Minimizing System Modification in an Incremental Design Approach

Pop, Paul; Eles, Petru; Pop, Traian; Peng, Zebo

Published in: Ninth International Symposium on Hardware/Software Codesign. CODES 2001 (IEEE Cat. No.01TH8571)

Link to article, DOI: 10.1109/HSC.2001.924672

Publication date: 2001

Document Version Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Minimizing System Modification in an Incremental Design Approach

Paul Pop, Petru Eles, Traian Pop, Zebo Peng

Department of Computer and Information Science
Linköpings universitet, Sweden
Incremental Design Process

- Start from an already existing system with applications
- Implement new functionality on this system
  - Mapping and Scheduling

- To reduce design and testing time:
  - As few as possible modifications of the existing applications

- After the new functionality has been implemented:
  - It should be easy to add functionality in the future
Mapping and Scheduling Problem

Modify (re-map) so that the current applications will fit.

Map and schedule so that the future applications will have a chance to fit.

Do not exist yet at Version N!
Problem Formulation

Input

- A set of *existing* applications modelled using process graphs.
- A *current* application to be mapped modelled using process graphs.
- Each process graph in the application has its own *period* and *deadline*.
- Each process has a *potential set of nodes* to be mapped on and a *WCET*.
- The system architecture is given.

Output

- A mapping and scheduling of the *current* application, so that:
  - **Requirement a:** constraints of the *current* application are satisfied and minimal modifications are performed to the *existing* applications.
  - **Requirement b:** new *future* applications can be mapped on the resulted system.

Notes

- Hard real-time applications
- Static cyclic scheduling of processes and messages
- Time-triggered protocol, TDMA
Mapping and Scheduling Strategy

- Initial mapping and scheduling
  
  a) - Satisfying the constraints for the current application
      - Minimizing the modification cost
  
  b) - Prediction of success in adding future applications
      - Minimizing the objective function


\[ C = w_1^P (C_1^P) + w_1^m (C_1^m) + w_2^P \max(0, t_{need} - C_2^P) + w_2^m \max(0, b_{need} - C_2^m) \]
Characterizing Existing Applications

\[ R(\{\Gamma_7\}) = 20, \quad R(\{\Gamma_3\}) = 50, \quad R(\{\Gamma_3, \Gamma_7\}) = 70, \]
\[ R(\{\Gamma_4, \Gamma_7\}) = 90 \text{ (the modification of } \Gamma_4 \text{ triggers the modification of } \Gamma_7), \]
\[ R(\{\Gamma_2, \Gamma_3\}) = 120, \quad R(\{\Gamma_3, \Gamma_4, \Gamma_7\}) = 140, \quad R(\{\Gamma_1\}) = 150, \quad \ldots \]

The total number of possible subsets is 16.
Mapping and Scheduling, Requirement a)

- Mapping and scheduling of the *current* application, so that:
  Constraints of the *current* application are satisfied and minimal modifications are performed to the *existing* applications.

- Subset selection problem
  Select that subset $\Omega$ of existing applications which guarantees that the current application fits and the modification cost $R(\Omega)$ is minimized:

$$R(\Omega) = \sum_{\Gamma_i \in \Omega} R_i$$
Mapping and Scheduling Strategy

- Initial mapping and scheduling

- Requirement a)
  Minimizing the modification cost \( R(\Omega) \), subset selection:
  - Exhaustive Search (ES)
  - Ad-Hoc Solution (AH)
  - Subset Selection Heuristic (SH)

- Requirement b)
  Minimizing the objective function:
Experimental Results

Average Modification Cost $R(\Omega)$

- **AH**
- **SH**
- **ES**

Number of processes
Conclusions

- Mapping and scheduling of distributed embedded systems for hard-real time applications.

- Incremental design process
  - Already existing system,
  - Implement new functionality,
  - a) Existing system modified as little as possible,
  - b) new functionality can be easily added to the system.

- Mapping strategy
  - a) Subset selection to minimize modification cost,
  - b) Two design criteria, objective function.