スクリプトの生成方法

このスクリプトパターンには、指定可能なパラメータの種類と指定方法、生成されるスクリプトの使い方などの説明も形式的に記述しておく(図３)。パラメータを割当てるときはこの説明を参照しながら定義する。

```
%usage start 編集生成編集
   編集ツール１：最初に起動するエディタ
   生成ツール　：編集した仕様書を元に、別の仕様書を
                 生成するツール
   編集ツール２：生成された仕様書を編集するエディタ
%usage end
%param 編集ツール１,編集ツール２,生成ツール
%param 編集シート=$1,生成シート=$2
%param 生成可能ステータス=完成
%copy 編集 {編集ツール=@編集ツール１}
if [$SHEET_STATUS = @生成可能ステータス]
then
 %copy 生成問い合わせ
 if $?
 then
   %copy 生成{生成元シート=@編集シート,
              生成先シート=@生成シート}
   %copy 編集{編集ツール=@編集ツール２,
              編集シート=@生成シート}
 fi
fi
```

図３　スクリプトパターンの例

スクリプトパターンは開発作業のノウハウを明示的に表現し、システムに登録する手段を提供している。作業者は各々の技術水準に応じて(1)標準提供されたスクリプトをそのまま使用(2)パラメータを変更(3)生成されたスクリプトを修正(4)独自のスクリプトパターンを登録、の４段階から選択して作業誘導のサービスを受けられる。

6．おわりに

予め定義した手順に従って作業を誘導する方法について述べた。熟練者が持つ作業のノウハウを初級者が利用できること、他の作業者の作業状況を検知して適切に誘導できることに利点がある。今後本システムを適用評価し、有効なスクリプトパターンを蓄積し、改良してゆく。

7．参考文献

[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，(1992).

[2] 山中他：ＣＳＳ統合開発環境(5)－プロジェクト管理－，第４５回情報処理学会全国大会論文集，(1992).