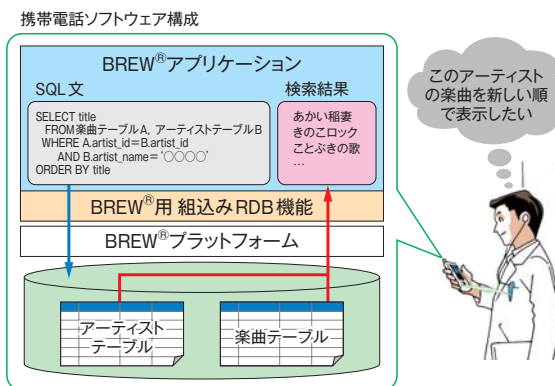


4 ソフトウェア

● BREW[®]用 組込みRDB機能

携帯電話のアプリケーションプラットフォームであるBREW[®]上で高機能なデータ管理を実現するための、組込みRDB (Relational DataBase) 機能を開発した。

BREW[®]は、Qualcomm社製の携帯電話チップセット用のソフトウェア実行環境である。開発した組込みRDB機能は、オープンソースソフトウェアをベースとしてBREW[®]独自の制約に対応させるために大幅に修正し、この環境上で動作するようにしている。これにより、BREW[®]に本格RDB機能を付加することができた。また、高速で安全なデータ管理機能を実現でき、BREW[®]アプリケーションの開発効率向上も期待できる。



SQL : Structured Query Language

BREW[®]用 組込みRDB機能の概要

System architecture of embedded relational database for BREW[®]

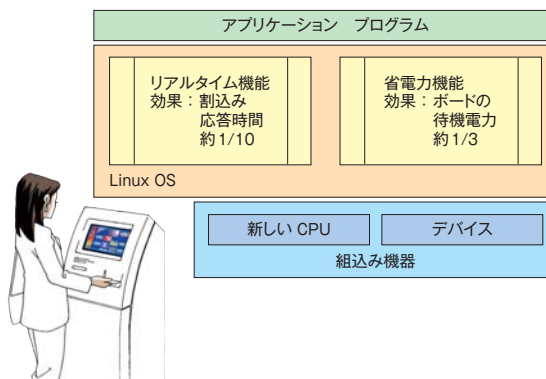
● 組込みLinuxの機能強化

Linuxオペレーティングシステム(OS)は、豊富な機能を持ち広い応用分野があるが、組込み機器へ使用するには追加しなければならない機能がある。

今回、リアルタイム性能を向上させるため、割込み処理、スケジューリング、及びタイマ高精度化に関して、機器の特性に合わせた機能強化を行った。これにより応答時間を従来の約1/10に短縮することができ、すばやい反応が求められる機器や、限られた時間間隔で繰り返し処理する機器の性能向上が期待できるようになった。

また、省電力向けの機能強化を行い、ボードの待機電力を従来の約1/3に削減できることを確かめた。

今後もLinux開発者のコミュニティと連携し、組込み機器向け機能強化に取り組んでいく。



LinuxOSの組込み機器向け拡張例

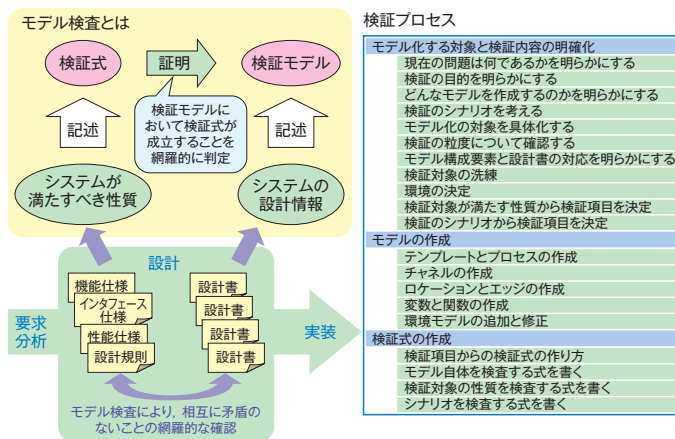
Experimental enhancement of Linux operating system functions

● ソフトウェア開発現場へのモデル検査適用ガイドライン

ソフトウェア設計の高品質化技術として注目されているモデル検査を、組込みソフトウェアの設計プロセスに組み込む手法を確立し、ガイドラインを作成した。

このガイドラインの特長は、以下を明確化したことである。

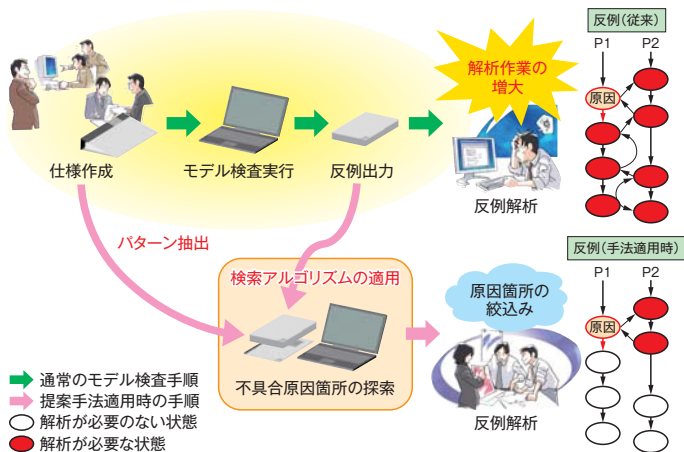
- (1) モデル検査のために必要な設計成果物
 - (2) どの設計成果物から何が検証できるのか。更に効果的な検証のために重要となる“システムが満たさなければならない性質”を設計成果物から導出する手法
 - (3) 設計情報を漏れなく検証モデルに反映させるための留意点と、検証したい性質に応じた検証式の書き方
- このガイドラインに従って、設計時にモデル検査を適用し、活用できることを確認した。今後適用事例を拡大していく。



ソフトウェア設計プロセスへのモデル検査適用方針

Strategy for application of model checking to software design process

● 効率的モデル検査手法によるソフトウェア品質向上



提案手法適用時及び従来のモデル検査手順と解析作業の比較
Comparison between conventional and newly proposed model checking processes

設計品質向上の手段として注目されているモデル検査技術について、適用を効率化する手法を開発した。

モデル検査では、不具合を検出して動作ログ(反例)を出力するが、膨大な出力があると原因解析に時間を要するという課題があった。この課題を解決するため、並列プロセスで発生する不具合原因を類型化することで、原因箇所を絞り込むためのパターンを抽出し、パターンによる検索を機械的に実行するアルゴリズムを考案した。

この手法の適用により、不具合原因と無関係な箇所の解析を排除でき、解析作業の効率化が約6倍向上した。

関係論文：東芝レビュー. 62, 9, 2007, p.46-49.

● 直交表を利用したソフトウェアテスト

直交表の適用領域, 適用のタイミングなどを計画

ガイド(実践的適用方法やノウハウ)

因子と水準の特定, 禁則条件の設定

テストケース生成ツール

テストケース(抜粋)

目標値	部分検定	検定方法	検定結果	初回不具合数
0%	行わない	目数	行わない	test01_bug
85%	=	時間	行う	test01_bug
100%	<	時間	行う	test01_bug
0%	行わない	時間	行う	test01_bug
85%	=	時間	行う	test01_bug
100%	<	時間	行う	test01_bug
0%	行わない	目数	行わない	test01_bug
85%	=	時間	行わない	test01_bug
100%	<	時間	行わない	test01_bug

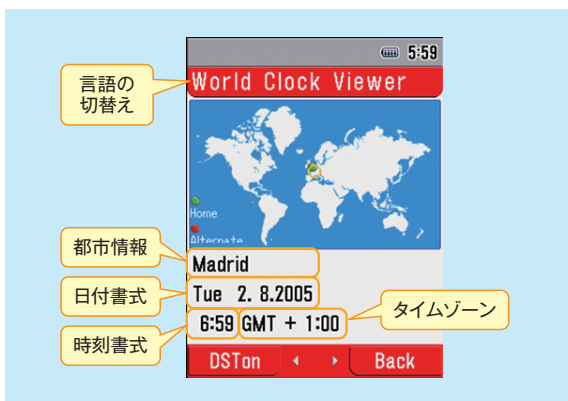
直交表を利用したソフトウェアテストの概要
Overview of software testing using orthogonal array

ソフトウェアにおける組合せテストは、ソフトウェアの規模に対して組合せ数が指数関数的に増加する。これを合理的に抑える方法として直交表の活用が有効である。そこで当社は、直交表を利用したテストケース生成ツールを開発し、実践的適用方法やノウハウをまとめたガイドを作成した。

従来、2因子間の組合せテスト生成と実行の作業は非常に時間を要していたが、このツールでは、禁則条件(実現できない組合せ)などを設定したうえで、2因子間の組合せを100%満たすテストケースが自動生成できるので、テストの効率化と大幅な時間短縮ができる。

今後、このツールとガイドを広く展開して、テストの設計から実行まで品質確認の精度を高め、信頼性のより高いソフトウェア開発を進めていく。

● 携帯電話のマルチリンガル化機能



携帯電話のマルチリンガル化の例
Example of multilingual cellular phone display

携帯電話の海外展開を踏まえ、多種多様な地域や言語に応じて効率的に開発を行う、マルチリンガル化機能を開発した。

携帯電話向けソフトウェアでのマルチリンガル化とは、国際化と多言語化を意味する。前者により、ソフトウェアを複数の国や地域の人でも同じように利用できるようになり、後者により、ソフトウェア上で様々な文字や言語などの文化的習慣を混在させて利用できるようになる。この機能により、国や地域又は言語ごとに異なる日付書式やタイムゾーンなどのデータを、個々のアプリケーションから分離し、利用時にデータを切り替えることで、様々な言語や地域で利用可能な携帯電話を高品質かつ短期間で提供できるようにした。