

リアルタイム制御用ソフト設計製作支援システム (NSCASE)

加藤博一 中北輝雄

新日本製鐵株式会社技術開発本部

リアルタイムシステム用ソフトウェア開発支援システムNSCASE (Nippon Steel Computer Aided Software Engineering system)を開発し、新日本製鐵社内外の制御用計算機システムの開発に全面的に適用中である。NSCASEはソフトウェアの生産性向上、品質向上を狙ったソフトウェア開発支援システムであり、ターゲットマシンからの独立およびソフトウェアライフサイクル全体の支援を特徴としている。

COMPUTER AIDED SOFTWARE ENGINEERING SYSTEM FOR REAL TIME CONTROL SYSTEM (NSCASE)

Hirokazu Kato Teruo Nakakita

Technical Development Bureau
Nippon Steel Corporation

20-1 Shintomi Futtsu-City, Chiba Prefecture, 299-12 Japan

Nippon Steel developed a computer aided software engineering system for real-time systems, called the NSCASE (Nippon Steel Computer Aided Software Engineering system). The NSCASE is extensively used in the development of process control computer system for Nippon Steel and other companies. The NSCASE is aimed at improving software productivity and quality, and is characterized by the independence from target machines and the support of an entire software life cycle.

1 はじめに

新日本製鐵では昭和30年代後半より、リアルタイム制御用コンピュータのアプリケーション・システム（以下単に“システム”）のエンジニアリングを数多く行ってきており、既に社内で600、社外で100を越すプロジェクトの実績を持つ。また社内で常に約270に及ぶシステムを保守している。このような数多くのプロジェクトの経験を踏まえ、ソフトウェアの高品質化とソフトウェア・エンジニアリングの効率化を鋭意推進してきた。特に設計方法論、ドキュメント・フォーマットおよびマルチベンダー対策、CASE支援システムなどの開発・適用を重点的に行ってきた。

ここに紹介するのは一連のソフトウェアの高品質化・高効率化の集大成ともいべきリアルタイム制御用アプリケーション・ソフトウェア開発を設計から製作・テスト・保守までを一貫して支援するシステム - NSCASE (Nippon Steel Computer Aided Software Engineering system) - である。

NSCASEは1984年から2ステップに分けて開発に着手し、88年3月に開発を完了し、社内外プロジェクトに適用している。

2 NSCASE開発の背景

制御用コンピュータのアプリケーション・ソフトウェア（以下APソフト）はオンライン・リアルタイム性が厳密に要求される点、また高い品質が要求される点で非常に開発コストがかかる。一方、生産ラインのコンピュータ装備率は急速な伸びを示しており、コンピュータがライン・コントロールの中核的な位置付けになってきている。

更に、ハードウェア費用に対してAPソフトの開発費用のウェイトがかなり大きくなってきている。

このような事情から、APソフトの生産性向上を図ることは約15年前からの大きな課題であった。一方、当社ではいろいろなメーカーの機種を採用しており、APソフトを開発する上で機種毎のOS、言語、コマンド仕様等を知っていなければならない、マルチベンダー対応という課題もあった。このためNSCASE開発以前の対策として次のような標準化をまず実施した。

2.1 設計方法論の標準化

各製鐵所（当社は9製鐵所）にて、バラバラにマニュアル化していたものを1975年から全社統一化の諸活動を進め、1983年からの試使用を経て、1985年にPROCESS III (PROCESS Computer Engineering System version-III) という名称のソフトウェアエンジニアリング手法、ドキュメンテーションの標準化・マニュアルを完成させると共に、それを全社標準として採用し、設計における属人性を排除して、APソフトの品質向上、開発効率向上に成果をあげている。

2.2 APソフトインタフェースと言語の標準化

制御用コンピュータには多くの機種がありマシン毎のOSや言語の違いが再利用による信頼性向上、効率化の障害であった。このためプログラミング関連の仕様について、当社標準の言語仕様 (FORTRAN-77の採用)、拡張OS仕様 (標準セミ)、テスト仕様記述言語等の規定を行ってソフトインタフェースを標準化した。この結果マルチベンダー対応、ソフトの再利用が容易になると共にターゲットマシンからの独立という課題に対して、ホスト汎用計算機によるソフト開発が可能となった。

図1に標準言語および標準セミの位置付けの概要を示す。

図に示すようにプログラミング言語を標準言語化することにより、ソフトウェア技術者が各ターゲットマシン独自の言語の知識を修得する必要もなく、APソフト開発ができると共に、異メーカー間におけるAPソフトの再利用が実現できる。また標準セミはOSとAPソフトとの中間に位置し、タスク制御、ファイル入出力、CRT入出力、帳票出力等を容易に行うためのアプリケーションプログラムインタフェース（API）を提供している。標準セミは各マシン独自のセミに標準セミ対応のロジックを付加することにより、標準言語の採用と同様にAPソフトの容易な開発、再利用を可能とした。

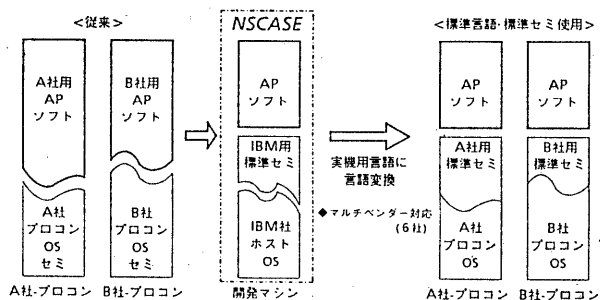


図1. 標準言語および標準セミの位置付け

3. NSCASE開発のコンセプトおよび実現手段

3.1 設計書の世界とデータプロセッシングの世界との一体化

CASEシステムは、ソフトウェアライフサイクル全体を一貫して支援することが重要である。特に多大のマンパワーを要し、品質が後工程に大きな影響を与える設計段階の支援はCASEシステムの中で重要なキーとなる。設計段階で複雑なリアルタイムシステムの仕様を定義する設計書は、テキストだけでなく、図形、表をミックスしたMMD (Multi Media Document) の様式で構成されている。設計支援を行うには、このMMD表現された設計書を作成するEWSとデータ処理を行うホストコンピュータ（以下ホスト）間で双方向のデータ交換、再加工することが必要となる。NSCASEではEWS上のドキュメントの表現コードを汎用コードに変換し、更にホスト上には、この汎用コードを解釈、再加工できるAPI (Application Program Interface) を設け、これを可能とした。

3.2 論理構造による設計の一元化

設計書、ソースコード、テストデータ等の多くの情報およびデータは重複して記述されている場合が多い。このため、ソフトウェア開発の各フェーズにおいてソフトウェア情報が相互に自由に参照、利用出来るように情報を一元管理することが望ましい。NSCASEでは、ホスト上にシステムディクショナリ (SD)、ソフトウェアベース (SB) およびソースライブラリ (SL) を持つことにより、ソフトウェア情報の一元管理を行っている。これにより設計情報をプログラム製作、テスト段階で容易に利用できるようにし、またソフトウェア成果物の再利用が容易に行えるようにした。

3.3 ターゲットマシンからの独立

制御用計算機システムのターゲットマシンとしてはコンピュータメーカー各社のミニコンピュータが使用されている。このミニコンピュータのオペレーティングシステム (OS)、拡張OS (OSとAPソフト間のインタフェース)、言語仕様は各メーカー間で異なっており、APソフトの可搬性を大きく阻害している。そこで当社では、拡張OS仕様、言語仕様の標準を定義し、これらを使用することによりAPソフトの可搬性を確保した。また、この標準に従ったプログラム開発をNSCASEで支援することにより、ターゲットマシンから独立したソフトウェア開発支援環境を実現した。

3. 4 設計方法論等の変化への柔軟性

現在、当社のリアルタイムシステム開発の生産技法としてはウォーター・フォール型の設計方法論であるPROCESⅢを制定している。しかし生産技法は時代と共に変化するものであり、CASEシステムとしてはこの変化に柔軟に対応できる必要がある。そこでNSCASEはSD及びSB構築、自動検証機能、ドキュメント自動生成等の手順を外部定義化し、容易にカスタマイズできるようにした。

4 NSCASEの概要

4. 1 機能構成

NSCASEは、図2に示すように、設計支援のCADとプログラム製作・テスト支援のCAM/CATに分けられる。

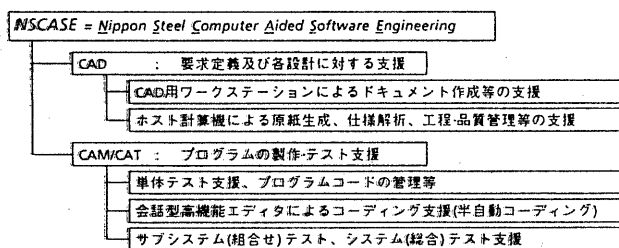


図2. 機能構成図

4. 2 システム構成

NSCASEのシステム構成図を図3に示す。ホスト上に各種支援機能を構築し、このホストと設計支援端末として図形処理に優れ、ネットワーク構成されたEWSを接続し、プログラム製作・テスト支援端末として3270エミュレーション機能を有するプログラミング・ワークステーション（PWS）を接続している。本システム構成は一貫支援の場合を示しているが、部分支援システムを構築することも可能である。

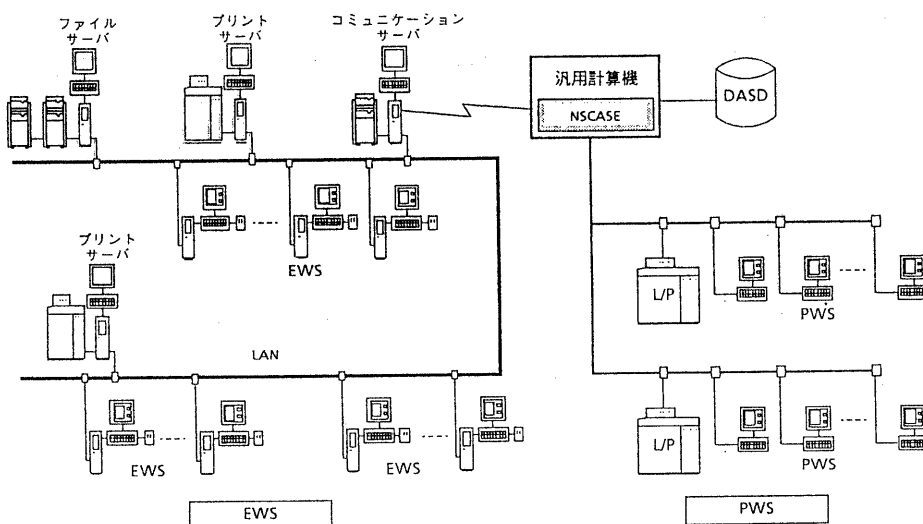


図3. システム構成図

4.3 システム概要

図4に業務フェーズとNSCASEとの関係を示す。設計支援の基本としてPROCES III、製作・テスト支援の基本として標準言語・標準セミを用いている。

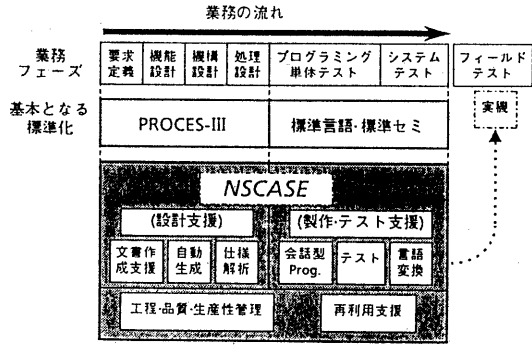


図4. 業務フェーズとNSCASEとの関係

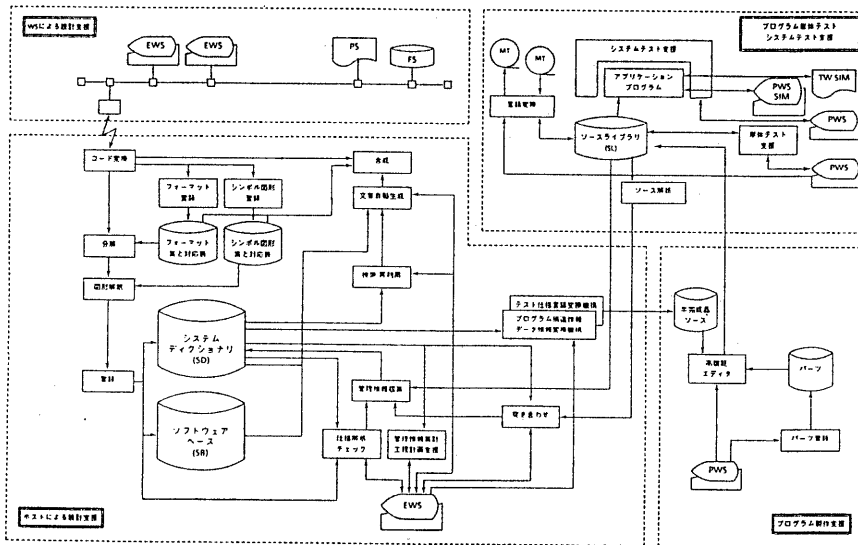


図5. システム概要図

次にNSCASEのシステム概要を図5に従って、APソフトを作る時の作業の流れに従って説明する。システム分析・設計工程では設計者が図中の「WSによる設計支援」システム及び「ホストによる設計支援」システムを（以下「CADシステム」と称す）使って仕様書を作成していく。仕様書の表現は基本的に以下の3つの形式によっている。これにより仕様書の表現形式化を図り、設計者の世界とコンピュータでのデータ・プロセッシングの世界との一体化を実現できる。

- ① F I F (Fill In Form) 方式
- ② 図形表現
- ③ (プロジェクト毎の) 統一用語辞書

手続きには予めEWS上で仕様書のフォーマット類、図形表現に使用する各種の図形類、用語類を作成しCADシステムに登録しておき、必要に応じてプロジェクト毎にどれとどれを使用するかを指定する。設計者は仕様書を作成するに当たって、フォーマットなどをプロジェクト毎に指定されたものから選択してEWS上に持ってきて、これらを仕様書用原紙、図形テンプレート、カナ漢字変換辞書として

使用する。作成された仕様書は日時、設計者のサイン入りでEWSからCADシステムの方へ送ることとなる。仕様書が送付されることによりCADシステムは自動的に仕様書のフォーマット番号、図形番号などを基に仕様書の内容を分解・解釈し、SD及びディクショナリに対応した情報集としてのSBにこれらの分解・解釈結果を登録する。

所定の設計フェーズで、かつ所定の仕様書が集まった段階で、設計者はCADシステムに対して様々な指示を出すことができる。例えば、登録した全部の仕様書にその仕様の内容解析と各種のチェックをさせ、仕様書の完成度合いなどをレポートिंगさせたり、別の新しい仕様書を自動生成させたり、あるいは仕様の内容を仕様書とは別の角度から表現した参考資料を生成させたりできる。レポートングなどはCADシステムからEWSに端末モードで出力されるが、自動生成される仕様書や参考資料は所定のフォーマットに記述された形式でLAN経由でEWSに出力される。

このようにしていくつかの設計フェーズを順次経て最終的にはプログラム・コードの作成が可能な段階まで仕様がブレイクダウンされる。

次に、プログラミングに入り、ここでの作業はCAM/CAT（会話型高性能エディタなど）を使用して行われる。まずプログラマからの指示でプログラム1本ごとに各種の仕様が編集されたうえでソース・コード化され、ソース・コードの骨格、テスト・データの骨格が自動生成される。次にプログラマは上述の骨格ソース・コードとあらかじめ登録されているプログラム・パーツ・コードとを基に高性能エディタを使用して完全なソース・コードを作成する。

作成されたプログラム・ソース・コードおよび各プログラムに対する複数テスト・ケースのセットをCAM/CAT（単体テスト支援機能）に任意のタイミングでかけることができる。

これにより、単体テストを高速・大量処理できると共に、各プログラムについてプロジェクトの開発期間を通してのテスト履歴情報、品質管理情報の集約一元管理を行える。単体テストが終了したプログラムはシステム・テストにかけることとなるが、CAM/CAT（システム・テスト支援機能）が持っているプロセス・シミュレータの機能、各種の端末のシミュレート機能、その他シミュレーションの中断・再開、状況のセーブ・リロードなどの支援機能を使って実際の適用状況をシミュレーションすることにより、設計者などが中心となってAPソフトの機能を総合的にテストすることができる。

システム・テストが完了するとCAM/CATの機能を使ってターゲット・マシン用のソース・コードに言語変換して出荷となる。

これら一連の工程を通して、各種の管理データがCADシステムにより自動的に収集され、さらに集計・分析もさせることができる。収集される主なものは履歴および品質データなどである。このほかの機能として、予定データやリソース・データを入力することによりSD情報や実績情報と組み合わせて工程計画立案支援も行える。設計情報、ソース・コードの再利用の面では過去のプロジェクトのソフトウェア資産、パッケージ化、パーツ化されたソフトウェア資産がそれぞれ検索でき、電子化された状態で再利用、再加工できる。

（1）F I F方式について

設計用のフォーマットはほとんど自由に決めることができる。例えば図6に示すような各種のフォーマットをEWSの上で設計し、登録できる。設計書は文章・テーブル・ベクトル図形（将来はイメージも）が混在型で表現できるMMD形式になっている。各フォーマットにはフォーマット・ナンバーおよびそれを構成している各フレームごとのナンバーおよび名称がEWSで設定されており、最下層に位置するブロックに記述されている文章、テーブル、ベクトル図形などのオブジェクトが後述するSD上の

何のメンバー・タイプの何の属性をどのように定義するものかを関係付けることができるようになっていいる。この関係付けは基本的にテーブル形式で埋める簡単な方式が取られており、ホスト端末から入力し、カスタマイズや変更も容易にできるようになっている。

このようにして予め原紙フォーマットを登録しておく一方、設計情報が書き込まれた設計書用紙がEWSからホストに送られてくると、その設計書用紙がSDの何のメンバーに関するものかが設計情報として記述されているため、SDの対応するメンバーに上述のメンバー・タイプの属性構造にしたがって登録されるようになっていいる。また、各設計書用紙ごとの設計情報は設計書再現のためとメンバーに関する詳細情報の保管のためにSBに保管され、設計書をそのままLAN上のEWSに再現したり、ホスト上で再現したりすることができる。

以上の機能によりエンドユーザー毎に異なるフォーマットを容易に設計でき、容易にSDのリンクを定義できる。このことは設計仕様書の世界とデータ・プロセッシングの世界との連結の仕方をプロジェクトごとに自由に決定できることであり、NSCASEを非常に柔軟性のある設計支援システムにしていいる重要な1要素である。

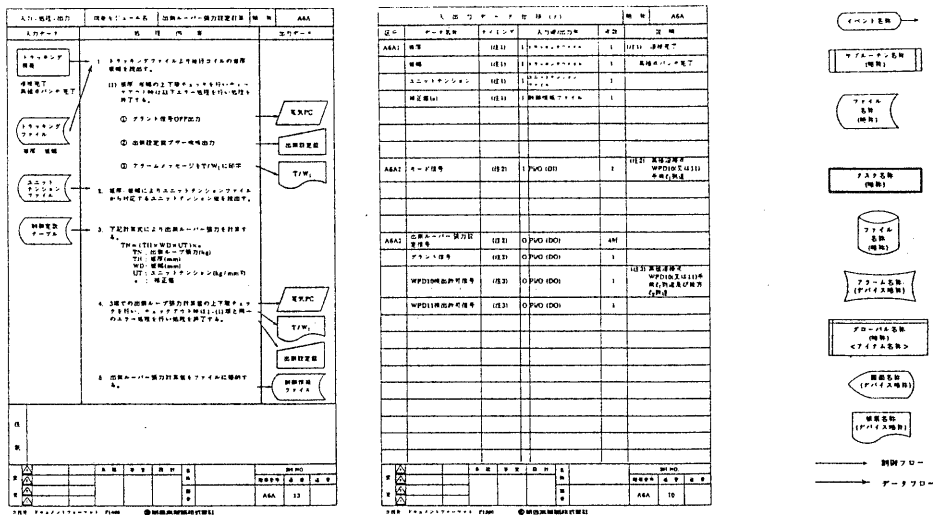


図6. 設計書用紙とシンボル図形の例

(2) 図形の解釈

コンピュータのソフトウェアに関する仕様は多くは図形的に記述され、図形による方が複雑な相互関係などの一貫性において、文章やテーブルによる記述より優れている場合が多い。また図形の表現による場合、予め決めたいくつかの要素図形の組み合わせでほとんどが表現されることから、図形処理に対しては次のような機能を備えることとした。即ち、要素図形として図7のようなシンボル図形を任意に設計し登録できるようにしている。

また、シンボル図形間の相互の関係を定める図形、具体的には各種の矢印線を設計し、登録できるようにしている。このシンボル図形や矢印線は、EWS上で設計しホストに登録する。

また、SDの構造と対応を設定できるようになっており、これらの対応関係の設計入力はEWS端末モードでできるようにしている。

これらのシンボル図形を設計で使用する際EWSから属性に対する設計情報の記述も可能である。

この図形解釈機能により、要求定義から処理設計に至るまで、および各種のデータ設計に幅広く図式表現を用いることが可能になり、かつ、そのカスタマイズが容易なため、柔軟に各種の設計方法論の要望に応えることもできる。

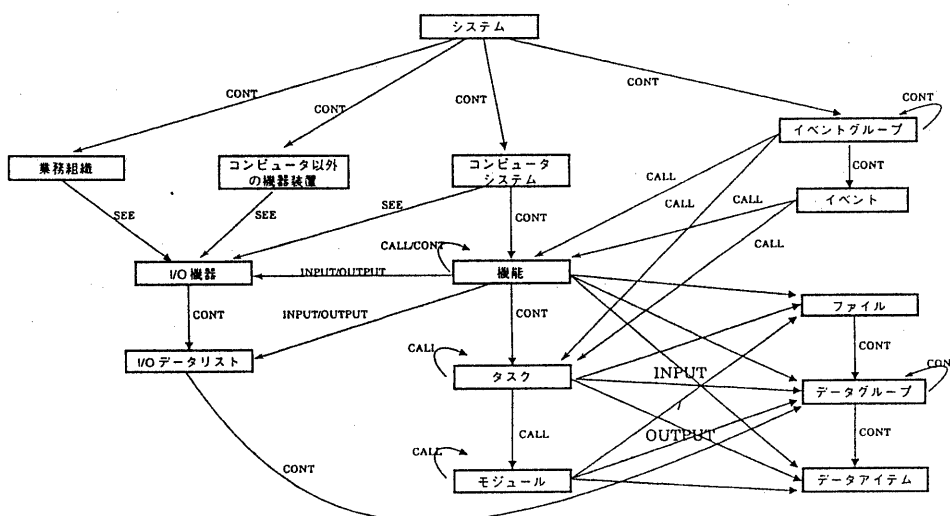


図7. システムディクショナリ (SD) の例

(3) 用語登録

データ名称はもちろん、各種の名称もプロジェクト毎に統一する必要がある。一方、各名称をユニークにしようとする一般に名称の長さが非常に長くなるため仕様記述に手間がかかるので、略称や同義語で記述しがちである。従って、用語の略称や同義語の管理、対応リストの生成をどこかで一元的に行う必要がある。NSCASEではこれにSDを使用してホスト上でやっている。

プログラミング段階では、データ名称とデータ・コード、プログラム名称とプログラム名コード、ファイル名称とファイル名コードの自動変換などのための全件管理を行っている。

このようにして用語の統一を全開発工程に渡って行え、その効果が大きいことは言うまでもない。

5 おわりに

ソフトウェア開発を一貫して支援するNSCASEの概要について述べた。本システムは高機能EWSとホスト間での双方向のMMD情報交換、およびターゲットマシンやOSから独立したソフトウェア開発支援環境等の特徴としている。現在、社内外プロジェクトのソフトウェア開発に全面的に適用されており、また一部の機能については既にUNIXマシンへの移植を完了、商品化もされている。本システムは鉄鋼プロセス用リアルタイムシステムにおけるソフトウェア開発のノウハウから誕生したシステムではあるが、一般のFAシステム開発等にも幅広く適用可能と考えている。

ソフトウェアの生産性向上、品質向上は今後とも永遠の課題であると思われる。本稿で報告したNSCASEの効果を最大限発揮させると共に、ダウンサイジング化、リバースエンジニアリングツールの開発等、より効果的なCASEの開発に今後も努力する所存である。