Construction Approaches for Component-Based Systems

PhD Thesis

Andreea Vescan
Supervisor: Prof. Dr. Militon Frenţiu

Babeş-Bolyai University
Cluj-Napoca

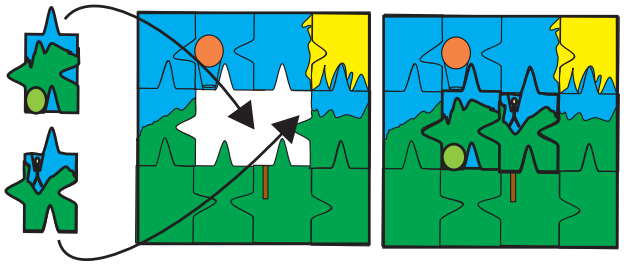
2008

Setting the context

- Component-Based Software Engineering
- Component integration and Component composition

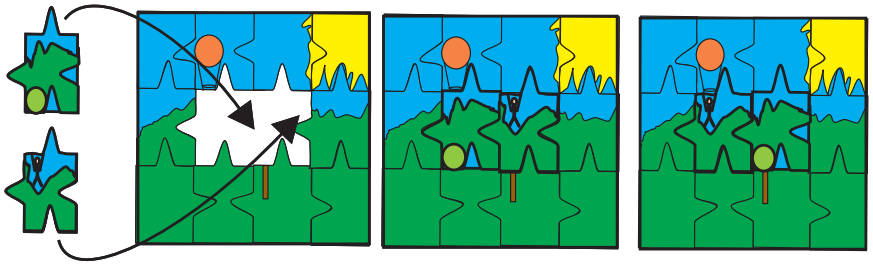
Setting the context

- Component-Based Software Engineering
- **Component integration** and Component composition



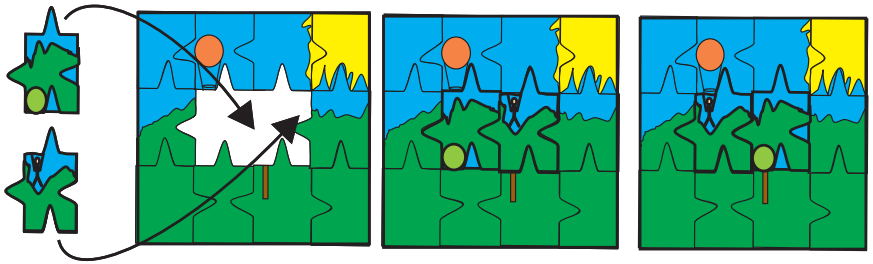
Setting the context

- Component-Based Software Engineering
- Component integration and **Component composition**



Setting the context

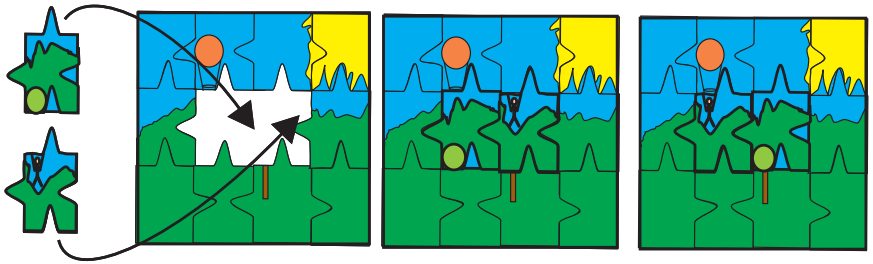
- Component-Based Software Engineering
- Component integration and Component composition



- Complete system construction
  - Component Selection Problem

Setting the context

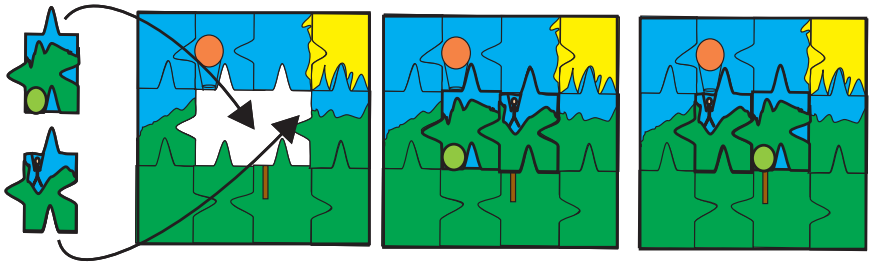
- Component-Based Software Engineering
- Component integration and Component composition



- Complete system construction
  - Component Selection Problem
- Partial system construction. Component Adaptation

Setting the context

- Component-Based Software Engineering
- Component integration and Component composition



- Complete system construction
  - Component Selection Problem
- Partial system construction. Component Adaptation
- Metrics in Component-Based Software Engineering

Component Selection Problem

Component Selection Problem

Consider the set $SR = \{r_1, r_2, ..., r_n\}$,
and the set $SC = \{c_1, c_2, ..., c_m\}$.
Each component c_i can satisfy a subset of the requirements from SR,

$$SR_{c_i} = \{r_{i_1}, r_{i_2}, ..., r_{i_k}\}.$$

The goal is to find a set of components Sol in such a way that every requirement r_j from the set SR may have assigned a component c_i from Sol where r_j is in SR_{c_i}.

Component Selection Problem

Consider the set $SR = \{r_1, r_2, ..., r_n\}$,
and the set $SC = \{c_1, c_2, ..., c_m\}$.
Each component c_i can satisfy a subset of the requirements from SR,

$$SR_{c_i} = \{r_{i_1}, r_{i_2}, ..., r_{i_k}\}.$$

The goal is to find a set of components Sol in such a way that every requirement r_j from the set SR may have assigned a component c_i from Sol where r_j is in SR_{c_i}.

- Simple Component Selection Problem ([Fox et al., 2004])
- Multicriteria-based Component Selection Problem ([Haghpanah et al., 2007])

Simple Component Selection Problem

Simple Component Selection Problem approaches

Simple Component Selection Problem

Simple Component Selection Problem approaches

- Backtracking-based composition approaches

Simple Component Selection Problem

Simple Component Selection Problem approaches

- Backtracking-based composition approaches
- Automata-based composition approaches

Simple Component Selection Problem

Simple Component Selection Problem approaches

- Backtracking-based composition approaches
- Automata-based composition approaches
- Artificial intelligence-based composition approaches

Backtracking-based composition approaches

Backtracking-based composition approaches

- All Possible Components Configurations (APCC) algorithm
  [Fanea and Motogna, 2004]

Backtracking-based composition approaches

- All Possible Components Configurations (APCC) algorithm [Fanea and Motogna, 2004]
- Temporal Components Composition Restraint (TCCR) algorithm [Vescan, 2006] (ISI Proceeding), [Vescan, 2007b]

Backtracking-based composition approach (cont.)

- Source component

Backtracking-based composition approach (cont.)

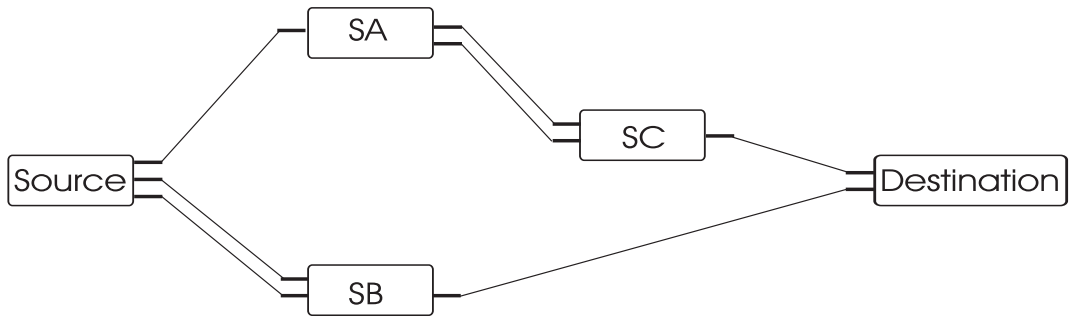
- Source component
- Destination component

Backtracking-based composition approach (cont.)

- Source component
- Destination component
- Simple component

Backtracking-based composition approach (cont.)

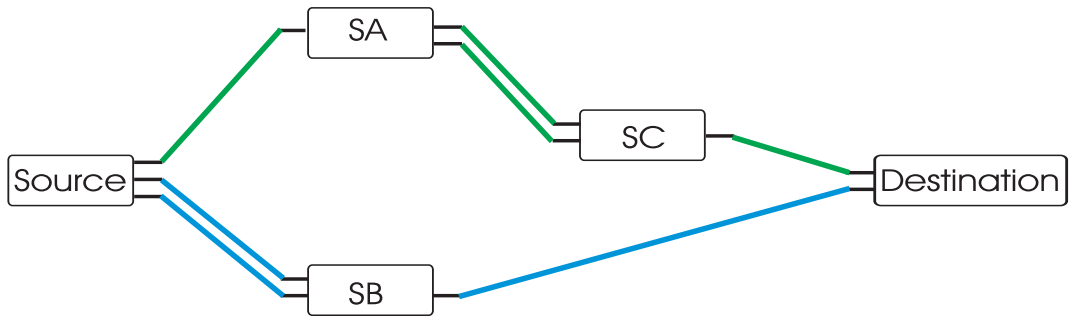
- Source component
- Destination component
- Simple component
- Compound component



Backtracking-based composition approach (cont.)

- Source component
- Destination component
- Simple component
- Compound component

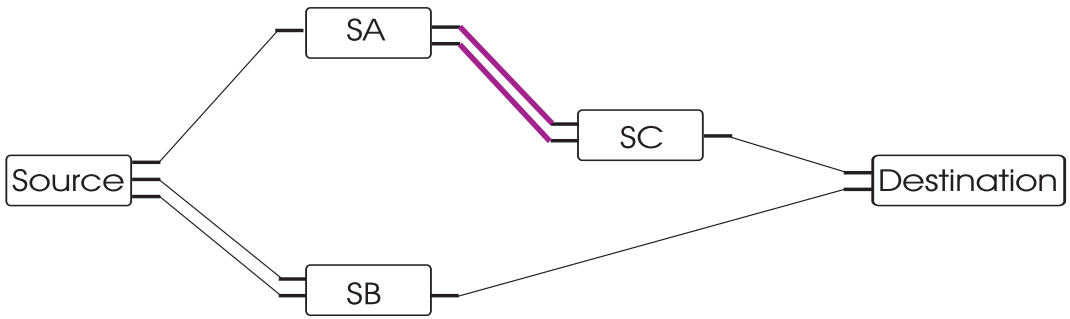
- Parallel composition - $SA \parallel SB$



Backtracking-based composition approach (cont.)

- Source component
- Destination component
- Simple component
- Compound component

- Parallel composition - $SA \mid\mid SB$
- Serial composition - $SA + SC$
([Motogna and Parv, 2002])



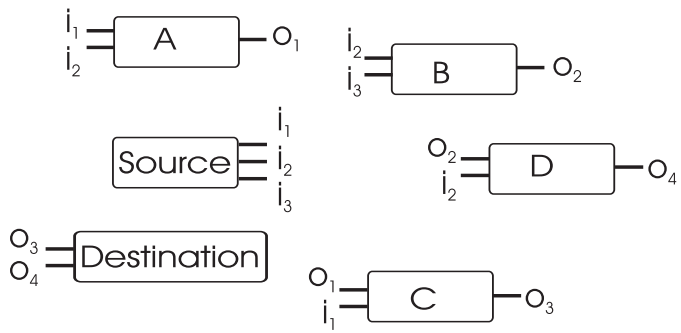
Backtracking-based composition approach (cont.)

All Possible Components Configurations (APCC) algorithm

Backtracking-based composition approach (cont.)

All Possible Components Configurations (APCC) algorithm

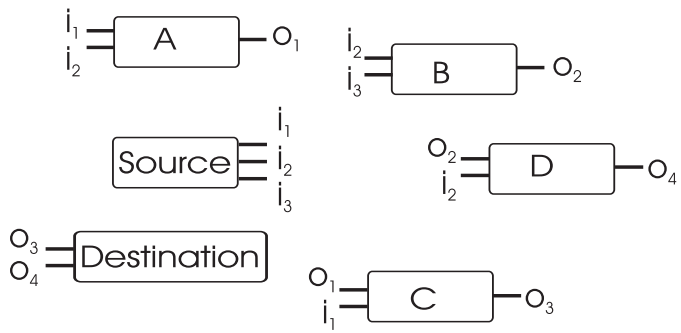
- Steps:



Backtracking-based composition approach (cont.)

All Possible Components Configurations (APCC) algorithm

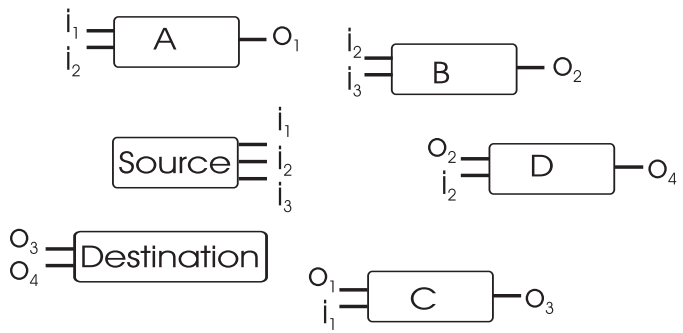
- Steps:
- Check disjoint in/out condition;



Backtracking-based composition approach (cont.)

All Possible Components Configurations (APCC) algorithm

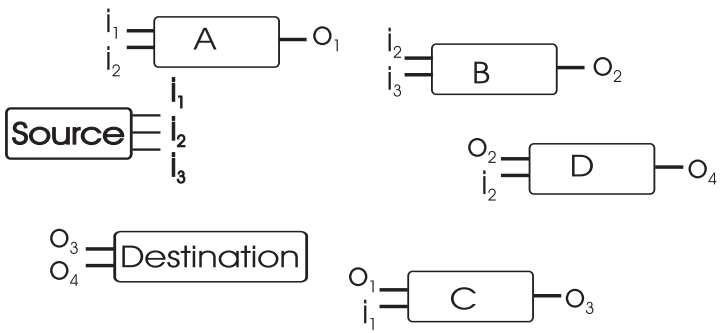
- Steps:
- Check disjoint in/out condition;
- Compute the interdependencies: $A + C$ and $B + D$.



Backtracking-based composition approach (cont.)

All Possible Components Configurations (APCC) algorithm

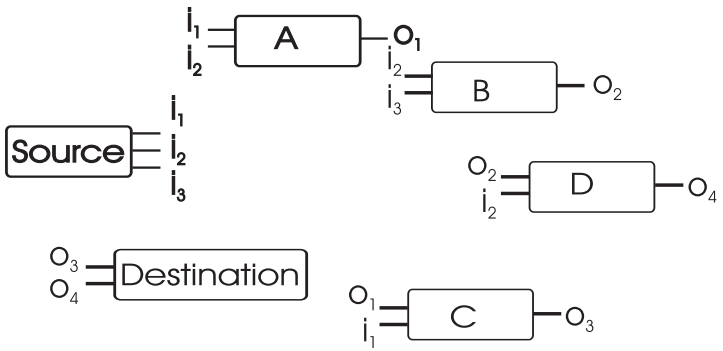
- Steps:
- Check disjoint in/out condition;
- Compute the interdependencies: $A + C$ and $B + D$.
- APCC algorithm



Backtracking-based composition approach (cont.)

All Possible Components Configurations (APCC) algorithm

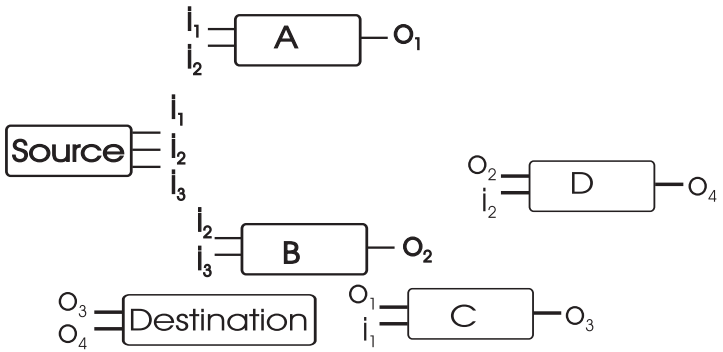
- Steps:
- Check disjoint in/out condition;
- Compute the interdependencies: $A + C$ and $B + D$.
- APCC algorithm



Backtracking-based composition approach (cont.)

All Possible Components Configurations (APCC) algorithm

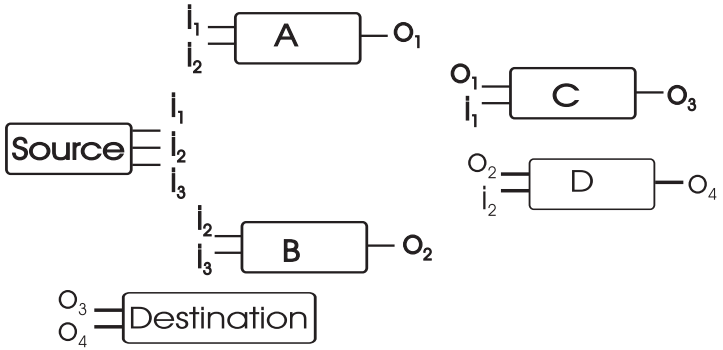
- Steps:
- Check disjoint in/out condition;
- Compute the interdependencies: $A + C$ and $B + D$.
- APCC algorithm



Backtracking-based composition approach (cont.)

All Possible Components Configurations (APCC) algorithm

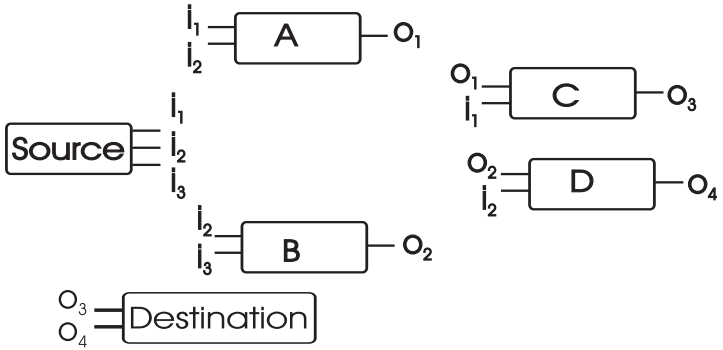
- Steps:
- Check disjoint in/out condition;
- Compute the interdependencies: $A + C$ and $B + D$.
- APCC algorithm



Backtracking-based composition approach (cont.)

All Possible Components Configurations (APCC) algorithm

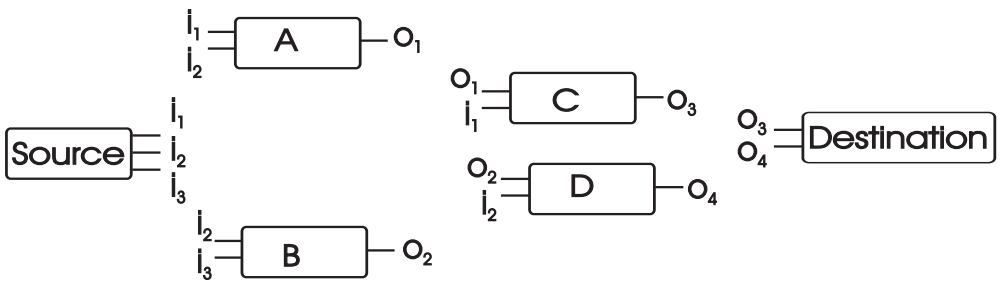
- Steps:
- Check disjoint in/out condition;
- Compute the interdependencies: $A + C$ and $B + D$.
- APCC algorithm



Backtracking-based composition approach (cont.)

All Possible Components Configurations (APCC) algorithm

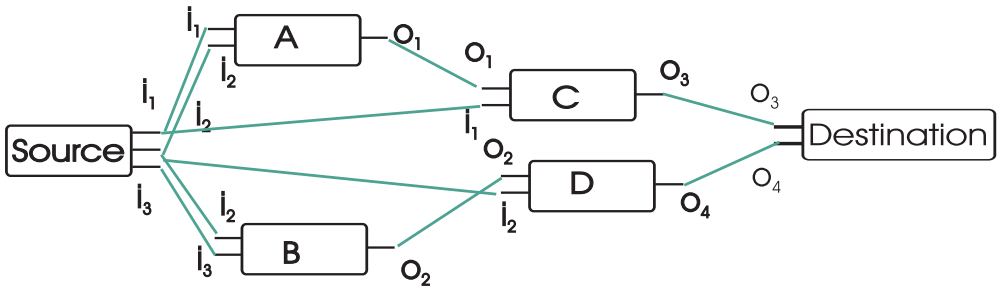
- Steps:
- Check disjoint in/out condition;
- Compute the interdependencies: $A + C$ and $B + D$.
- APCC algorithm



Backtracking-based composition approach (cont.)

All Possible Components Configurations (APCC) algorithm

- Steps:
- Check disjoint in/out condition;
- Compute the interdependencies: $A + C$ and $B + D$.
- APCC algorithm



Solutions:
1) ((A||B)+C)+D
2) ((A||B)+D)+C
3) ((A+C)||B)+D
4) ((B||A)+C)+D
5) ((B||A)+D)+C
6) ((B+D)||A)+C

Backtracking-based composition approach (cont.)

Temporal Components Composition Restraint (TCCR) algorithm

Backtracking-based composition approach (cont.)

Temporal Components Composition Restraint (TCCR) algorithm

- **Data composition restraint** $CSender \xrightarrow{data} CReceiver$.

Backtracking-based composition approach (cont.)

Temporal Components Composition Restraint (TCCR) algorithm

- **Data composition restraint** $CSender \xrightarrow{data} CReceiver$.
- **Temporal composition restraint** $CPrevious \xrightarrow{before} CAfter$.

Backtracking-based composition approach (cont.)

Temporal Components Composition Restraint (TCCR) algorithm

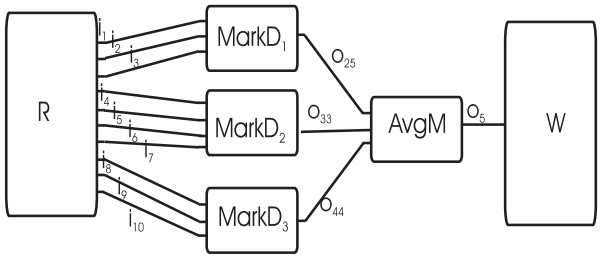
- **Data composition restraint** $CSender \xrightarrow{data} CReceiver$.
- **Temporal composition restraint** $CPrevious \xrightarrow{before} CAfter$.
- Example: Average mark computation system.

Backtracking-based composition approach (cont.)

Temporal Components Composition Restraint (TCCR) algorithm

- **Data composition restraint** $CSender \xrightarrow{data} CReceiver$.
- **Temporal composition restraint** $CPrevious \xrightarrow{before} CAfter$.
- Example: Average mark computation system.

- Data composition restraints: $MarkD_1 \xrightarrow{o_{25}} AvgM$, $MarkD_2 \xrightarrow{o_{33}} AvgM$, $MarkD_3 \xrightarrow{o_{44}} AvgM$.

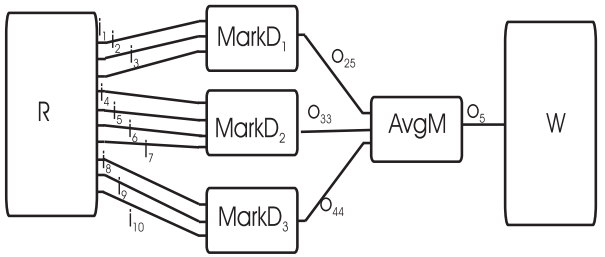


Backtracking-based composition approach (cont.)

Temporal Components Composition Restraint (TCCR) algorithm

- **Data composition restraint** $CSender \xrightarrow{data} CReceiver$.
- **Temporal composition restraint** $CPrevious \xrightarrow{before} CAfter$.
- Example: Average mark computation system.

- Data composition restraints: $MarkD_1 \xrightarrow{o_{25}} AvgM$, $MarkD_2 \xrightarrow{o_{33}} AvgM$, $MarkD_3 \xrightarrow{o_{44}} AvgM$.
- Temporal composition restraint: $MarkD_2 \xrightarrow{before} MarkD_3$.

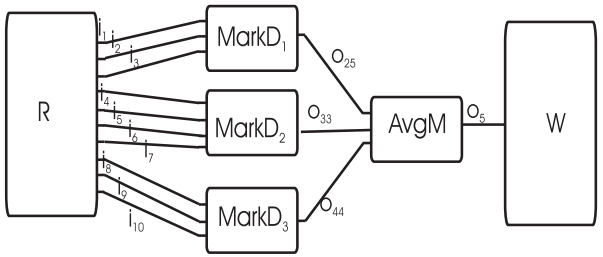


Backtracking-based composition approach (cont.)

Temporal Components Composition Restraint (TCCR) algorithm

- **Data composition restraint** $CSender \xrightarrow{data} CReceiver$.
- **Temporal composition restraint** $CPrevious \xrightarrow{before} CAfter$.
- Example: Average mark computation system.

- Data composition restraints: $MarkD_1 \xrightarrow{o_{25}} AvgM$, $MarkD_2 \xrightarrow{o_{33}} AvgM$, $MarkD_3 \xrightarrow{o_{44}} AvgM$.
- Temporal composition restraint: $MarkD_2 \xrightarrow{before} MarkD_3$.



1	R	$MarkD_1$	$MarkD_2$	$MarkD_3$	$AvgM$	W
2	R	$MarkD_2$	$MarkD_1$	$MarkD_3$	$AvgM$	W
3	R	$MarkD_2$	$MarkD_3$	$MarkD_1$	$AvgM$	W

Automata-based composition approaches

Automata-based composition approaches

- MakeAllModels algorithm - Input data-based construction
  [Fanea et al., 2006] (indexed BDI)

Automata-based composition approaches

- MakeAllModels algorithm - Input data-based construction [Fanea et al., 2006] (indexed BDI)
- **C**ontrol**F**low and **D**ata**F**low **S**yntactic **C**omposition algorithm Task-based construction [Vescan and Motogna, 2006b] (indexed Mathematical Reviews), [Vescan and Motogna, 2006a]

Automata-based composition approaches (cont.)

Automata representation ([Parv et al., 2004])

Automata-based composition approaches (cont.)

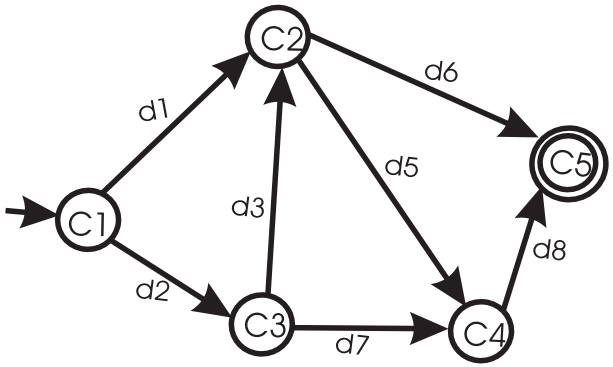
Automata representation ([Parv et al., 2004])

A system of components is defined as a finite automaton $A = (Q, \Sigma, \delta, q_0, F)$, where:

Automata-based composition approaches (cont.)

Automata representation ([Parv et al., 2004])
A system of components is defined as a finite automaton $A = (Q, \Sigma, \delta, q_0, F)$, where:

- Q is the set of states,
- Σ is the input alphabet,
- δ is the transition function,
- q_0 is the initial state,
- F is the set of final states.

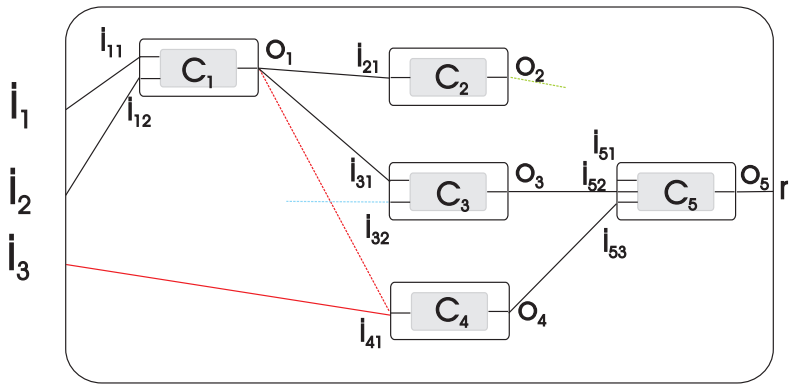


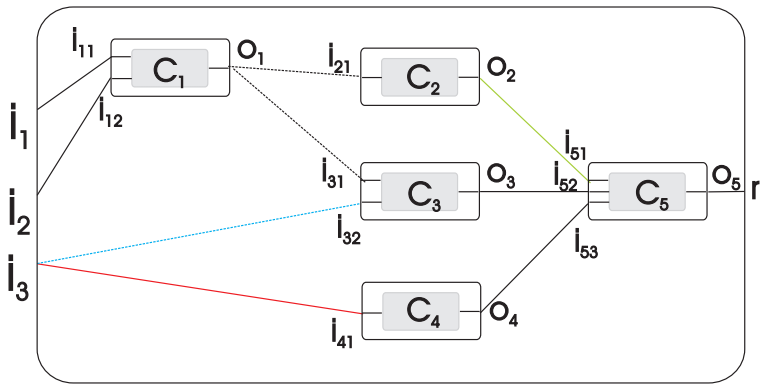
Automata-based composition approaches (cont.)

Composition rules

Automata-based composition approaches (cont.)

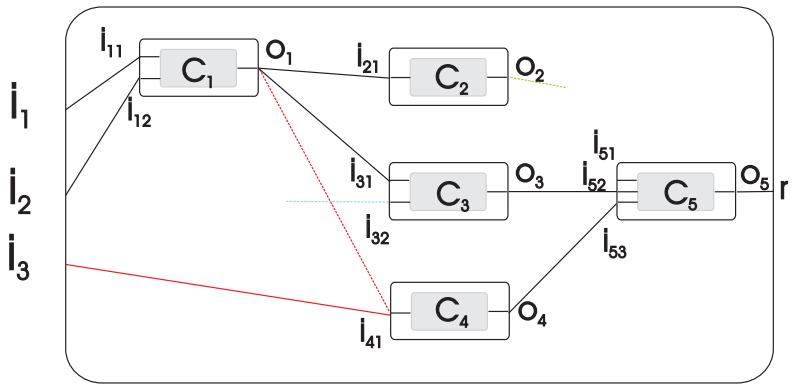
Composition rules

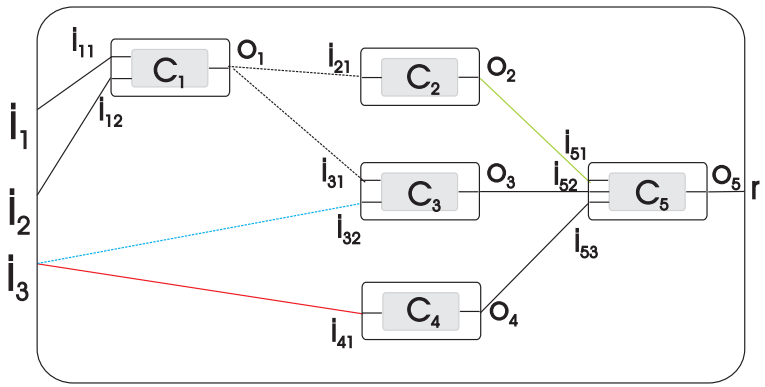




Automata-based composition approaches (cont.)

Composition rules

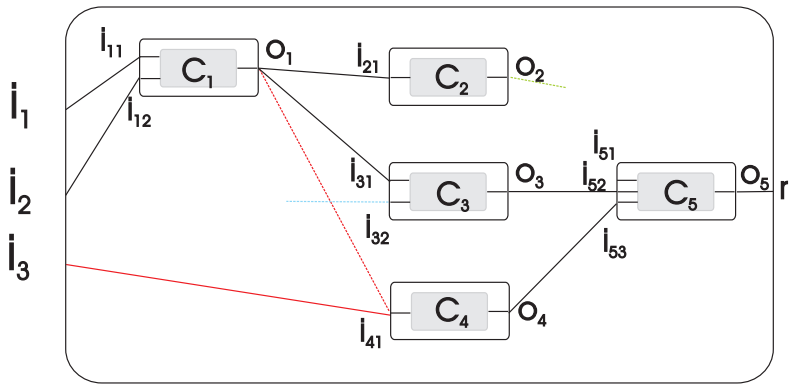


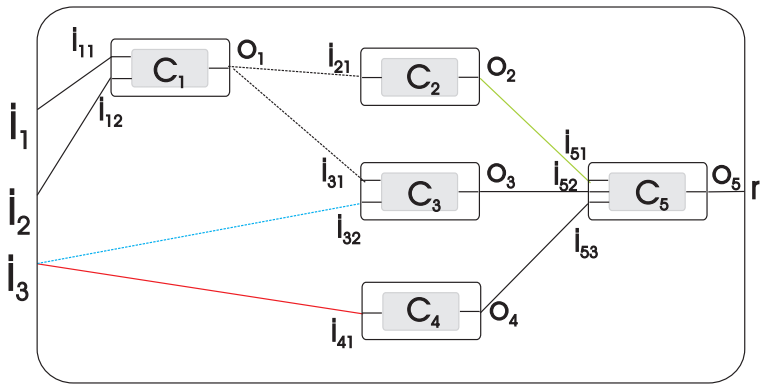


- No lost data

Automata-based composition approaches (cont.)

Composition rules

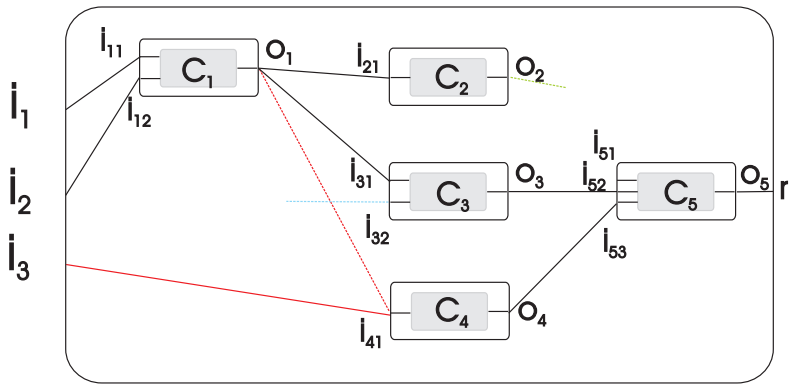


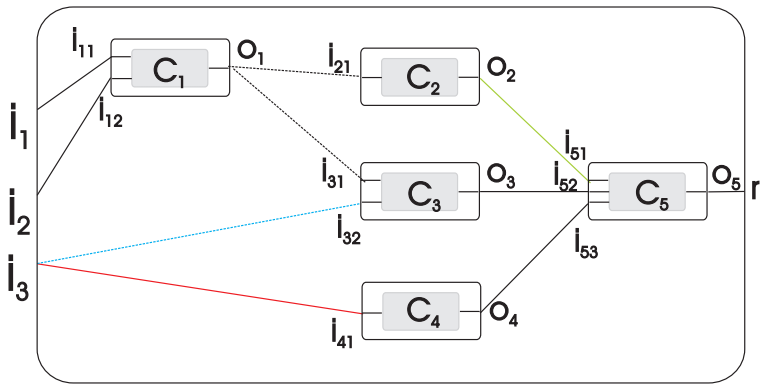


- No lost data
- One provider/inport

Automata-based composition approaches (cont.)

Composition rules





- No lost data
- One provider/inport
- "Broadcasting"

Automata-based composition approach (cont.)

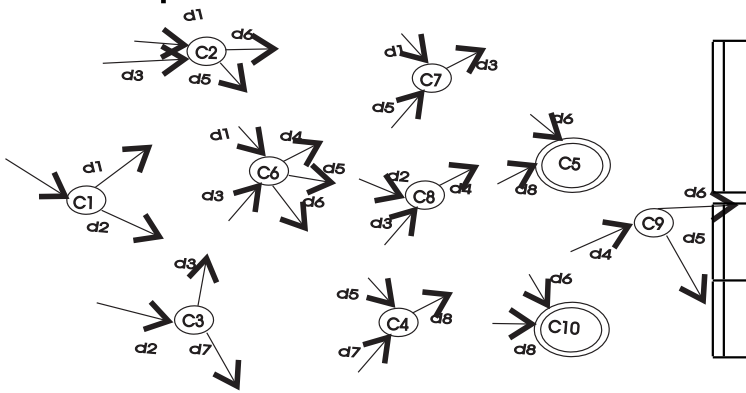
MakeAllModels algorithm

Input data-based construction

Automata-based composition approach (cont.)

MakeAllModels algorithm

Input data-based construction

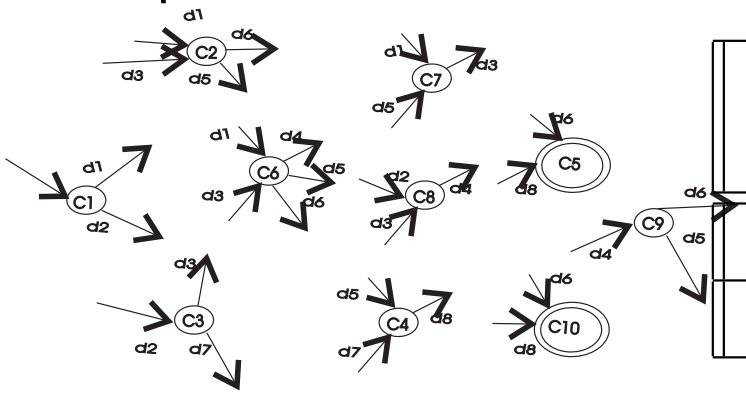


All Sol.	Sol. no Lost Data	Sol. one provider/input	Final
1323	40	64	1
100%	3.02%	4.83%	0.07%

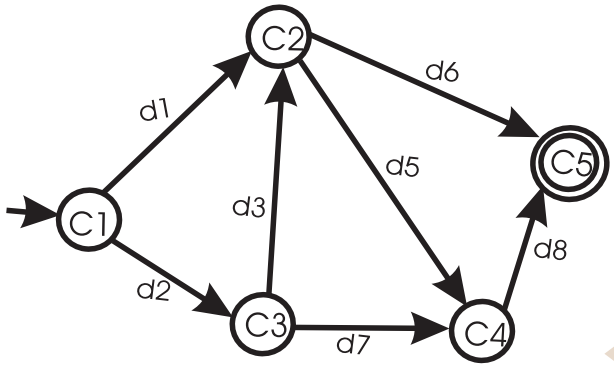
Automata-based composition approach (cont.)

MakeAllModels algorithm

Input data-based construction



All Sol.	Sol. no Lost Data	Sol. one provider/input	Final
1323	40	64	1
100%	3.02%	4.83%	0.07%



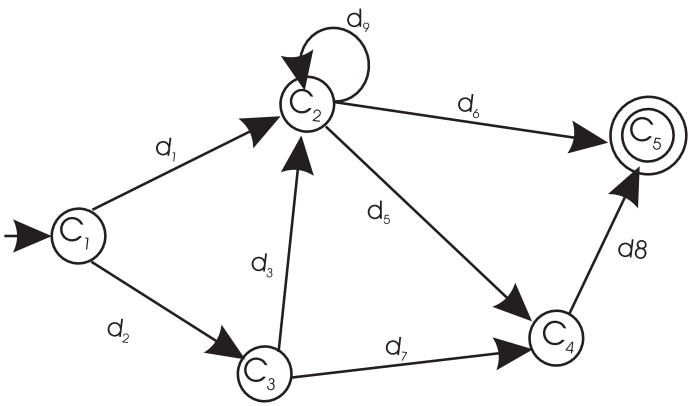
Automata-based composition approach (cont.)

Control**F**low and **D**ata**F**low **S**yntactic **C**omposition algorithm
Task-based construction

Automata-based composition approach (cont.)

Control**F**low and **D**ata**F**low **S**yntactic **C**omposition algorithm

Task-based construction



Sol. no.	Task execution order
1	$r, t_1, t_2, t_4, t_3, t_5, w$
2	$r, t_1, t_2, t_4, t_5, t_3, w$
3	$r, t_1, t_4, t_2, t_3, t_5, w$
4	$r, t_1, t_4, t_2, t_5, t_3, w$
5	$r, t_1, t_4, t_3, t_2, t_5, w$
6	$r, t_4, t_1, t_2, t_3, t_5, w$
7	$r, t_4, t_1, t_2, t_5, t_3, w$
8	$r, t_4, t_1, t_3, t_2, t_5, w$

Artificial intelligence-based composition approach

- Evolutionary Algorithms-based approaches

Artificial intelligence-based composition approach

- Evolutionary Algorithms-based approaches
  - Evolutionary Algorithms to generate Components Execution Order [Fanea and Diosan, 2005a] (indexed BDI)
  - Evolutionary Algorithms to analyze Component Composition [Fanea and Diosan, 2006a] (indexed ISI-SCI-E)

Artificial intelligence-based composition approach

- Evolutionary Algorithms-based approaches
  - Evolutionary Algorithms to generate Components Execution Order [Fanea and Diosan, 2005a] (indexed BDI)
  - Evolutionary Algorithms to analyze Component Composition [Fanea and Diosan, 2006a] (indexed ISI-SCI-E)
- Genetic Programming-based approaches

Artificial intelligence-based composition approach

- Evolutionary Algorithms-based approaches
  - Evolutionary Algorithms to generate Components Execution Order [Fanea and Diosan, 2005a] (indexed BDI)
  - Evolutionary Algorithms to analyze Component Composition [Fanea and Diosan, 2006a] (indexed ISI-SCI-E)
- Genetic Programming-based approaches
  - Multi Expression Programming-based approach [Fanea and Diosan, 2005b]
  - Cartesian Genetic Programming-based approach [Fanea and Diosan, 2006b] (indexed ISI-SCI-E)

Artificial intelligence-based composition approach

- Evolutionary Algorithms-based approaches
  - Evolutionary Algorithms to generate Components Execution Order [Fanea and Diosan, 2005a] (indexed BDI)
  - Evolutionary Algorithms to analyze Component Composition [Fanea and Diosan, 2006a] (indexed ISI-SCI-E)
- Genetic Programming-based approaches
  - Multi Expression Programming-based approach [Fanea and Diosan, 2005b]
  - Cartesian Genetic Programming-based approach [Fanea and Diosan, 2006b] (indexed ISI-SCI-E)
- P-Systems-based approach [Fanea and Diosan, 2006c] (indexed BDI)

Artificial intelligence-based composition approach

Evolving Components Execution Order (ECEO)

Artificial intelligence-based composition approach

Evolving Components Execution Order (ECEO)

- APCC algorithm:
  $((((((((((C_3||C_6)+C_7)||C_1)+C_2)||C_4)+C_9)+C_5)+C_8)+C_{10})+C_{11})$.
- ECEO

$$\frac{Genes:\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11}{Values:2\ 3\ 1\ 3\ 5\ 1\ 2\ 6\ 4\ \ 6\ \ 7}, \quad \frac{Genes:\ 3\ 6\ 1\ 7\ 2\ 4\ 9\ 5\ 8\ 10\ 11}{Values:1\ 1\ 2\ 2\ 3\ 3\ 4\ 5\ 6\ \ 6\ \ 7}.$$

Algorithm	Solutions	Time
APCC	1680	1*sec*
ECEO	3017	1.3*sec*

Table: Experiment: Eleven involved components with ten dependences

Artificial intelligence-based composition approach (cont.)

Evolutionary Algorithms to analyze Component Composition

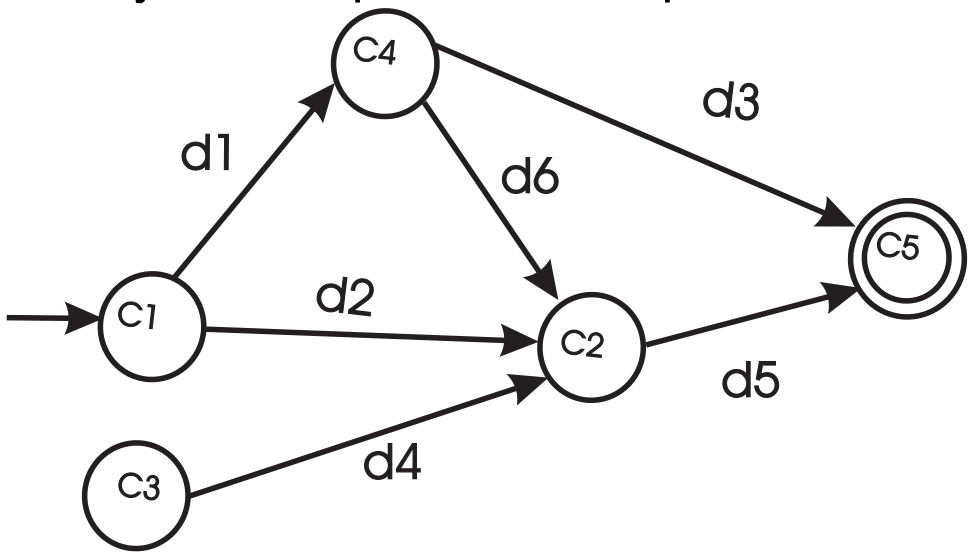
Artificial intelligence-based composition approach (cont.)

Evolutionary Algorithms to analyze Component Composition

Automata representation

$A = (Q, \Sigma, \delta, q_0, F)$, where

- $Q = \{C_1, C_3, C_4, C_2, C_5\}$,
- $\Sigma = \{...\}$, $\delta = \{...\}$,
- $q_0 = \{C_1\}$
- $F = \{C_5\}$



EA representation

$$\underbrace{[1(4,2)]}_{c_1}\underbrace{[1(5)]}_{c_2}\underbrace{[1(2)]}_{c_3}\underbrace{[1(5,2)]}_{c_4}\underbrace{[1()]}_{c_5}$$

Artificial intelligence-based composition approach (cont.)

Multi Expression Programming-based approach

Artificial intelligence-based composition approach (cont.)

Multi Expression Programming-based approach

Representation

Terminals: $T = \{c_1, c_2, ..., c_n\}$;

Functions: $F = \{+, ||\}$.

Artificial intelligence-based composition approach (cont.)

Multi Expression Programming-based approach

Representation

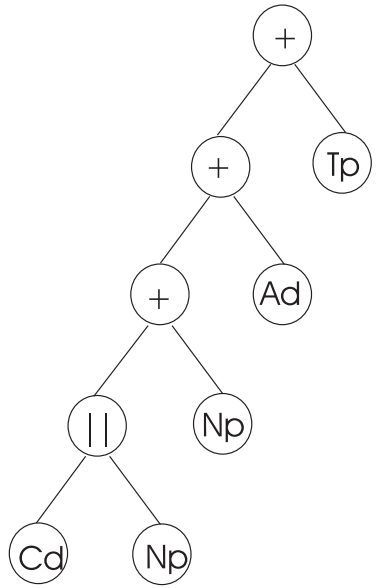
Terminals: $T = \{c_1, c_2, ..., c_n\}$;

Functions: $F = \{+, ||\}$.

Example: Verify if the pair (nextPrime(cd(a, b)), nextPrime(c)) is a pair of twin numbers.

(cd =greatest common divisor).

1: Np
2: Cd
3: || 1, 2
4: Np
5: +3,4
6: Ad
7: +5,6
8: Tp
9: +7,8

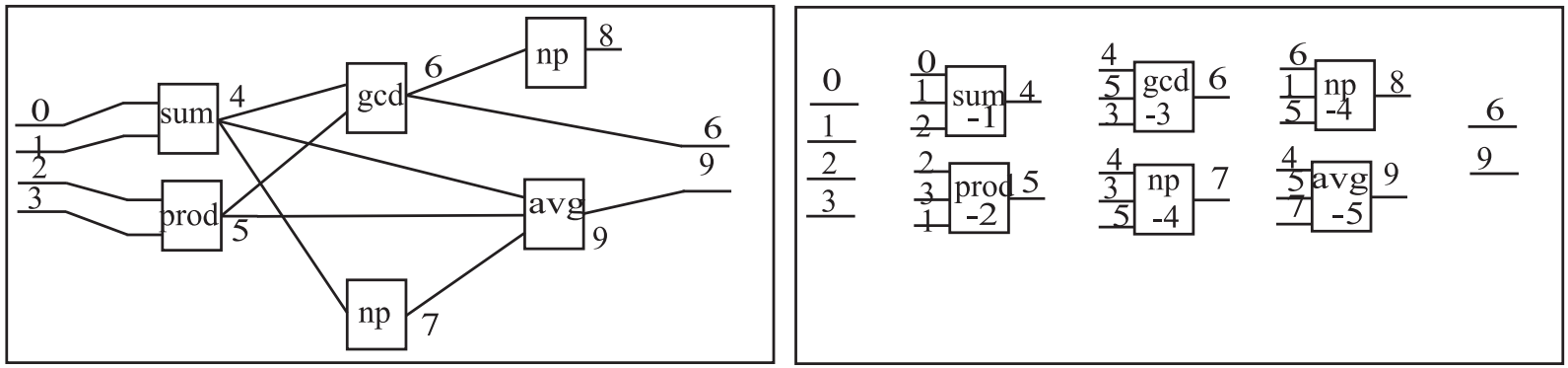


Artificial intelligence-based composition approach (cont.)

Cartesian Genetic Programming-based approach

Artificial intelligence-based composition approach (cont.)

Cartesian Genetic Programming-based approach

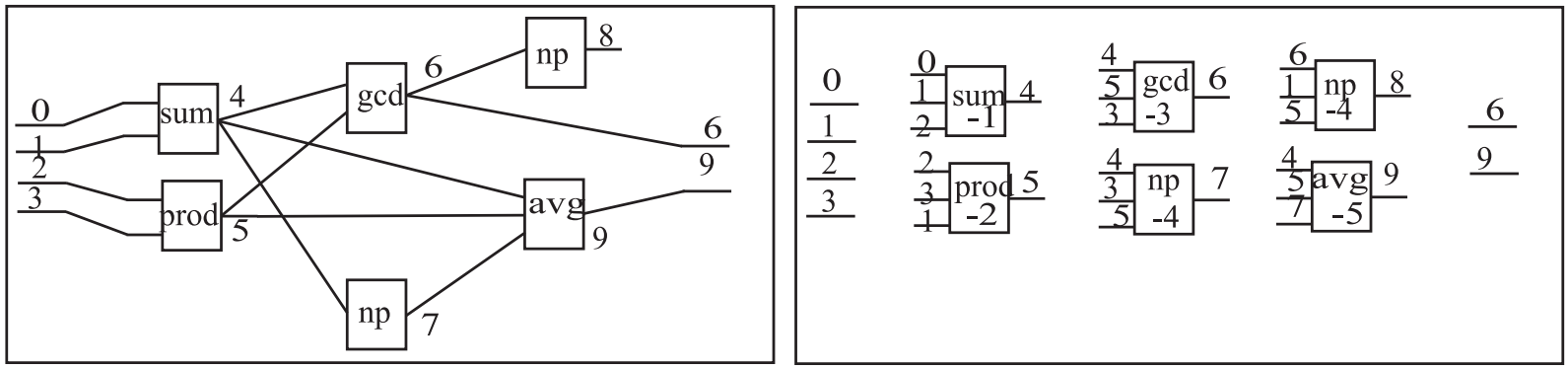


a) Component-based system

b) Cartesian Genetic Program

Artificial intelligence-based composition approach (cont.)

Cartesian Genetic Programming-based approach



a) Component-based system

b) Cartesian Genetic Program

A possible chromosome:

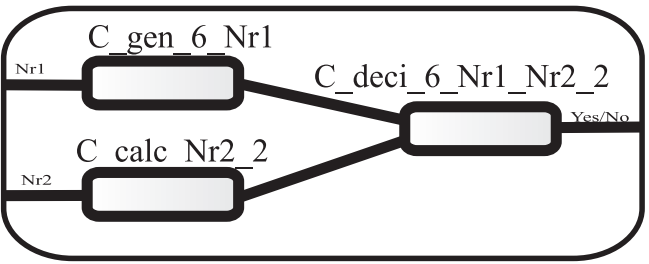
$$\underbrace{0\ 1\ 2\ -1}_{i_4}\ \underbrace{2\ 3\ 1\ -2}_{i_5}\ \underbrace{4\ 5\ 3\ -3}_{i_6}\ \underbrace{4\ 3\ 5\ -4}_{i_7}\ \underbrace{6\ 1\ 5\ -4}_{i_8}\ \underbrace{4\ 5\ 7\ -\ 5}_{i_9}\ \underbrace{6}_{o_0}\ \underbrace{9}_{o_1}$$

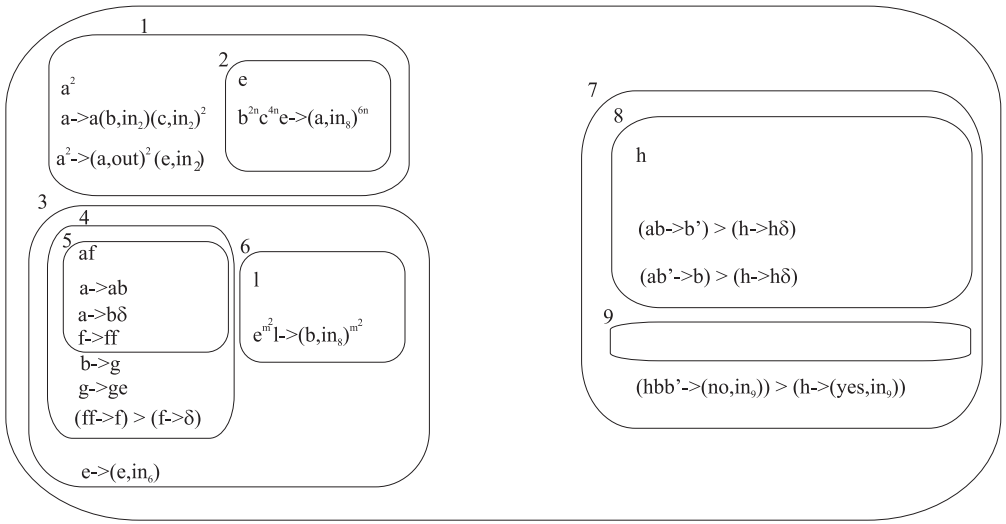
Artificial intelligence-based composition approach (cont.)

P-Systems-based approach

Artificial intelligence-based composition approach (cont.)

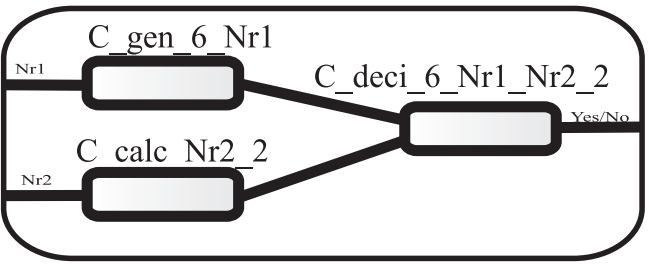
P-Systems-based approach

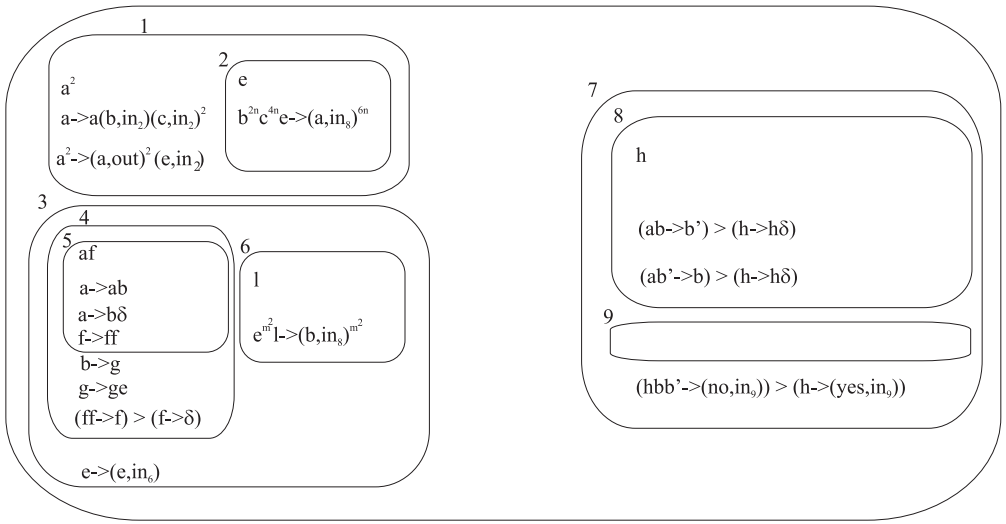




Artificial intelligence-based composition approach (cont.)

P-Systems-based approach





- Example: Decides if the number $6 * Nr_1$ (with Nr_1 given) is divided by the number Nr_2^2 (with Nr_2 given).
- First model execution $Nr_1 = 5$ and $Nr_2 = 4$.
- Second model execution $Nr_1 = 3$ and $Nr_2 = 3$.

Multicriteria-based Component Selection Problem

Multicriteria-based Component Selection Problem approaches

Multicriteria-based Component Selection Problem

Multicriteria-based Component Selection Problem approaches

- Greedy-based composition approach [Vescan, 2008c] (indexed BDI)

Multicriteria-based Component Selection Problem

Multicriteria-based Component Selection Problem approaches

- Greedy-based composition approach [Vescan, 2008c] (indexed BDI)
- Branch and Bound-based composition approach [Vescan and Pop, 2008] (indexed MathSciNet)

Multicriteria-based Component Selection Problem

Multicriteria-based Component Selection Problem approaches

- Greedy-based composition approach [Vescan, 2008c] (indexed BDI)
- Branch and Bound-based composition approach
  [Vescan and Pop, 2008] (indexed MathSciNet)
- Artificial intelligence-based composition approaches
  [Vescan, 2008b] (indexed IEEE), [Vescan, 2008a] (ISI Proceeding),
  [Vescan and Grosan, 2008b] (ISI Proceeding),
  [Vescan and Grosan, 2008a] (indexed IEEE),
  [Vescan et al., 2008] (ISI Proceeding),
  [Vescan, 2008e](indexed ISI-SCI-E)

Multicriteria-based Component Selection Problem

Multicriteria-based Component Selection Problem approaches

- Greedy-based composition approach [Vescan, 2008c] (indexed BDI)
- Branch and Bound-based composition approach
  [Vescan and Pop, 2008] (indexed MathSciNet)
- Artificial intelligence-based composition approaches
  [Vescan, 2008b] (indexed IEEE), [Vescan, 2008a] (ISI Proceeding),
  [Vescan and Grosan, 2008b] (ISI Proceeding),
  [Vescan and Grosan, 2008a] (indexed IEEE),
  [Vescan et al., 2008] (ISI Proceeding),
  [Vescan, 2008e](indexed ISI-SCI-E)

Greedy-based composition approach

Subalgorithm *Greedy(SC, SR, Sol)* **is**:
Begin
Sol := 0; RSR := SR;
While (RSR<>0) **execute**
@Choose c from SC;
Sol := Sol U {c}
RSR := RSR - SR_c
Endwhile
End;

Greedy-based composition approach

Subalgorithm *Greedy(SC, SR, Sol)* **is**:
Begin
Sol := 0; RSR := SR;
While (RSR<>0) **execute**
@Choose c from SC;
Sol := Sol U {c}
RSR := RSR - SR_c
Endwhile
End;

- Ratio Selection Function

$$\frac{SR_c \bigcap RSR}{cost(c)} \ is\ maximal.$$

Greedy-based composition approach

Subalgorithm *Greedy(SC, SR, Sol)* **is**:
Begin
Sol := 0; RSR := SR;
While (RSR<>0) **execute**
@Choose c from SC;
Sol := Sol U {c}
RSR := RSR - SR_c
Endwhile
End;

- Ratio Selection Function

$$\frac{SR_c \bigcap RSR}{cost(c)} \ is\ maximal.$$

- Ratio and Dependencies Selection Function

Greedy-based composition approach

Subalgorithm *Greedy(SC, SR, Sol)* **is**:
Begin
Sol := 0; RSR := SR;
While (RSR<>0) **execute**
@Choose c from SC;
Sol := Sol U {c}
RSR := RSR - SR_c
Endwhile
End;

- Ratio Selection Function

$$\frac{SR_c \bigcap RSR}{cost(c)} \ is\ maximal.$$

- Ratio and Dependencies Selection Function
  - $\frac{SR_c \bigcap RSR}{cost(c)}$ *is maximal*;

Greedy-based composition approach

Subalgorithm *Greedy(SC, SR, Sol)* **is**:
Begin
Sol := 0; RSR := SR;
While (RSR<>0) **execute**
@Choose c from SC;
Sol := Sol U {c}
RSR := RSR - SR_c
Endwhile
End;

- Ratio Selection Function

$$\frac{SR_c \bigcap RSR}{cost(c)} \ is\ maximal.$$

- Ratio and Dependencies Selection Function
  - $\frac{SR_c \bigcap RSR}{cost(c)}$ *is maximal*;
  - the dependencies must be satisfied.

Branch and Bound-based composition approach

Branch and Bound-based composition approach

```
Subalgorithm BranchAndBound(SC, SR, Sol) is:
Begin
  @Select a component from SC and added to the List L;
  While (still exist unexpanded nodes in the list) execute
     @Select an unexpanded node n with f min.;
     @Expand the node and generate a list of successors SUCC;
     For (each succ from SUCC) execute
       @Compute g associate to succ;
        If (succ is not in L) then
          @Add succ in L as unexpand;
        Else
           If ( g(succ) < g value of the found node in L )  then
             @The node found in the list is directed to the actual
                parent of succ with new value g and if the node
                was expand it is marked as unexpand);
          EndIf;
         EndIf;
     EndFor;
  EndWh;
End;
```

Branch and Bound-based composition approach

```
Subalgorithm BranchAndBound(SC, SR, Sol) is:
Begin
  @Select a component from SC and added to the List L;
  While (still exist unexpanded nodes in the list) execute
     @Select an unexpanded node n with f min.;
     @Expand the node and generate a list of successors SUCC;
     For (each succ from SUCC) execute
       @Compute g associate to succ;
        If (succ is not in L) then
          @Add succ in L as unexpand;
        Else
           If ( g(succ) < g value of the found node in L )  then
             @The node found in the list is directed to the actual
                parent of succ with new value g and if the node
                was expand it is marked as unexpand);
          EndIf;
         EndIf;
     EndFor;
  EndWh;
End;
```

Each node n has an associated value:

Branch and Bound-based composition approach

```
Subalgorithm BranchAndBound(SC, SR, Sol) is:
Begin
  @Select a component from SC and added to the List L;
  While (still exist unexpanded nodes in the list) execute
     @Select an unexpanded node n with f min.;
     @Expand the node and generate a list of successors SUCC;
     For (each succ from SUCC) execute
       @Compute g associate to succ;
        If (succ is not in L) then
          @Add succ in L as unexpand;
        Else
           If ( g(succ) < g value of the found node in L )  then
             @The node found in the list is directed to the actual
                parent of succ with new value g and if the node
                was expand it is marked as unexpand);
           EndIf;
         EndIf;
     EndFor;
  EndWh;
End;
```

Each node n has an associated value: $f(n) = g(n) + h(n)$;

Branch and Bound-based composition approach

```
Subalgorithm BranchAndBound(SC, SR, Sol) is:
Begin
  @Select a component from SC and added to the List L;
  While (still exist unexpanded nodes in the list) execute
     @Select an unexpanded node n with f min.;
     @Expand the node and generate a list of successors SUCC;
     For (each succ from SUCC) execute
        @Compute g associate to succ;
        If (succ is not in L) then
           @Add succ in L as unexpand;
        Else
           If ( g(succ) < g value of the found node in L ) then
              @The node found in the list is directed to the actual
                 parent of succ with new value g and if the node
                 was expand it is marked as unexpand);
           EndIf;
        EndIf;
     EndFor;
  EndWh;
End;
```

Each node n has an associated value: $f(n) = g(n) + h(n)$;

- g(n) - the cost of the components that were used until now (from the root node to node n) to construct the solution;

Branch and Bound-based composition approach

```
Subalgorithm BranchAndBound(SC, SR, Sol) is:
Begin
  @Select a component from SC and added to the List L;
  While (still exist unexpanded nodes in the list) execute
     @Select an unexpanded node n with f min.;
     @Expand the node and generate a list of successors SUCC;
     For (each succ from SUCC) execute
       @Compute g associate to succ;
       If (succ is not in L) then
          @Add succ in L as unexpand;
       Else
          If ( g(succ) < g value of the found node in L )  then
            @The node found in the list is directed to the actual
               parent of succ with new value g and if the node
               was expand it is marked as unexpand);
          EndIf;
       EndIf;
     EndFor;
  EndWh;
End;
```

Each node n has an associated value: $f(n) = g(n) + h(n)$;

- g(n) - the cost of the components that were used until now (from the root node to node n) to construct the solution;
- h(n) - the number of remaind requirements that need to be satisfied (to reach the final solution starting from the current node n).

Evolutionary algorithms-based composition approaches

- Requirements-based chromosome representation

Evolutionary algorithms-based composition approaches

- Requirements-based chromosome representation
- Components-based chromosome representation

Evolutionary algorithms-based composition approaches

- Requirements-based chromosome representation
  Multiobjective problem: *fCost* and *fNoComp*, and one constraint (requirements dependencies).

- Components-based chromosome representation
  Multiobjective problem: *fRemReq*, *fCost* and *fNoComp*.

Evolutionary algorithms-based composition approaches

- Requirements-based chromosome representation
  Multiobjective problem: *fCost* and *fNoComp*, and one constraint (requirements dependencies).
  - Multiobjective optimization problem using weighted sum method (*EArWS*).

- Components-based chromosome representation
  Multiobjective problem: *fRemReq*, *fCost* and *fNoComp*.

Evolutionary algorithms-based composition approaches

- Requirements-based chromosome representation
  Multiobjective problem: *fCost* and *fNoComp*, and one constraint (requirements dependencies).
  - Multiobjective optimization problem using weighted sum method (*EArWS*).
  - Multiobjective optimization problem using Pareto dominance principle (*EArP*).
- Components-based chromosome representation
  Multiobjective problem: *fRemReq*, *fCost* and *fNoComp*.
  - Multiobjective optimization problem using Pareto dominance principle (*EAcP*).

Evolutionary algorithms-based composition approaches

- Requirements-based chromosome representation
  Multiobjective problem: *fCost* and *fNoComp*, and one constraint (requirements dependencies).
  - Multiobjective optimization problem using weighted sum method (*EArWS*).
  - Multiobjective optimization problem using Pareto dominance principle (*EArP*).
- Components-based chromosome representation
  Multiobjective problem: *fRemReq*, *fCost* and *fNoComp*.
  - Multiobjective optimization problem using Pareto dominance principle (*EAcP*).

Evolutionary algorithms-based composition approaches (cont.)

Evolutionary algorithms-based composition approaches (cont.)

$SR = \{r_0, r_1, r_2, r_3, r_4, r_5\}$ and $SC = \{c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}$.

c_0	$\{r_0, r_3\}$	8
c_1	$\{r_2, r_5\}$	7
c_2	$\{r_0\}$	6
c_3	$\{r_0\}$	9
c_4	$\{r_1\}$	6
c_5	$\{r_2, r_4\}$	14
c_6	$\{r_3, r_4, r_5\}$	15
c_7	$\{r_4, r_5\}$	14
c_8	$\{r_1, r_2\}$	7
c_9	$\{r_0, r_4, r_5\}$	14

Dep.	r_0	r_1	r_2	r_3	r_4	r_5
r_0		$\surd$				
r_1						
r_2	$\surd$				$\surd$	
r_3		$\surd$				
r_4	$\surd$					
r_5		$\surd$				

Evolutionary algorithms-based composition approaches (cont.)

Approach	r_1	r_0	r_4	r_2	r_3	r_5	*cost*	*no*
GreedyR	8	0	1	8	0	7	36	4
GreedyRd	4	0	9	8	0	9	35	4
BB	8	2	6	8	6	6	28	3
BBd	4	2	6	1	6	6	34	4
EArWS	8	2	6	8	6	6	28	3

Table: GreedyR, GreedyRd, BB, BBd and EArWS Solutions

Evolutionary algorithms-based composition approach (cont.)

- *EArP* and *EAcP*

Evolutionary algorithms-based composition approach (cont.)

- *EArP* and *EAcP*
- Dominance measure

$$DM(A, B) =$$

$$\frac{\text{Dominates (A, B)}}{\text{DifSol(A)} + \text{DifSol(B)}}$$

Evolutionary algorithms-based composition approach (cont.)

- *EArP* and *EAcP*
- Dominance measure

$$DM(A, B) =$$

$$\frac{\text{Dominates (A, B)}}{\text{DifSol(A)} + \text{DifSol(B)}}$$

Evolutionary algorithms-based composition approach (cont.)

- *EArP* and *EAcP*
- Dominance measure

$$DM(A, B) = \frac{\text{Dominates (A, B)}}{\text{DifSol(A)} + \text{DifSol(B)}}$$

Approach	No of nondom.	DM
10 individuals – 10 generations		
EArP	56	0.5
EAcP	34	0
10 individuals – 20 generations		
EArP	83	0.5
EAcP	72	0
10 individuals – 50 generations		
EArP	165	0.5
EAcP	52	0

Partial system construction. Component Adaptation

Partial system construction. Component Adaptation

- Component function adaptation architectures [Vescan, 2008d] (ISI Proceeding)

Component function adaptation architectures

- Adaptation architectures:

Component function adaptation architectures

- Adaptation architectures:
  - Serial (or sequential) adaptation architecture (SAA);

Component function adaptation architectures

- Adaptation architectures:
  - Serial (or sequential) adaptation architecture (SAA);
  - Parallel adaptation architecture (PAA);

Component function adaptation architectures

- Adaptation architectures:
  - Serial (or sequential) adaptation architecture (SAA);
  - Parallel adaptation architecture (PAA);
  - Alternative adaptation architecture (AAA);

Component function adaptation architectures

- Adaptation architectures:
  - Serial (or sequential) adaptation architecture (SAA);
  - Parallel adaptation architecture (PAA);
  - Alternative adaptation architecture (AAA);
  - Repetitive adaptation architecture (RAA).

Component function adaptation architectures

- Adaptation architectures:
  - Serial (or sequential) adaptation architecture (SAA);
  - Parallel adaptation architecture (PAA);
  - Alternative adaptation architecture (AAA);
  - Repetitive adaptation architecture (RAA).
- The behavior adaptation constraints are given for each architecture.

Component function adaptation architectures

- Adaptation architectures:
  - Serial (or sequential) adaptation architecture (SAA);
  - Parallel adaptation architecture (PAA);
  - Alternative adaptation architecture (AAA);
  - Repetitive adaptation architecture (RAA).
- The behavior adaptation constraints are given for each architecture.
- The construction of the new composed component from the two (one for RAA) selected components is described.

Software metrics to quantify quality attributes of CBD
Metrics-based selection of a component assembly

Metrics in Component-Based Software Engineering

Metrics in Component-Based Software Engineering

- Software metrics to quantify quality attributes of CBD
  [Serban and Vescan, 2007a] (indexed BDI)

Metrics in Component-Based Software Engineering

- Software metrics to quantify quality attributes of CBD
  [Serban and Vescan, 2007a] (indexed BDI)
- Metrics-based selection of a component assembly
  [Serban and Vescan, 2007b] (indexed BDI)

Software metrics to quantify quality attributes of CBD

- Assembly
  $DR = (C, D), D \subseteq C \times C.$

Software metrics to quantify quality attributes of CBD

- Assembly
  $DR = (C, D), D \subseteq C \times C.$
- A dependency $d = (c_1, c_2) \in D.$

Software metrics to quantify quality attributes of CBD

- Assembly
  $DR = (C, D), D \subseteq C \times C$.
- A dependency $d = (c_1, c_2) \in D$.

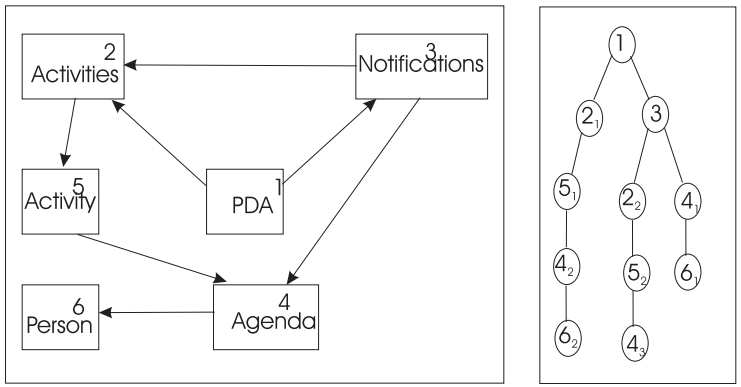


Figure: Personal Digital Assistant components and dependencies and the associated DT

Software metrics to quantify quality attributes of CBD

- Assembly
  $DR = (C, D), D \subseteq C \times C.$
- A dependency $d = (c_1, c_2) \in D$.
- Adapted metrics

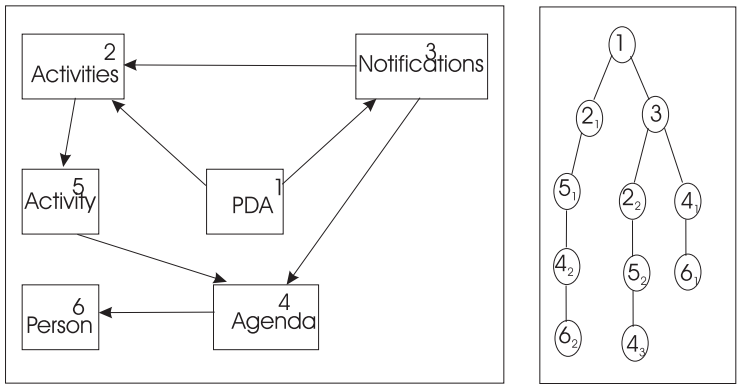


Figure: Personal Digital Assistant components and dependencies and the associated DT

Software metrics to quantify quality attributes of CBD

- Assembly
  $DR = (C, D), D \subseteq C \times C$.
- A dependency $d = (c_1, c_2) \in D$.
- Adapted metrics
  - c_1 **is coupled** with c_2 if $(c_1, c_2) \in D$.

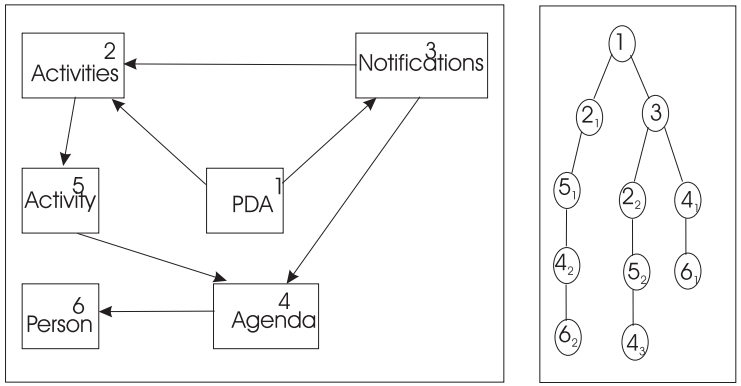


Figure: Personal Digital Assistant components and dependencies and the associated DT

Software metrics to quantify quality attributes of CBD

- Assembly
  $DR = (C, D), D \subseteq C \times C$.
- A dependency $d = (c_1, c_2) \in D$.
- Adapted metrics
  - c_1 **is coupled** with c_2 if $(c_1, c_2) \in D$.
  - Coupling Between Components (CBC(c))

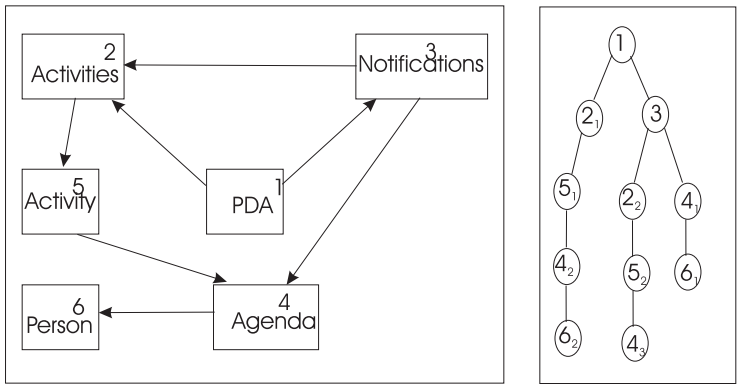


Figure: Personal Digital Assistant components and dependencies and the associated DT

Software metrics to quantify quality attributes of CBD

- Assembly
  $DR = (C, D), D \subseteq C \times C$.
- A dependency $d = (c_1, c_2) \in D$.
- Adapted metrics
  - c_1 **is coupled** with c_2 if $(c_1, c_2) \in D$.
  - Coupling Between Components (CBC(c))
- Defined metrics

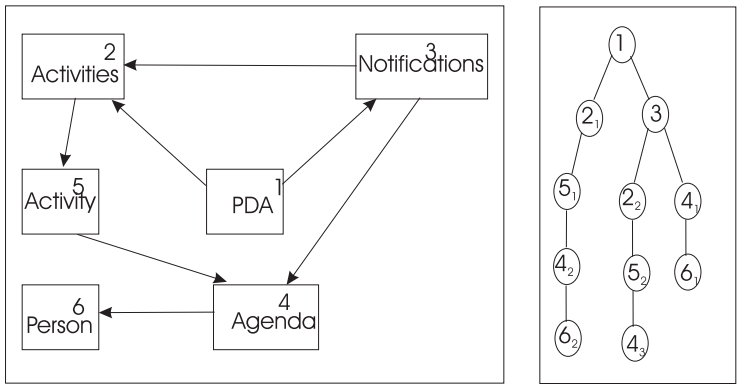


Figure: Personal Digital Assistant components and dependencies and the associated DT

Software metrics to quantify quality attributes of CBD

- Assembly
  $DR = (C, D), D \subseteq C \times C$.
- A dependency $d = (c_1, c_2) \in D$.
- Adapted metrics
  - c_1 **is coupled** with c_2 if $(c_1, c_2) \in D$.
  - Coupling Between Components (CBC(c))
- Defined metrics
  - Depth Dependency Tree (DDT(c));

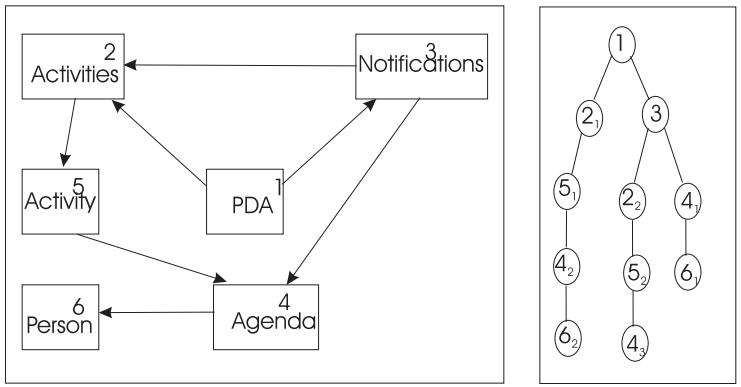


Figure: Personal Digital Assistant components and dependencies and the associated DT

Software metrics to quantify quality attributes of CBD

- Assembly
  $DR = (C, D), D \subseteq C \times C$.
- A dependency $d = (c_1, c_2) \in D$.
- Adapted metrics
  - c_1 **is coupled** with c_2 if $(c_1, c_2) \in D$.
  - Coupling Between Components (CBC(c))
- Defined metrics
  - Depth Dependency Tree (DDT(c));
  - Breadth Dependency Tree (BDT).

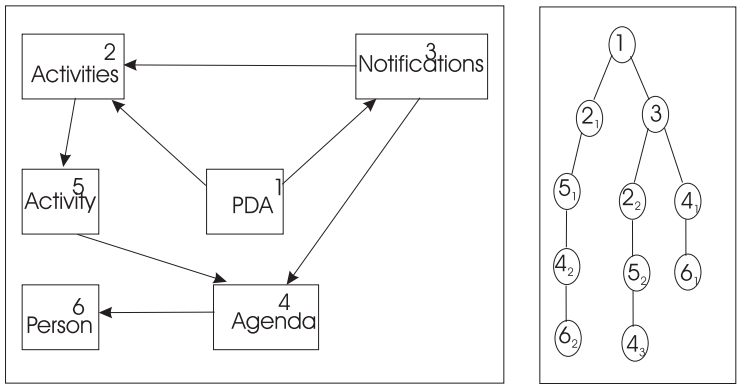


Figure: Personal Digital Assistant components and dependencies and the associated DT

Metrics-based selection of a component assembly

- The influence of metrics values on quality attributes

	Reusability	Functionality	Understandability	Maintainability	Testability
PSU	m/+	m/-	m/+	m/+	m/+
RSU	m/+	-	m/+	m/+	m/+
CPSU	m/+	m/-	m/+	m/+	m/+
CRSU	m/+	-	m/+	m/+	m/+
IDC	m/+	m/-	m/+	m/+	m/+
IIDC	m/+	-	m/+	m/+	m/+
OIDC	m/+	m/-	m/+	m/+	m/+
AIDC	m/+	-	m/+	m/+	m/+
CCG	M/-	M/+	M/-	M/-	M/-
CCTG	M/-	M/+	M/-	M/-	M/-

Metrics-based selection of a component assembly

- The influence of metrics values on quality attributes
- Component Coupling Grade;

	Reusability	Functionality	Understandability	Maintainability	Testability
PSU	m/+	m/-	m/+	m/+	m/+
RSU	m/+	-	m/+	m/+	m/+
CPSU	m/+	m/-	m/+	m/+	m/+
CRSU	m/+	-	m/+	m/+	m/+
IDC	m/+	m/-	m/+	m/+	m/+
IIDC	m/+	-	m/+	m/+	m/+
OIDC	m/+	m/-	m/+	m/+	m/+
AIDC	m/+	-	m/+	m/+	m/+
CCG	M/-	M/+	M/-	M/-	M/-
CCTG	M/-	M/+	M/-	M/-	M/-

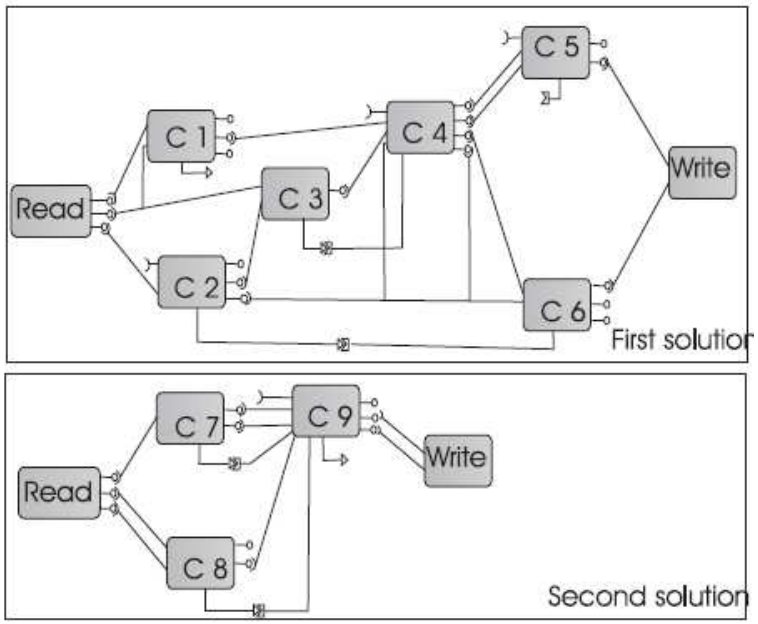
Metrics-based selection of a component assembly

- The influence of metrics values on quality attributes
- Component Coupling Grade;
- Component Coupling Total Grade.

	Reusability	Functionality	Understandability	Maintainability	Testability
PSU	m/+	m/-	m/+	m/+	m/+
RSU	m/+	-	m/+	m/+	m/+
CPSU	m/+	m/-	m/+	m/+	m/+
CRSU	m/+	-	m/+	m/+	m/+
IDC	m/+	m/-	m/+	m/+	m/+
IIDC	m/+	-	m/+	m/+	m/+
OIDC	m/+	m/-	m/+	m/+	m/+
AIDC	m/+	-	m/+	m/+	m/+
CCG	M/-	M/+	M/-	M/-	M/-
CCTG	M/-	M/+	M/-	M/-	M/-

Metrics-based selection of a component assembly

- The influence of metrics values on quality attributes
- Component Coupling Grade;
- Component Coupling Total Grade.



Metrics-based selection of a component assembly

- The influence of metrics values on quality attributes
- Component Coupling Grade;
- Component Coupling Total Grade.

	PSU	RSU	IDC	IIDC	OIDC	CCTG
C_1	0.33	1	0.5	1	0.25	2
C_2	0.66	0.50	0.66	0.5	0.75	1
C_3	1	1	1	1	1	2
C_4	1	0.75	0.88	0.80	1	3
C_5	0.50	0.66	0.50	0.50	0.50	2
C_6	0.33	1	0.71	1	0.33	3
C_R	1	–	1	–	1	–
C_W	–	1	1	1	–	2

CPSU	CRSU	AIDC
0.68	0.82	0.77

First solution

	PSU	RSU	IDC	IIDC	OIDC	CCTG
C_7	1	1	1	1	1	1
C_8	0.50	1	0.80	1	0.66	2
C_9	0.66	0.75	0.70	0.83	0.50	3
C_R	1	–	1	–	1	–
C_W	–	1	1	1	–	2

CPSU	CRSU	AIDC
0.80	0.88	0.90

Second solution

Execution model

Execution model

- Execution model [Fanea, 2005]

Execution model

- Execution model [Fanea, 2005]
- Improved execution rules [Vescan, 2007a] (indexed BDI)

Execution model

- A connection

Execution model

- A connection
- Operations

Execution model

- A connection
- Operations
  - propagation;
  - evaluation.

Execution model

- A connection
- Operations
  - propagation;
  - evaluation.
- State of execution:
  $State =$
  $(operation, componentForEval)$.

Execution model

- A connection
- Operations
  - propagation;
  - evaluation.
- State of execution:
  $State =$
  $(operation, componentForEval)$.
- Execution rules

Execution model

- A connection
- Operations
  - propagation;
  - evaluation.
- State of execution:
  $State = (operation, componentForEval)$.
- Execution rules
  - Alternative rule;
  - Repetitive rule.

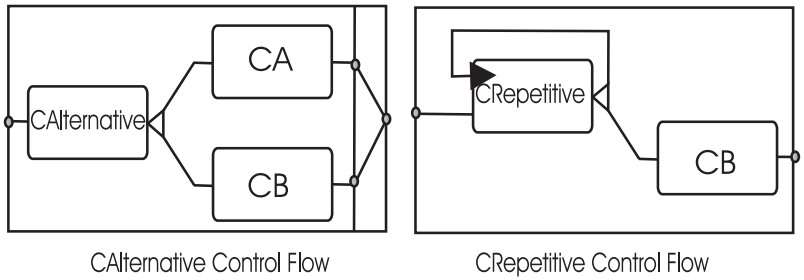


Figure: Alternative and Repetitive rules

Applications

Applications

- Traveling Salesman Problem [Fanea and Pintea, 2005] (indexed BDI), [Vescan and Pintea, 2006a] (indexed IEEE), [Vescan and Pintea, 2006b] (indexed IEEE), [Vescan and Pintea, 2007]

Applications

- Traveling Salesman Problem [Fanea and Pintea, 2005] (indexed BDI), [Vescan and Pintea, 2006a] (indexed IEEE), [Vescan and Pintea, 2006b] (indexed IEEE), [Vescan and Pintea, 2007]
- Labyrinth Problem [Pintea and Vescan, 2007a] (indexed BDI)

Applications

- Traveling Salesman Problem [Fanea and Pintea, 2005] (indexed BDI), [Vescan and Pintea, 2006a] (indexed IEEE), [Vescan and Pintea, 2006b] (indexed IEEE), [Vescan and Pintea, 2007]
- Labyrinth Problem [Pintea and Vescan, 2007a] (indexed BDI)
- Airport Gate Assignment Problem [Pintea and Vescan, 2007b] (indexed BDI)

Traveling Salesman Problem (TSP)

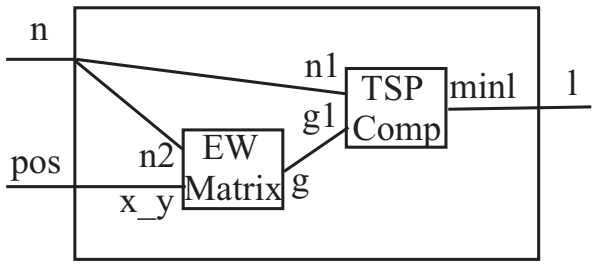


Figure: The inside structure of the TSP component-based system

Traveling Salesman Problem (TSP)

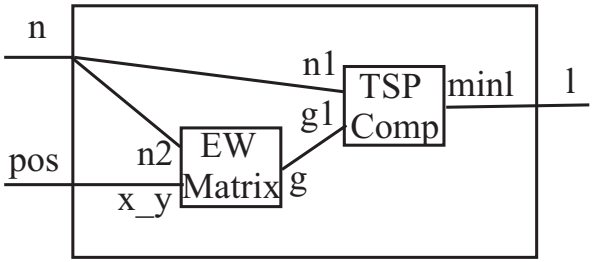


Figure: The inside structure of the TSP component-based system

- Sequential and recursive backtracking approaches

Traveling Salesman Problem (TSP)

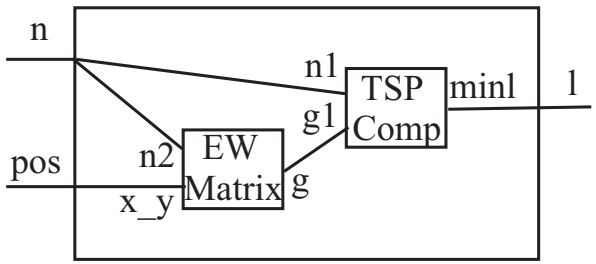


Figure: The inside structure of the TSP component-based system

- Sequential and recursive backtracking approaches
- Component Ant System-based approach

Traveling Salesman Problem (TSP)

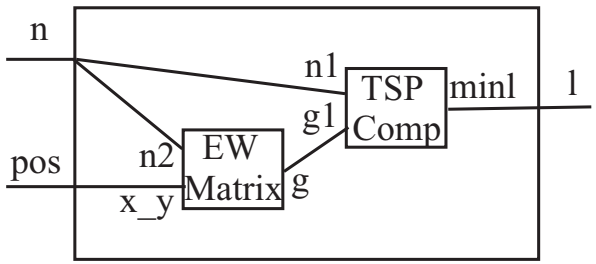


Figure: The inside structure of the TSP component-based system

- Sequential and recursive backtracking approaches
- Component Ant System-based approach
- Component Ant Colony-based approach

Labyrinth Problem (LP)

Component Ant Colony-based approach

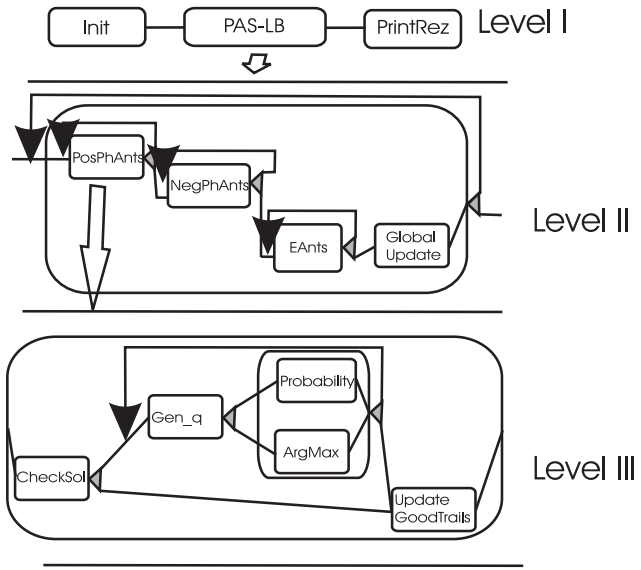


Figure: Architectural levels

Airport Gate Assignment Problem (AGAP)

Component Ant Colony-based approach

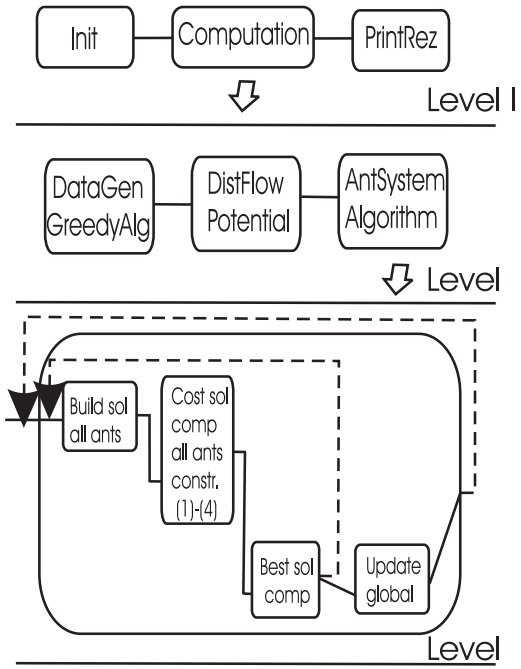


Figure: Architectural levels

Conclusions

Component Selection Problem

Conclusions

Component Selection Problem

- Simple Component Selection Problem
- Multicriteria Component Selection Problem

Conclusions

Component Selection Problem

- Simple Component Selection Problem
  - Backtracking-based composition approaches
  - Automata-based composition approaches
  - Artificial intelligence-based composition approaches
- Multicriteria Component Selection Problem

Conclusions

Component Selection Problem

- Simple Component Selection Problem
  - Backtracking-based composition approaches
  - Automata-based composition approaches
  - Artificial intelligence-based composition approaches
- Multicriteria Component Selection Problem
  - Greedy-based composition approach
  - Branch and Bound-based composition approach
  - Artificial intelligence-based composition approaches

Conclusions

Complet system construction - Component Selection Problem

- Simple Component Selection Problem
  - Backtracking-based composition approaches
  - Automata-based composition approaches
  - Artificial intelligence-based composition approaches
- Multicriteria Component Selection Problem
  - Greedy-based composition approach
  - Branch and Bound-based composition approach
  - Artificial intelligence-based composition approaches

Partial system construction -

Conclusions

Complet system construction - Component Selection Problem

- Simple Component Selection Problem
  - Backtracking-based composition approaches
  - Automata-based composition approaches
  - Artificial intelligence-based composition approaches
- Multicriteria Component Selection Problem
  - Greedy-based composition approach
  - Branch and Bound-based composition approach
  - Artificial intelligence-based composition approaches

Partial system construction - Component adaptation architectures

Conclusions

Complet system construction - Component Selection Problem

- Simple Component Selection Problem
  - Backtracking-based composition approaches
  - Automata-based composition approaches
  - Artificial intelligence-based composition approaches
- Multicriteria Component Selection Problem
  - Greedy-based composition approach
  - Branch and Bound-based composition approach
  - Artificial intelligence-based composition approaches

Partial system construction - Component adaptation architectures

Metrics in Component-Based Software Engineering

Conclusions

Complet system construction - Component Selection Problem

- Simple Component Selection Problem
  - Backtracking-based composition approaches
  - Automata-based composition approaches
  - Artificial intelligence-based composition approaches
- Multicriteria Component Selection Problem
  - Greedy-based composition approach
  - Branch and Bound-based composition approach
  - Artificial intelligence-based composition approaches

Partial system construction - Component adaptation architectures
Metrics in Component-Based Software Engineering
Execution model and Applications - TSP, LP,AGAP

Future work

Further work can be done in the following directions:

- checking if the constructed component configuration supports a given sequence of tasks;
- checking the behavior of components after syntactic composition;
- how can we use AI methods to analyze the behavior of a component-based system or to predict the behavior;
- the use of metrics for the Multicriteria Component Selection Problem.

Questions

- Thank You For Your Attention!


Fanea, A. (2005).
In: Proceedings of the Cluj Computer Science Academic Colloquium pp. 87–92,.


Fanea, A. and Diosan, L. (2005a).
Studia Universitas Babes-Bolyai, Seria Informatica L (2), 23–32.


Fanea, A. and Diosan, L. (2005b).
In: Proceedings of the Cluj Computer Science Academic Colloquium pp. 93–98,.


Fanea, A. and Diosan, L. (2006a).
In: Proceeding of the International Conference on Computers, Communications and Control, ISSN: 1841-9836 (indexed ISI-SCI-E) pp. 474–479,.


Fanea, A. and Diosan, L. (2006b).
In: Proceeding of the International Conference on Computers, Communications and Control, ISSN: 1841-9836 (indexed ISI-SCI-E) pp. 480–485,.


Fanea, A. and Diosan, L. (2006c).
The International Journal of Information Technology and Intelligent Computing, ISSN: 1895-8648 1 (3), 499–508.


Fanea, A. and Motogna, S. (2004).
In: Proceedings of the Cluj Academic Days National Symposium pp. 160–167,.


Fanea, A., Motogna, S., and Diosan, L. (2006).

Studia Universitas Babes-Bolyai, Seria Informatica LI (1), 13–20.


Fanea, A. and Pintea, C. M. (2005).
Annals of Oradea University, Fascicola Matematica XII (0), 91–100.


Fox, M. R., Brogan, D. C., and Paul F. Reynolds, J. (2004).
In: WSC '04: Proceedings of the 36th conference on Winter simulation pp. 429–434, Winter Simulation Conference.


Haghpanah, N., Moaven, S., Habibi, J., Kargar, M., and Yeganeh, S. H. (2007).
In: APSEC pp. 159–166, IEEE Computer Society.


Motogna, S. and Parv, B. (2002).

Bul. Stiint. Univ. Baia Mare, Seria B, Matematica-Informatica XVIII, 269–274.


Parv, B., Motogna, S., and Petrascu, D. (2004).
In: Proceedings of the International Conference on Computers and Communications pp. 325–329,.


Pintea, C.-M. and Vescan, A. (2007a).
In: Proceedings of International Conference on Fundamental Science, ICFS 2007, Applied mathematics and computer science section, ISBN: 978-973-759-367-2 pp. 82–86,.


Pintea, C.-M. and Vescan, A. (2007b).
Studia Universitas Babes-Bolyai, Seria Informatica e LII (1), 21–32.


Serban, C. and Vescan, A. (2007a).

Creative Mathematics and Informatics, ISSN: 1843-441X 16, 143–150.


Serban, C. and Vescan, A. (2007b).
Special Issue of Studia Universitatis Babes-Bolyai Informatica: Proceeding of The International Conference on Knowledge Engineering: Principles and Techniques, ISSN: 1224-869X , 324–331.


Vescan, A. (2006).
In: The 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, ISBN:978-0-7695-2740-6 (ISI Proceeding) pp. 195–200,.


Vescan, A. (2007a).

In: Proceeding of the Second Annual International Conference of Students, Post-graduates and Young Scientists, Computer Science and Engineering - 2007, ISBN: 978-966-553-649-9 pp. 20–24,.


Vescan, A. (2007b).
In: Proceeding of the 3rd Doctoral Workshop on Mathematical and Engineering Methods in Computer Science, ISBN: 978-80-7355-077-6 pp. 249–256,.


Vescan, A. (2008a).
In: Proceedings of the 2nd UKSim European Symposium on Computer Modeling and Simulation, ISBN: 978-0-7695-3325-4 (indexed IEEE) pp. 58–63,.


Vescan, A. (2008b).

In: Proceedings of the First IEEE International Conference on the Applications of Digital Information and Web Technologies, ISBN: 978-1-444-264-9 (indexed IEEE) pp. 252–257,.


Vescan, A. (2008c).
In: Proceedings of the 6th ICAM - International Conference on Applied Mathematics p. (accepted),.


Vescan, A. (2008d).
In: The 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, ISBN: 978-3-540-85566-8 (ISI Proceeding) pp. 319–326,.


Vescan, A. (2008e).
Neural Network world, ISSN: 1210-0552 (indexed ISI-SCI-E) , (accepted).


Vescan, A. and Grosan, C. (2008a).
In: Proceedings of the Fourth International Workshop on Evolutionary Multiobjective Optimization - Design and Applications (indexed IEEE) p. (accepted),.


Vescan, A. and Grosan, C. (2008b).
In: Proceedings of the 3rd International Workshop on Hybrid Artificial Intelligence Systems, ISBN: 978-3-540-87655-7 (ISI Proceeding) pp. 164–171,.


Vescan, A., Grosan, C., and Pop, H. F. (2008).
In: Proceedings of the 2nd International Workshop on Evolutionary Techniques in Data Processing, ISBN: 1529-4188 (indexed IEEE) pp. 509–513,.


Vescan, A. and Motogna, S. (2006a).
In: The 32nd EUROMICRO Software Engineering and Advanced Applications (SEAA), Proceeding of the Work in Progress session, ISBN: 3-902457-11-2 pp. 13–14,.


Vescan, A. and Motogna, S. (2006b).
Pure Mathematics and Applications (PUMA) 17 (3-4), 527–537.


Vescan, A. and Pintea, C.-M. (2006a).
In: Proceeding of The 6th International Conference Communication, ISBN: 978-973-718-479-5 (indexed IEEE) pp. 253–256,.


Vescan, A. and Pintea, C.-M. (2006b).

In: Proceeding of the 2nd International Conference on Intelligence Computer Communication and Processing, ISBN: 973-662-235-5 (indexed IEEE) pp. 23–28,.


Vescan, A. and Pintea, C.-M. (2007).
Journal of Applied Mathematical Science, ISSN: 1312–885X 1 (25–28), 1347–1357.


Vescan, A. and Pop, H. F. (2008).
In: Proceedings of the Work In Progress Session of the 3rd IFIP TC2 Central and East European Conference on Software Engineering Techniques p. (accepted),.