

4F-8

Smalltalk-80における共同開発支援のための
共有ブラウザの開発菅沼 拓夫 宇田川 則幸 藤田 茂 福島 学 城) 健
千葉工業大学

1. はじめに

近年の計算機技術の発達、また計算機を結ぶネットワーク技術の発達により、ネットワーク上の複数のマシンによる共同作業への要求が高まっている。これに伴い、グループによる大規模ソフトウェアの共同開発などにおいて、開発したソフトウェアモジュールを他の開発者とファイルサーバ等を用いて共有することが多く行われている。しかし、これだけでは作業効率が十分に向上されにくく、共有資源の管理を含めた支援システムが必要である。このため、ソフトウェアモジュールの共有のみでなく、実時間会話機能などを追加することにより、共同開発をより円滑に行えるように支援するシステムの開発[1]等が行われている。

一方、ソフトウェアの開発環境として、コンパイラを提供するのではなく、Smalltalk-80[2]、Xerox Lisp[3]、Symbolics Lisp[3]等のように環境を提供する言語がある。これらの言語においては、環境が個人に閉じており、環境の機能を容易に変更・拡張することが可能であるため、開発者は自分の利用し易いよう環境に手を加えることにより、プログラム開発効率を向上させることが可能である。しかしその反面、複数人が別々の環境を用いて共同開発を行うため、開発したソフトウェアモジュールが個人の環境に特化した形となり易く、それらを統合して一つのアプリケーションとする場合のモジュール間の整合がとれず、統合が困難になるという問題が生じる。従って、環境型言語を使用して共同開発を行う際にはこれら個人環境間の差異を吸収し、共同作業を支援するシステムが必要である。

現在我々は、環境型言語であるSmalltalk-80を用いてソフトウェアの共同開発を行っている。本研究の目的は、Smalltalk-80の環境型言語としての利点を維持しながらLAN上のワークステーションを利用してソフトウェアの共同開発を行うための支援システムを開発することである。特に本稿では支援システムのツールの一つとして、ネットワーク上で共有されるソフトウェアモジュールのブラウジングを行うための共有ブラウザについて述べる。

2. 共有ブラウザ

現在我々は、Smalltalk-80を用いて演習支援システム[4]、プロダクションシステム[5]等のソフトウェアの共同開発を行っている。現状では、Smalltalkのソースファイルをファイルサーバ上で共有し、それを各々の環境にロードし参照することにより、他の開発者のモジュールとの整合をとりながらアプリケーションの共同開発を行っている。しかしこの方法の問題点として、ソフトウェアモジュールのライブラリが膨大になると、その中

から必要なものを見つけ出すのが困難であること、ロードする際に現在システム内にあるモジュールとの競合が発生すること等が挙げられる。これらの問題点を解決し、共同開発を容易に行うための機能として、Smalltalkによる共同開発の支援機構の一つとして、Shared Software Module Browser (以下共有ブラウザと呼ぶ)の開発を行った。共有ブラウザに要求される項目としては次のようなものが挙げられる。

- ・ Smalltalk環境内のBrowserとは独立したものであること
- ・ Smalltalkが提供しているBrowserと同等のブラウジング機能を有すること
- ・ 開発されたモジュールのバージョン管理を行うこと
- ・ モジュールの使用法、バージョン、バグ情報等を含むこと
- ・ 共有モジュールと既モジュールの整合性チェック機能を有すること
- ・ モジュールの変更履歴の参照が可能であること

3. 通信支援機構 (Communication Support System)

本システムのようにネットワーク上で使用されるソフトウェアを開発する場合、マシン間の通信制御に関する記述は大きな負担となる。このため本システムは通信支援機構(CSS)[6-7]上のアプリケーションとして開発する。CSSの目的は、モジュール間の通信制御に関する処理をモジュールの動作とは独立して行うことにより、モジュール間通信を支援し、ネットワーク上に分散構築するモジュール開発時の負担を軽減することである。

CSSはBlackboard Architectureに基づき、モジュールに対しデータの共有領域を提供することにより、複数モジュール間のデータ交換を支援する。モジュールは共有領域内のデータを読み書きする事によって、他のモジュールと通信を行いながら協調的に処理を進める。複数のモジュールがネットワーク上の異なるマシン上で起動された場合においても、CSSがマシン間での通信記述を吸収するため、モジュールから見た場合、同一マシン上のモジュール間での通信時と同様な共有領域に対するアクセスにより、ネットワークを意識することなく他のマシン上のモジュールとの通信を行うことが出来る。従ってモジュール開発者はネットワークでの通信記述を行う必要がないため、開発効率の向上が望める。

また、CSSにおいては水平分散型共有領域方式を採用しており、特にサーバを設けることなく共有領域方式を実現しているため、特定マシンに対するアクセス集中の軽減を実現している(図1)。

An Implementation of a Shared Software Module Browser for Cooperative Development in Smalltalk-80 Environments

Takuo Suganuma, Noriyuki Udagawa, Shigeru Fujita, Manabu Fukushima, Ken'iti Kido
Chiba Institute of Technology

