



概要

○自己説明型コンポーネント

信頼性（安全性）の高いシステムを効率良く開発するには、理解しやすく、扱いやすいコンポーネントの利用が不可欠です。開発者が必要とするコンポーネントの情報を効果的かつリアルタイムに提供するため、高度の自己説明性を備えたのが自己説明型コンポーネントです。

○本技術の独自性

- ・ 処理の意図やアルゴリズムなど、プログラムの「意味」を記録できる言語群
- ・ 情報量を制御して理解を助ける、適応型インタフェース技術

○効果

自己説明型コンポーネントを用いた開発では、作成したいプログラムのアルゴリズム的性質を指定することで開発を行います。既存のプログラミング言語では記録できない、プログラムの内部動作に関する意味的情報が記録されるので、後のレビュー、デバッグ、再利用時の効率向上につながります。また、この意味的情報を用いてコンポーネントの安全性評価をより高い次元で行うことも可能となります。

実用化の可能性

＜次世代プログラミング言語と開発環境＞

計算の流れなどのアルゴリズム的要素の記述と具体的な処理内容の記述を高度にバランスした全く新しいプログラミングツール

＜再利用を促進するコンポーネント・ライブラリ＞

さまざまな機能を自己説明型コンポーネントのライブラリとして提供

＜コミュニケーションツール＞

ソフトウェアの仕様や動きを顧客と確認したり、開発者同士で共有するための記述言語

UBICからのメッセージ

○現状のプログラミングでは、コンピュータにやってもらいたい事（人の考え）をプログラマーがプログラミング言語に翻訳して命令文の形でコンピュータに指示します。この一連の命令文がプログラムです。翻訳に人が介入するので「誤り」を皆無にするのは困難です。

○この技術は「人の考え」をコンピュータがプログラミング言語に翻訳しプログラムを作ります。ですから、翻訳に起因した「誤り」は皆無で「人の考え」を正確に記述できればバグの無いプログラムを作ることが出来、さらに、プログラムに関して「人の考え」上で議論できるので現状の「プログラム」上での議論よりもずっと効率的です。

研究概要図

