



# 無線センサネットワークにおける非機能要求の優先順位によるアプリケーションの実行継続

日本ユニシス株式会社/  
総合研究大学院大学

末永 俊一郎  
Shunichiro Suenaga

suenaga@nii.ac.jp  
shunichiro.suenaga@unisys.co.jp

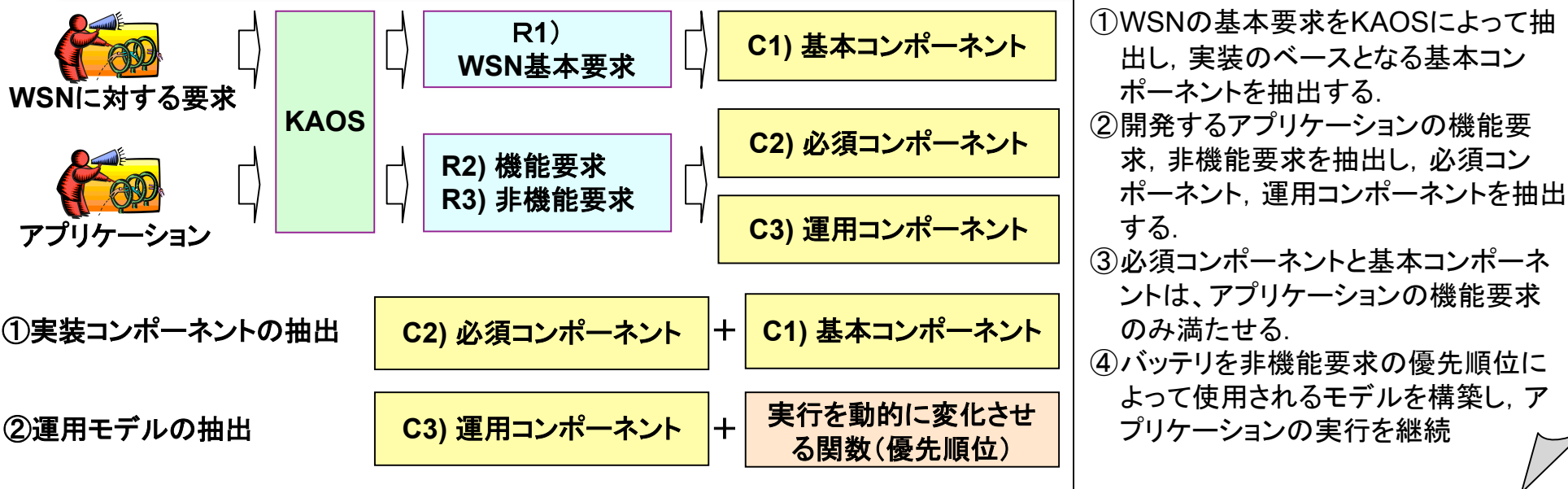
## WSNアプリケーション開発の課題

1. 開発をアドホックに実施すると、プログラムメモリを無駄に消費して効率が悪い
2. バッテリーが減少した際、アプリケーションの実行に重要でない処理がバッテリーを消費し、肝心の処理が実行できない場合がある。

## 課題に対する解決策

1. KAOSによってアプリケーションの機能要求,非機能要求を抽出し,実装時に必要となるコンポーネントを実装前に抽出。
2. KAOSから非機能要求を抽出し,優先順位をつけて動的に変化するバッテリーに合わせアプリケーションの振る舞いを変更する。

## 手法概要



## 運用モデル(関数)

非機能要求の優先順位から処理を実行する強さをモデル化

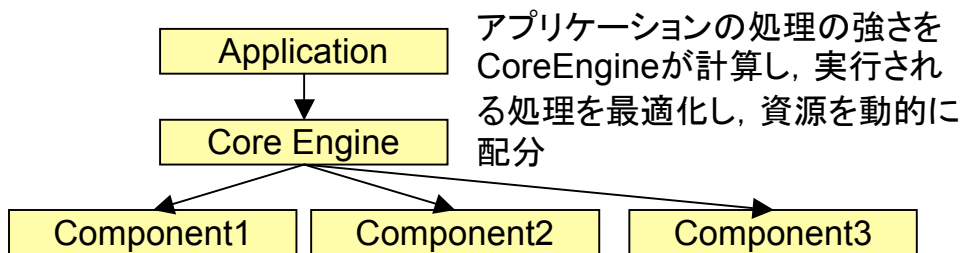
Priority : 非機能要求の優先順位

Constraints : 全ノードが実行できないと意味を成さない処理

```

if (Constraints == NONE) {
    Strength = C ×  $\frac{BatteryLevel}{Priority \times PowerConsumption}$ 
}

```



### ■実装環境

TinyOS 2.0, NesC, TOSSIM, Xubuntos, Eclipse

## 評価・結論

### ■評価項目

アプリケーション稼動時間の違い

提案手法 VS 動的に変更しない場合

### KAOSの利用価値

KAOSを用いない場合のコンポーネント選択

KAOSを利用したコンポーネント選択

### ■結論

- バッテリーの残量に応じて,各処理ごとに資源を動的に割り当てることで,アプリケーションの必須機能の実行を長引かせることができる。
- 実装前に実装すべきコンポーネントを抽出することで,無駄なコンポーネントの実装を避けてコード容量を減らすことができる。

(WSNのように)資源制約を考慮しながら開発・運用を行う際,有力な開発アプローチの一つとなる