トップエスイー修了制作

組込みソフトウェアへのKobrA法の適用

キヤノン株式会社　仲 顕照　naka.akiteru@canon.co.jp

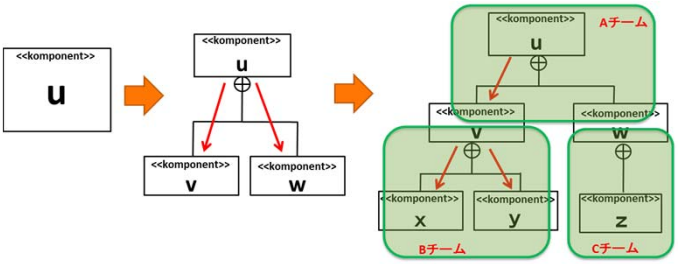
コンポーネント分割における課題

大規模ソフトウェアを品質よく開発するためには，システム全体を複数のソフトウェア部品（コンポーネント）に分割し，コンポーネント単位で品質を確保する必要がある．しかしながら，複数人で分業を考慮した上で，適切なコンポーネント分割することは困難である．

KobrA法による解決

KobrA法は，トップダウン的にコンポーネント分割を行い，コンポーネントをツリー構造で構築するため，複数人での分業を考慮したコンポーネント分割が可能である．組込みソフトウェアに対して，KobrA法を適用した場合のフィージビリティスタディを行い，並行性に関する拡張を行った．

KobrA法の適用と課題



コンポーネントが満たすべき仕様を厳密に定義　仕様

上位のコンポーネントが定義した仕様を満たす範囲で実現方法を検討　実現

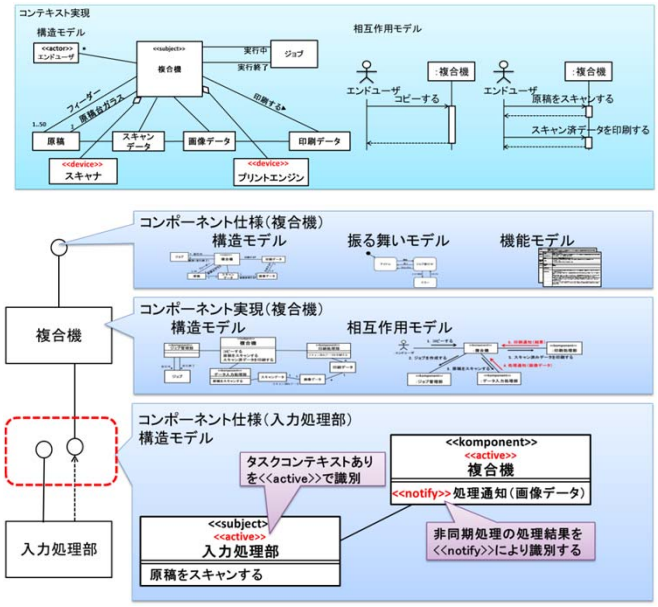
コンポーネントの単位で分業を可能にする

どのコンポーネントが並行動作可能なのかを設計できていない

並行性を考えると、当然必要となる非同期通信のことが考慮されていない

並行性や非同期通信のことを考えると、分割指針が不十分

KobrA法の表記の追加



⇒ 並行性（<<active>>, <<passive>>）や非同期通信（<<notify>>）を表現するための表記を追加し、分割指針にも利用する

分割指針へのパターンの適用

名前	分割の階層に関するパターン
状況	コンポーネント分割の階層数を決める
問題	コンポーネント分割によって処理の委譲を行っていく必要がある
解決	・上位のコンポーネントによる中継がある場合は、階層数優先 ・上位のコンポーネントによる中継がない場合は、参照数優先 階層数優先　参照数優先
結果	適切なコンポーネント分割ができる
フォース	・一般的に階層数を増やさない方が、全体的にコンポーネント数も減り、シンプルな設計とできる ・階層を作ることで処理の委譲が進む

名前	共通利用するコンポーネントに関するパターン
状況	複数のコンポーネントから共通に利用されるコンポーネントをどの位置に分割するかを決める
問題	適切な依存関係を持ったコンポーネント分割が必要である
解決	・共通利用されるコンポーネントがパッシブならば、独立分割方式 ・共通利用されるコンポーネントがアクティブならば、従属分割方式 独立　従属
結果	適切なコンポーネント分割ができる
フォース	・並行性を考慮すると、独立分割方式だと、タイミング制御によって複雑な依存関係が発生する場合がある ・独立分割方式の方が、一般的にはシンプルな設計にできる

⇒ パターン適用により並行性を考慮した分割結果を得る