

ソフトウェア開発における協調支援環境 Vela

— (1) ねらいとアーキテクチャ —

2G-1

落水 浩一郎 山口 高平

静岡大学

1. はじめに

我々は、プロセス・モデルを「新しい標準化の枠組」として位置づけ、以下の2種類の協調支援機能の実現を目標としたプロトタイプ V e l a (Versatile Environment based on Logic programming and AI techniques) の開発を進めている。(1) 設計の熟練者から初心者への技術移転の促進、(2) プロジェクト構成員間の意思伝達、合意形成、情報交換の促進。本稿では、V e l a の背景思想とアーキテクチャ設計の結果を述べる。

2. 環境の構成要素と相互関係

図1に環境の構成要素とその相互関係を示す。図1において、オブジェクト・ベースの状態変化がトリガーとなりプロセス指向タスクや情報指向タスクが起動される[1]。プロセス指向タスクは実行時に結合されるプロセス記述に従ってオブジェクトの状態を変化させるツール群を起動する。情報指向タスクはプロジェクト構成員間の交信をひきおこす。すべてのタスクはプロジェクト構造にもとづいて順序づけられている。

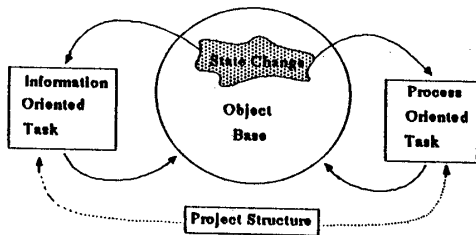


図1 環境の構成要素と相互関係

3. プロセス・モデル

図1に対応するプロセス・モデルを図2に示す。

Control	Task Network
Subprocesses	Process Scripts
Activities	Tool & Communication
Typed Objects	Artifacts / Products

図2 Vela におけるプロセス・モデル

図2において、最下層(第4層)は型付けられたオブジェクトの世界であり開発過程で必ず作成されなければならない情報の仕様化に対応する。第3層は、それらのオブジェクトの生成や状態の変化を引き起こすツール群や人間による情報交換の活動に対応する。第2層

は「検査の手順」、「承認の手順」等のサブ・プロセスであり、ツール機能の統合、ツールと人間のインタフェース仕様を定義する。第1層はそれらのサブ・プロセスを状況に応じて起動する起動制御の層である。

起動制御の層はさらに二階層にわかれる。

(1) プロジェクト・レベル

ソフトウェア開発を以下のように定義する。「ソフトウェア開発は複数のタスクからなり、それらはプロジェクト開始時または進行時に動的に生成される。各タスクはプロジェクト管理者からみた仕事の最小単位であり、実行順序はタスク・ネットワークにより制御される(図3)。」プロセス指向タスクはサブ・プロセスを動的に結合して実行することで達成される。情報指向タスクは、ソフトウェア開発の進行に伴って作りこまれる種々の制約の緩和や解除の必要性から起こるプロジェクト構成員間の交信を支援する。

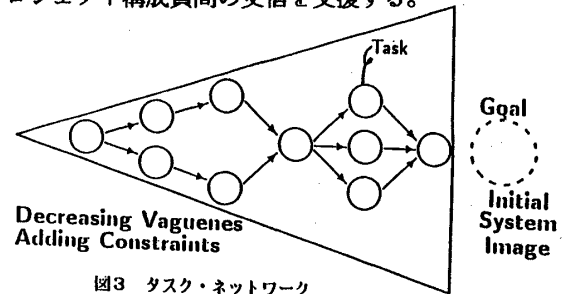


図3 タスク・ネットワーク

(2) 開発者レベル

プロセス実行系は以下のような機能を有する必要がある。(1) 「現在、自分はどういう状況にあるのか」、「次に何をすればよいのか」、「問題に依存して発生する特殊な状況にどうのように対応すればよいのか」等の疑問に対して適切な指示を与えてくれるナビゲーション機能。(2) 過去に実施されたプロセスは必ずしも現在の状況に適合しないので、プロセス記述を動的に変更することを支援する機能。(3) プロセス実行時に発生するソフトウェア開発に特有のバックトラックの問題、「状況に応じてバックトラックの必要性が認識され、またバックトラックする点も状況に応じて決定される」に対処する機能。

第2層におけるサブプロセスに関しては、環境内部に知識獲得の手段を組み込む必要がある。また、その

ような知識の洗練化の機構も組み込む必要がある。このような機構として種々の知識プログラミング技法の適用が有効である [2]。

4. プロトタイプVelaのアーキテクチャ

以上の諸機能を統合したプロトタイプVelaのアーキテクチャを図4に示す。

システムの機能は大きく、「知識獲得・生成フェーズ」「知識利用フェーズ」にわけられる。これは、ソフトウェア開発に対する社内標準化の流れが、「新規分野の開拓にともなう開発経験の蓄積」「開発経験の整理による標準化」「標準工程をもとに類似の情報システムを開発しつつノウハウを蓄積する」のようになることに忠実に従ったものである。

(1) 知識獲得・生成フェーズ (標準化の幼年時代)

まず、開発履歴を事例ベースに蓄積しつつ、現在の状況によく適合する開発事例を事例型推論の技法を利用して再利用する。この過程で対象分野に依存したシステム開発の成功例、失敗例が蓄積され、さらに、開発者が遭遇する状況に応じて事例を再利用するための索引が再構成フェーズとのインタラクションにより動的に構成される。

(2) 知識利用フェーズ (標準化の青年時代)

事例ベースの内容が十分であると判断されたとき、形式化の過程を経て、知識ベースとオブジェクト・ベースにより標準化が実施される。以後のソフトウェア開発はこの枠組に従って実施される。作業手順に関する処方箋(プロセス記述)を利用する効果は、対象領域に関する問題解決経験が豊かである熟練者のプロセスを再利用することにより、経験の少ないプロジェクト構

成員の作業の質を向上させることにある。非決定性に対するナビゲーションやバックトラックの支援はメタ・インタプリタが担当する。背景理論として、知的後戻りや仮説推論の技法を適用する。

(3) 協調(交信)支援

分散環境上にタスクが実体化されて実行される時、複数のタスク間で生じる交信要求をプロジェクト構造に基づいて支援する。以下の機能が必要である。(1) 交信ドメインの決定支援…中間生成物名や作業内容を入力として、プロジェクト・ベースを参照しつつ、関連する人々の間に情報交換のための通信チャンネルを自動設定してくれる機能(2) ネットワーク上を長期にわたって流れ、討論の進行につれて自動的に成長することにより、意志伝達、合意形成、情報交換の結果の文書化を支援する情報構造の提供。

5. おわりに

Velaの特徴は、開発経験がある程度蓄積され、社内標準化が必要とされている対象領域に対してソフトウェア開発における柔軟な標準化の方法論と、標準化を推進するための基礎構造を提供することにある。

謝辞

本研究はSDAコンソーシアムの補助金と、科研費重点領域研究(1)(課題番号02249109)の一部の援助のもとにおこなわれた。記して謝意を表する。
参考文献 [1]K.Ochimizu,X.Fang,T.Ohta, "Vela: A Software Design Support Environment based on Logic Programming Paradigm", 静岡大学電子科学研究科研究報告書(印刷中)[2] 山口、落水: プロセス知識の獲得と利用: 第41回情報処理学会全国大会。

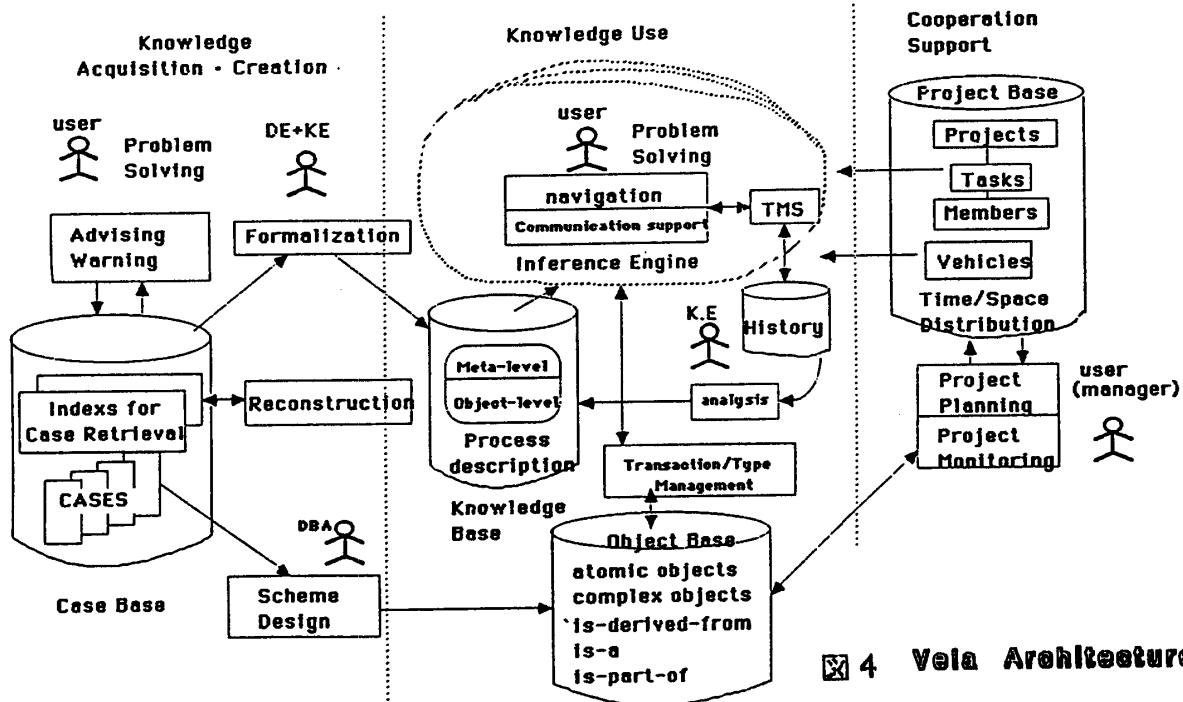


図4 Vela Architecture