

異性体ソフトウェア群のバイナリ生成・管理手法に関する提案

3K-4

臼田尚志 萩原秀和 (高岳製作所) 河口信夫 稲垣康善 (名古屋大学)

1 はじめに

X 端末用ソフトウェア (X 端末ソフト) に代表される組込み機器用ソフトウェアにおいては、複数の機種への対応やユーザからの要求などにより、主要機能が同じで対応機種、機能の異なる実行ファイル (バイナリ) が多数必要となる。よって、ソースプログラム (ソース) とバイナリの管理やディスク資源の有効活用のためには、ソースとバイナリとの密接なバージョン管理が必要となる。

臼田ら [1] は、X 端末ソフトに代表されるソフトウェアとして異性体ソフトウェア群の概念を提案し、ソース管理に RCS [2] を用いることでソースの統一化を行う手法を提案した。

しかし、ソース管理を行うことで、単一のソースから対応機種、機能の異なるバイナリが複数生成されるため、多数のバイナリの管理が必要となる。また、機種や機能の間に依存関係、排他関係などの制約がある場合、従来のバイナリ生成手法をそのまま適用することは困難である。

本研究では、この問題を解決するため、異性体ソフトウェア群のバイナリ生成、管理を行う上で問題点を明確にし、制約に基づくバイナリ生成手法と、ソースのバージョン管理と密接に結び付いたバイナリ管理手法を提案する。

2 異性体ソフトウェア群の特徴及び問題点

異性体ソフトウェア群では、単一のソースから、主要機能は同じで対応機種、機能の異なるバイナリが複数生成される。また、ソースは、機能追加、バグ修正、性能向上などにより頻繁にバージョンアップを繰り返すため、それぞれに複数のバイナリが存在することになる。さらに、バイナリの生成条件は対応機種、機能によって異なり、それらは互いに依存関係、排他関係などの制約を持つ場合がある。

X 端末における制約の例を表 1 に示す。例えば、表示機能に関してはカラーとモノクロがあるが、両方の機能を同時に利用することはできないため、これらの間には排他関係が成立する。また、FD (フロッピディスク) フォーマット機能やファイル転送機能は FDD (FD ドライブ) を使用するため、FDD ドライバが必要となり、これらの機能の間には依存関係が成立する。

バイナリを生成する場合、対応機種、機能の間にある依存関係、排他関係などの制約を考慮しないと、不完全なバイナリを生成してしまうことになる。よって、異性体ソフトウェア群のバイナリを生成する際には、従来の Makefile に基づくバイナリ生成手法をそのまま適用することは困難である。また、バイナリ生成後、ユーザからの問い合わせやトラブルなどに即座に対応するためには、機種、機能、ソースなどのバイナリ情報を管理するための機構が必要となる。

現在高岳において、OEM を除く X 端末は 21 機種あり、X11R5 以降におけるこれらの機種に対応したソースおよびバイナリ数は、ソース管理を行う前ではそれぞれ 70,192 であり、ソース管理後ではそれぞれ 3,15 であった。これらは製品としてリリースしたものであり、リリースしていない開発用のものを含めると相当数にのぼる。これらを何らかのシステムなしに管理していくことは困難である。

表 1: 機種、機能に関する制約の例

機種・機能名	関係	機能名
カラー表示機能	排他	モノクロ表示機能
FD フォーマット機能	依存	FDD ドライバ
ファイル転送機能	依存	FDD ドライバ
PS/2 キーボード対応	排他	OADG キーボード対応
プリンタ制御機能	依存	パラレルポートドライバ
パラレルポート使用機能	依存	パラレルポートドライバ
プリンタ制御機能	排他	パラレルポート使用機能
機種:CSU	依存	カラー表示機能
機種:MSU	依存	モノクロ表示機能
機種:CSLFD	依存	カラー表示機能
機種:CSLFD	依存	FDD ドライバ

3 バイナリ生成、管理手法

上記の問題点を解決するため、図 1 に示すように、バイナリ生成システムを用いたバイナリの自動生成手法と、バイナリ管理による各種情報の管理手法を提案する。

A Method of Generating and Managing Execution Files for Isomeric Softwares

Hisashi Usuda, Hidekazu Hagiwara

Takaoka Electric MFG.

Nobuo Kawaguchi, Yasuyoshi Inagaki

Nagoya University

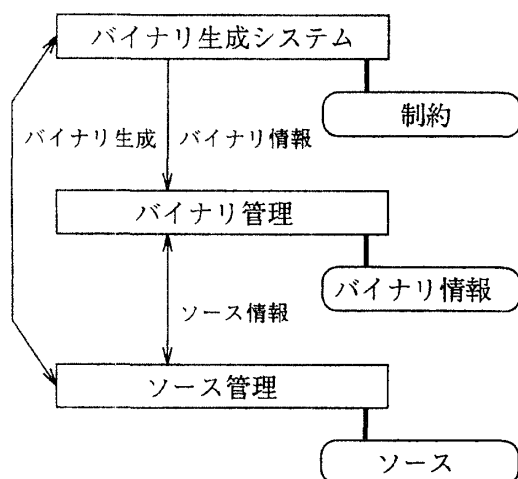


図 1: バージョン管理システム

3.1 バイナリ生成システム

対応機種、必要な機能などのバイナリの生成条件を指定するだけで、ソースの Makefile を自動的に生成し、コンパイルを行う。ソースは単一であるため、これらの機能は、共通のソースファイル内において `#ifdef` を用いて記述されているか、別のソースファイルで記述されている。よって、前者の場合は必要な `define` を定義し、後者の場合はそのソースファイルを使用するように Makefile を自動編集する。

バイナリ生成に関する制約はあらかじめ登録してあり、指定したバイナリ生成条件が制約に従っているか自動的に検査することができる。これにより、対応機種や必要な機能を選択するだけで、新たなバイナリを簡単に生成することができる。

例として、前述の表を制約としてバイナリを生成する場合の手順を以下に示す。まず、機種に CSLFD、機能にファイル転送機能を生成条件として指定した場合、ファイル転送機能には FDD ドライバが必要になるが、これは CSLFD に組み込まれる機能であるため、それらを組み込むための `define` の定義やソースファイルの指定を含む Makefile を自動生成する。また、機種に CSU、機能にプリンタ制御機能を生成条件として指定した場合、プリンタ制御機能にはパラレルポートドライバが必要となるが、CSU には組み込まないため、生成システムは機種に問題がある旨をユーザに報告する。

3.2 バイナリ管理

RDBMS [3] を用いて個々のバイナリの対応機種、機能、ソース情報、生成条件、バグ情報等を管理する。各情報はそれぞれ別のテーブルで管理し、各テーブルの ID を格納する別のテーブルを用いてレコー

ド同士を関係付ける [4] (図 2)。但し、ソース情報としてはバージョンだけを記述しておき、パッチなどの情報は RCS から直接入手する。このようにソース管理との連携を行うことで、ソースに関する情報の入力をほとんど省略できることになる。

バイナリ管理を用いることで、トラブル発生時に、既知のトラブルかどうか、バグであれば修正済みであるかどうか、基のソースファイルなどの情報を収集することができ、製品出荷後のトラブル対処の効率向上に役立つことが期待される。

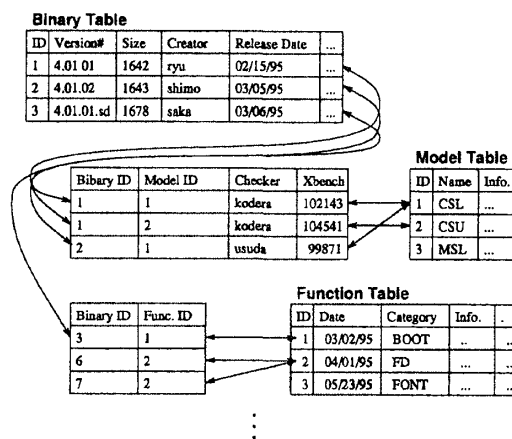


図 2: 各情報テーブルとその関連

4 まとめ

異性体ソフトウェア群の特徴、問題点を明確にし、バイナリを生成、管理するための手法を提案した。この手法に基づくバイナリ生成・管理システムは現在開発中である。今後の課題としては、実際の開発現場でフィールドテストを行い、生成・管理システムの導入による効果を測定することが挙げられる。

参考文献

- [1] 白田尚志, 萩原秀和, 河口信夫, 稲垣康善, “異性体ソフトウェア群のバージョン管理に関する一提案”, 1996 年度電気関係学会東海支部連合大会論文集, p.285(1996).
- [2] Walter F. Tichy, “RCS - A System for Version Control”, Software-Practice and Experience, Vol.15(7), pp.637-654 (1985).
- [3] Empress RDBMS
<http://www.empress.com/store/store1.htm#RDBMS>
- [4] Setrag Khoshafian, “オブジェクト指向データベース” 3.4 従来のデータベースとデータモデル, 共立出版, pp.89-96 (1995).