

ソフトウェア CALS への取組み

Nihon Unisys's Participation to Software-CALS Consortium

長谷川 邦夫

要約 本稿では日本ユニシスのソフトウェア CALS への取組みについて、ソフトウェア CALS 基盤の実証実験を中心に述べる。当社はソフトウェア CALS 実証コンソーシアムに参加しそこで二つのソフトウェア CALS 基盤、すなわち開発保守フェーズにおける現世代ソフトウェア CALS 基盤と次世代ソフトウェア CALS 基盤の実証実験に参加した。その中で行った CASE データ交換形式変換機能の開発と、ソフトウェア部品組立て型開発のための CORBA 結合技術の実証について報告する。

Abstract This paper describes how Nihon Unisys is approaching to Software-CALS and the experimental software proving of the basis of Software-CALS. We have joined the consortium for software proving, and participated two projects; the one is to prove the current software platform, and the another is to prove the next generation platform for the software development process on the basis of Software-CALS. The author reports the development of software that has the functionality for CASE data interchange and the experiment proving of the usefulness of the CORBA connection technology in the software development by componentware.

1. はじめに

ソフトウェア CALS 実証コンソーシアムは国内コンピュータメーカー、システムインテグレータなど 21 社で構成され、平成 8 年度および 9 年度にわたり、IPA（情報処理振興事業協会）による企業間高度電子商取引推進事業の一環としてソフトウェア CALS 基盤の実証実験を行ってきた。ソフトウェア CALS が目指すところはオープンでグローバルなソフトウェアの調達および供給の実現にあり、コンソーシアムはそのために次の三つのソフトウェア CALS 基盤を設定した。

- ・ 契約フェーズにおけるソフトウェア CALS 基盤
- ・ 開発保守フェーズにおける現世代ソフトウェア CALS 基盤
- ・ 部品組立て型ソフトウェア開発における次世代ソフトウェア CALS 基盤

それぞれのソフトウェア CALS 基盤の実証テーマは次の通りである。

- 1) 契約フェーズにおけるソフトウェア CALS 基盤
ソフトウェア開発の主に契約フェーズを対象とする、広域分散環境（インターネット）における協調的問題解決のプロセス。
- 2) 開発保守フェーズにおけるソフトウェア CALS 基盤
仮想企業による協調的ソフトウェア分散開発・保守のプロセス。
- 3) 部品組立て型ソフトウェア開発における次世代ソフトウェア CALS 基盤
分散オブジェクト環境における部品組立て型ソフトウェア開発の技術

これらの実証実験は、基盤を構成する標準規約やガイドおよびソフトウェアツールの開発を先行して行い、次いで環境を構築しそれぞれのテーマの実証実験を行う形で

進められた。

当社は開発保守フェーズにおける現世代ソフトウェア CALS 基盤の実証と部品組立て型ソフトウェア開発における次世代ソフトウェア CALS 基盤の実証に参加した。前者は仮想企業による協調的ソフトウェア開発を実現するための情報共有基盤を構築するものであり、当社は CASE データ交換形式 (CDIF: Case Data Interchange Format)⁴ の標準作成と交換形式変換ツールの開発を担当した。コンソーシアム内でこのツールを利用したシステムの開発・保守プロセスの実証実験で CDIF データ交換形式によるデータ交換の有効性が実証された。また次世代ソフトウェア CALS 基盤の実証は、CORBA (Common Object Request Broker Architecture)⁵ 環境における部品組み立て型の開発環境を構築するものであり、当社は CORBA 環境におけるソフトウェア部品結合によるシステムの結合テスト支援機能の開発および異なる ORB (Object Request Broker) を持つ CORBA 環境間の結合技術の実証実験を担当した。コンソーシアム内でのこの共同実証実験は CORBA による分散オブジェクト環境の実現に向けた具体的な技術実証となった。

本稿では CASE データ交換形式変換ツールと CORBA 結合技術の実証実験について述べる。変換ツールを用いた開発・保守プロセスの実証実験の内容および CORBA 結合技術の実証実験に使用した結合テスト支援機能の詳細についてはそれぞれ文献³⁾ および²⁾ に解説がある。ソフトウェア CALS 全般については文献¹⁾ を、ソフトウェア CALS 実証実験全体の成果についてはやはり文献³⁾ を参考にされたい。

2. ソフトウェア CALS と情報共有

仮想企業による協調的ソフトウェア開発 (以下、保守を含む) を支えるものは企業間での情報共有である。各社で電子的に作られた開発に関する情報は各社のあるいは共通のデータベースに蓄積され、それが企業間で共有されることによってソフトウェア開発が協調的にそして効率的に行われることができる。情報を共有することは情報を交換することであり、正しく交換し合うためには交換する情報の形式を企業間で取り決めておく必要がある。そして正しい相手と正しい情報を交換し合うためには交換する手順や仕組みが必要である。

ソフトウェア CALS では、ソフトウェア開発で共有する情報を大きく二つに分類している。

- ・プロジェクト管理データ (進捗データ等の開発プロジェクトの運営情報)
- ・プロダクトデータ (CASE データ等の開発ソフトウェアに関する情報)

そして、交換し合う情報の形式として、文書データの交換には SGML (Standard Generalized Markup Language)⁶⁾ を、またソフトウェア構造データの交換には CDIF を採用している。SGML 文書には他の形式のデータを含ませることが可能なので、SGML 文書であるソフトウェアの設計書にプログラム構造やデータ定義を CDIF データとしてを含め、すべてを SGML 文書として交換する。しかし本稿では SGML には触れず、CDIF データ交換について述べる。

ソフトウェア CALS では、情報を交換する仕組みとして MIL STD 974 で定められている CITIS (Contractor Integrated Technical Information Service: 契約者間の

契約に基づいて用意される統合された技術情報サービス)を参考にしたサービスをインターネットの WWW(World Wide Web)経由で提供できるようにしている。CITIS を参考にしたことにより、情報を正当に交換することが許されている企業間でのみ交換できることが保証されている。

このようにしてソフトウェア CALS では、ソフトウェア開発のプロジェクト管理データおよびプロダクトデータをインターネットを通して企業間で交換することができる。

3. CASE データ交換形式変換機能

3.1 CDIF

CDIF は 3 階層を成す 3 種類のモデルで構成される (図 1)。

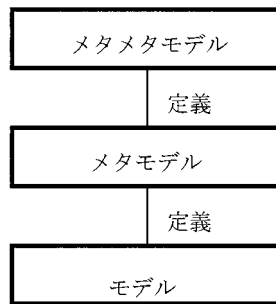


図 1 CDIF の階層

ユーザが CDIF を通して二つの CASE ツールの間で交換する目的の情報を単にモデルと言う。これはアプリケーションのデータを表現するもので、例えば COBOL のデータ定義記述やデータ構造図である。モデルを構成するデータ (要素) についての説明をメタモデルと言い、これは構成要素間の ERA (Entity-Relationship-Attribute) 表現で表現される。さらにメタモデルを記述するためのメタモデル構造要素の定義をメタメタモデルと言う。メタメタモデルは CDIF 規格の核であり、CDIF の拡張性を規定するものである。メタモデルで表現する対象を CDIF 規格の中のサブジェクトエリア (Subject Area) と言い、例えば CASE ツールで定義できるデータ定義や ER 図はそれぞれ一つのサブジェクトエリアである。このように CDIF は 3 階層の情報からなるが、データ交換の対象になるのはモデルである。

CDIF は既に EIA 標準であり、ISO でも標準化が進んでいる。したがってソフトウェア構造情報の交換形式として今後さらに有力なものになっていくと言える。

3.2 CASE データ交換形式標準

ソフトウェア CALS では国内のソフトウェア業界各企業で使用されている各種 CASE ツール間でデータ交換が可能になるよう CASE データ交換形式標準を作成し、その標準に準拠する形で CASE データ交換形式変換機能を開発した。今回の標準はソフトウェア CALS コンソーシアム参加企業の中の 4 社 (NTT ソフトウェア (株)、

(株)日立製作所、富士通(株)および当社)のCASEツール(MagicCASE、SEWB3、AA/BRMODELLER、StP/SE)間での交換を対象としている。これら4種のCASEツールが扱うサブジェクトエリアの定義(メタモデル)は同一であるが、モデルに各CASEツールの独自性が加えられている。従って、CASEツール間でモデルを交換するためには四つのCASEツールのモデルを共通に記述するメタモデルを抽出することが必要である。標準作成作業は主に、共通に使える範囲でCDIFにより規定されているサブジェクトエリアのサブセットを規定することであった。このサブセットの範囲内で、交換に情報を出す側は最大限の情報を出し、受け取る側は最大限の情報を受け取ることを原則とした、その結果、規定されたサブジェクトエリアは各社のCASEツールが持つ情報を十分に表現できるものとなった。

3.3 CASE データ交換形式変換機能

CASE データ交換形式標準に準拠するCASE データ交換形式変換機能は次の機能を持つ(図2)。

1) 各CASEツール対応の移出機能・移入機能

CASEツールのER(Entity Relationship)図情報あるいはデータ定義情報とCASEデータ交換形式との間の変換を行う。

2) CSV 移出機能・移入機能

CSV(Comma Separated Value)形式のデータ定義情報とCASEデータ交換形式との間の変換を行う。CSV形式に移出されたデータ定義情報は表計算ソフトウェアで扱うことができる。

あるCASEツールのモデルの中で、CASEデータ交換形式で定めるサブジェクトエリアの要素で表現できない部分は移出されない。またCASEデータ交換形式に含まれていても、あるCASEツールのモデルに含まれない部分はそのCASEツールに移入されない。

今回当社から実証実験に供したCASEツールは構造化分析・設計用のツールStP/SEである。当社はStP/SE対応のデータ定義情報(データ構造図—図3)の移出・移入機能を開発し、現世代ソフトウェアCALS基盤における開発・保守フェーズの実証に提供した。そこで初期の開発および開発後の保守を想定し、SI企業と開発企業の間でプログラムの設計情報をCDIFを用いて受け渡すことを実験し、他社のCASEツールとの間でデータ定義情報が正しく交換できることが実証されている。実証実験の詳細は文献³⁾を参照されたい。

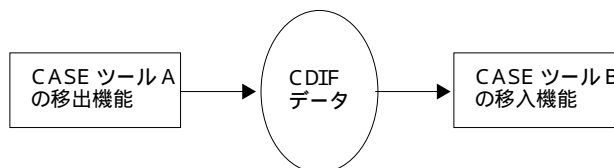


図 2 CASE データ交換形式変換機能

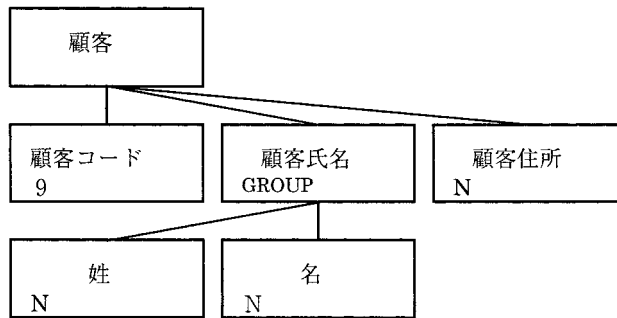


図 3 StP/SE で作成するデータ構造図の例

4. CORBA 結合技術の実証

4.1 ソフトウェア CALS と部品組立て型開発

ソフトウェア CALS 実証コンソーシアムでは次世代のソフトウェア開発は分散オブジェクトによる部品組立て型開発になり、企業間では開発情報の他にソフトウェア部品および開発環境を共有しながら開発が進められるようになると想定した。「作る」から「使う」への変革すなわちオープンな部品市場からソフトウェア部品を調達し、それを組み合わせることによってユーザの求めるアプリケーションを完成させる方法になる。保守のときは「直す」から脱却し、部品を「取替える」方法になる。ソフトウェア部品を「使う」、「取替える」ことにより開発者が独自に開発する部分を最小限にとどめようとするものである。このようなオープンなソフトウェア部品市場からの部品調達による開発・保守(コンポーネントウェア)によって、ソフトウェアの供給・調達の効率が飛躍的に向上されることが期待される。このことからソフトウェア CALS をして Software Commerce At Light Speed = SoftwareCALS とする呼び方も生まれている。

このような次世代のソフトウェア開発環境においては開発するソフトウェアの発注企業、システムインテグレータ、開発企業に加えて、部品ベンダ、部品ブローカと言う新しい役割を持つ企業が出現する。部品ベンダは部品を開発して提供し、部品ブローカは部品ベンダと発注企業間を仲介して、発注企業にそのソフトウェア開発に有用なソフトウェア部品を紹介することになる。さらに開発企業の仕事も変化し、部品の組立て、不足部品の開発、既存のアプリケーションの修正などを行うことになる。これらの役割を分担した企業が仮想企業を形成し、協調開発を進めていくことが想定される。

ソフトウェア CALS 実証コンソーシアムではこのような企業群が CORBA 基盤の上で部品組立て型開発を行うための基礎技術の実証実験を行ってきた。当社は日本情報通信(株)と共同して、部品ベンダと開発企業が異なる ORB 環境を持つ場合に必要な CORBA 結合技術の実証実験を担当した。この実証実験では当社が開発企業、日本情報通信(株)が部品ベンダの役割を担った。

4.2 CORBA 結合技術の実証実験

今後のアプリケーションは、アプリケーション間の連携を密にした複雑化と大規模

化および広域化が進み、その開発、保守および運用を可能にするためには分散オブジェクト・システムとして実現することが求められて来る。オブジェクト指向の考えと技術によって、システムを構成するオブジェクトはそのままソフトウェア部品になる。従ってソフトウェア部品は分散オブジェクトそのものであり、ソフトウェア部品を調達し、利用する環境基盤は分散オブジェクト基盤でなければならない。

分散オブジェクト基盤として CORBA の他に DCOM/ActiveX や Java-RMI があるが、この実証実験では大規模なサーバ・アプリケーションを含む部品組立て型開発の基盤として CORBA を選択した。CORBA は分散システムの一つの重要な基盤となりつつある。すなわち CORBA はビジネス・アプリケーションを分散オブジェクトの連携システムとして構築することを可能とし、分散されたアプリケーションやデータの統合を容易にする。特に CORBA 2.0 で規定された相互運用性に準拠すれば、相異なる ORB が連携し、利用者にとっては透過な一つの ORB 環境として利用できるため、プラットフォームに透過な広域分散システムの実現が可能になる。

部品ベンダおよび部品を利用してアプリケーションを開発する開発企業は、それぞれ任意の ORB 環境を備えることができる。部品を提供する環境と、部品を利用してアプリケーションを組立てる環境が、それぞれ CORBA による分散オブジェクト・システムである(図4)。部品は CORBA オブジェクトである。部品はオブジェクトであるから、利用する側(開発企業)はそれを自分の環境に移動させずに提供側(部品ベンダ)の環境においたままで利用(実行)できる。部品ベンダは部品の情報を自社の部品管理リポジトリを通して公開する。あるいはより簡便かつ広範な利用を可能にする方法としてインターネット上の WWW に公開することができよう。

こうして開発企業は、複数の部品ベンダの部品情報を参照して必要な部品を探索し、その稼動を部品ベンダの環境下で確認してアプリケーションに組み込むことになる。

当社が日本情報通信(株)と共同して行った実証実験のテーマは、異なる二つの CORBA 2.0 準拠 ORB 環境の一方が部品提供側(部品ベンダ)であり、もう一方が部品利用側(開発企業)という想定で、部品があたかも一つの ORB 環境に存在するかのように両者を結合する技術である。

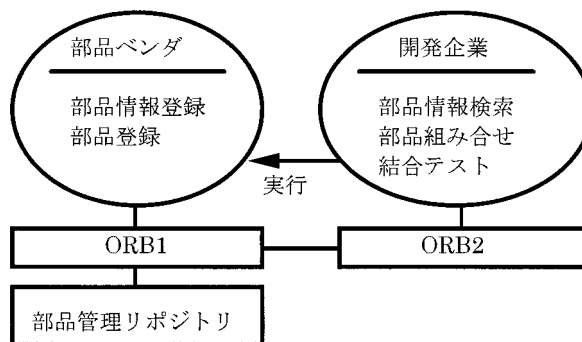


図 4 部品ベンダと開発企業間の CORBA 結合

4.3 CORBA 結合技術の実証のポイント

実証のポイントは、実際の人事考課サブシステムの開発を部品組立て型開発のプロセスで実施し、その過程で部品情報管理および部品のテストを支援する結合テスト支援機能（本実証実験のために開発）を利用して CORBA 結合の技術を確認することである。実証実験は次の作業過程を想定して行われた。

- ・ 部品ベンダにおいていくつかの部品について部品情報を部品管理リポジトリに登録する。
- ・ 開発企業が部品ベンダ内の部品管理リポジトリにアクセスし、開発に必要な部品の情報を検索する。当然、開発企業には部品管理リポジトリの存在は見えない。
- ・ 開発企業において、部品ベンダの部品と自分が作成した部品を組み合わせる人事考課サブシステムを想定し、部品ベンダ環境下の部品をテストする。部品のトラブルが発見されたら、その内容を部品ベンダに通知する。
- ・ 部品ベンダは部品のトラブルについて部品を修復し、その情報（トラブルの内容、部品の変更の理由、方法等）と改訂部品の情報を部品管理リポジトリに登録する。
- ・ 開発企業は部品のトラブル情報やその修復に関する部品変更情報を検索し、改訂された部品を使用するために、その情報を検索し、部品そのものを結合する。部品情報として以下のものを想定した。

- ・ 部品基本情報（部品の属性、複合部品の構造）
- ・ 部品変更情報（部品の不具合および変更に関する情報）
- ・ 部品テストケース情報（部品をテストしたテストケースに関する情報）

この実証実験のために用意した結合テスト支援機能は以下の機能で構成される。

- ・ 部品基本情報管理機能
- ・ 部品変更情報管理機能
- ・ 部品テストケース情報管理機能
- ・ 結合テスト支援用 CORBA 間接続機能
- ・ リポジトリ - ORB 間接続機能

はじめの三つの部品情報に関する管理機能（ここではまとめて部品情報管理機能とする）は当社が開発し、後の二つの接続機能は日本アイ・ピー・エム(株)が開発した。結合テスト支援機能の詳細については文献²⁾を参照されたい。

CORBA 結合の規定としては CORBA 2.0 の IIOP (Internet Inter-ORB Protocol) が規定されているが、この実証実験のように部品情報管理機能や部品管理リポジトリ機能等のアプリケーションと ORB を接続し、このアプリケーションのユーザにとって ORB やリポジトリを透過なものにするためには接続の規定が必要である。そのためにソフトウェア CALS 実証コンソーシアムとして

- ・ 結合テスト支援用 CORBA 間接続規約、および
- ・ リポジトリ-ORB 間接続規約

を定めた。結合テスト支援用 CORBA 間接続機能およびリポジトリ-ORB 間接続機能はそれぞれこの規約を、今回の実証実験環境に合わせて実装したインタフェース機能

である。

実証実験の環境は図5のように OS の異なる部品ベンダおよび開発企業の環境を LAN で接続したものである。

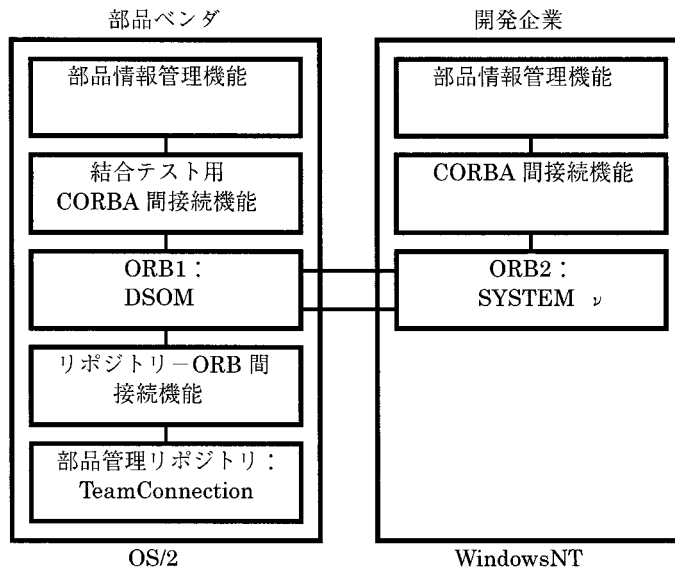


図 5 実証実験環境

ORB として IBM 社の DSOM および当社の SYSTEM v を、また部品管理リポジトリには IBM 社の TeamConnection を使用した。

4.4 実証実験の結果

具体的な人事考課サブシステムの開発を通して、開発企業と部品ベンダが異なる ORB 環境にある場合に CORBA を通してそれらを接続し、開発企業が部品ベンダの部品を利用する環境を構築できることが実証された。この環境が実現されれば、CORBA 部品の情報管理によって部品ベンダは部品の品揃えを増やし、部品を維持することが容易になる。開発企業にとっては利用できる部品の範囲が拡大する。また開発企業が CORBA 部品を遠隔地からテストできることは、開発企業が自分の負担を増大させることなくテスト環境を拡大でき、かつ即時にテストできるようになることを示している。このことから部品ベンダと開発企業の連携による部品組立て型開発の有効性が実証されたと言える。規約に沿って市販の CORBA 準拠とはいえ実装のことなる ORB を用いて CORBA 結合を実現できたが、この過程で用意したブリッジ機能やディレクトリ管理機能が有効に働いたことは、今後の ORB 接続の参考になると言えよう^[2]

しかし、この実験では次の点にまだ課題があることがわかった。

- ・ 部品情報に内容の吟味、そして如何にして目的に合う部品を発見できるか
- ・ 部品ベンダにおいていつどのようにして CORBA 部品を立ち上げるか
- ・ 部品を遠隔地からテストする場合の部品の使用契約を含めた流通機構

今後、実験を次のように拡大すればさらに実用化へ近づく実証が得られると考えられる。

- ・部品検索機構との連動

今回のコンソーシアムによる実証実験の中では、部品仕様表記要領に基づいて書かれた部品仕様から目的の部品を検索する技術の実証実験も行われた。CORBA 結合における部品情報管理機能とこの部品検索機能を連動させれば実用的な環境が実現できると思われる。

- ・インターネットにおける広域実験

今回の実証実験は LAN のもとで実施したが、協調的開発を行う企業間で安全にファアウォールを通過するためのインターネット・トンネル技術とセキュリティ技術を利用して、インターネット上で CORBA 結合を行うことが必要である。

- ・複数の部品ベンダとの接続

今回は部品ベンダーと開発企業がそれぞれ 1 社であることを想定して実験したが、ソフトウェア部品が実用化される場合は複数の部品ベンダーを利用することが必要になるとと思われる。

- ・ORB の組み合わせを増やす

今回は二つの ORB を対象にしたが、市場に出回っている他の ORB による実験を行い、ORB に透過な結合技術を確認しておくことは有効である。

5. おわりに

本稿ではソフトウェア CALS 実証実験のうち当社が関連した部分を紹介した。これだけでソフトウェア CALS 実証実験の評価や今後の動向を結論することはできないが、CDIF によるソフトウェア構造情報の交換と CORBA の下での部品情報管理および部品組立てが、ソフトウェア開発における企業間の協調が進み、新しい開発パラダイムが実現されてくるだろうことを想像させることは否めない。ソフトウェア CALS の考えを徐々に適用し標準の採用と情報の共有、ソフトウェアの部品化を図ることが Construction At Light Speed につながる道であろう。

本稿を執筆するにあたり、2 年間にわたりツールの開発と実証実験を実施した関連部署の方々および共同の開発と実証実験に携われたコンソーシアムのみなさんに感謝する次第である。

-
- 参考文献** [1] 長野宏宣, ソフトウェア CALS のねらいと実証実験について. 情報処理, Vol. 37, No. 12, 1996, 12, pp. 1083 ~ 1088.
 [2] 村山一美, SYSTEM v [nju:] と SOMobjects の相互運用性の実現 ユニシス技報 Vol. 17, No. 3, 1997.
 [3] ソフトウェア CALS 実証コンソーシアム, ソフトウェア CALS 環境構築用のソフトウェア開発および実証 最終成果発表会資料, 1998.
 [4] CDIF Overview EIA/IS-106, January, 1994.
 [5] 小野沢博文: 分散オブジェクト指向技術 CORBA, ソフトウェアリサーチセンター, 1996.
 [6] Standard Generalized Markup Language ISO/IEC 8879, 1986.

執筆者紹介 長谷川 邦夫 (Kunio Hasegawa)

1972年東北大学理学部地球物理学科卒業，同年日本ユニシス(株)入社．事務処理分野のシステム開発，システム開発方法論，開発支援ツールの研究開発，近年ソフトウェアCALS実証実験に従事．日本ソフトウェア科学会員．