ROLE ACTIVITY DIAGRAMS AS FINITE STATE PROCESSES

Costin Badica
Software Engineering Department
University of Craiova, Romania

HISTORY

- The INSPIRE project IST-10387-1999
- Supporting multiple views of business process models with a single core representation
- Views: RAD/AAD, IDEF0, IDEF3…
- Research on using process algebras for this purpose

BUSINESS PROCESSES

- Business process:
  - a set of partially ordered activities carried out inside an organization, intended to reach a business goal and to produce a result of value for the customer.
- Notations for business processes:
  - High-level visual notations addressed to the business management community.
  - Low-level foundational notations addressed to the computer science community.

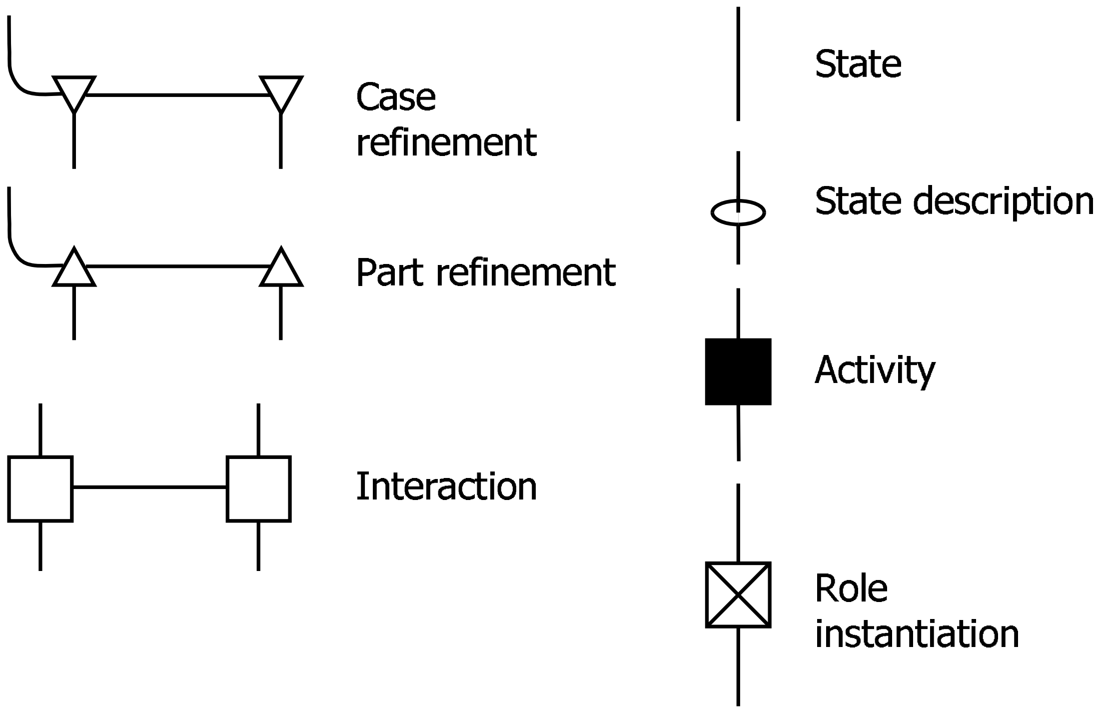
MOTIVATION

- RAD is a popular high-level visual notation for capturing the dynamics and role structure of an organization.
- RAD has been used for modeling, simulation and enactment.
- There are no references in the literature reporting on the formal semantics of RAD.
- We are trying to bridge the gap between low-level and high-level formalisms for business process modeling.

RAD NOTATION (I)

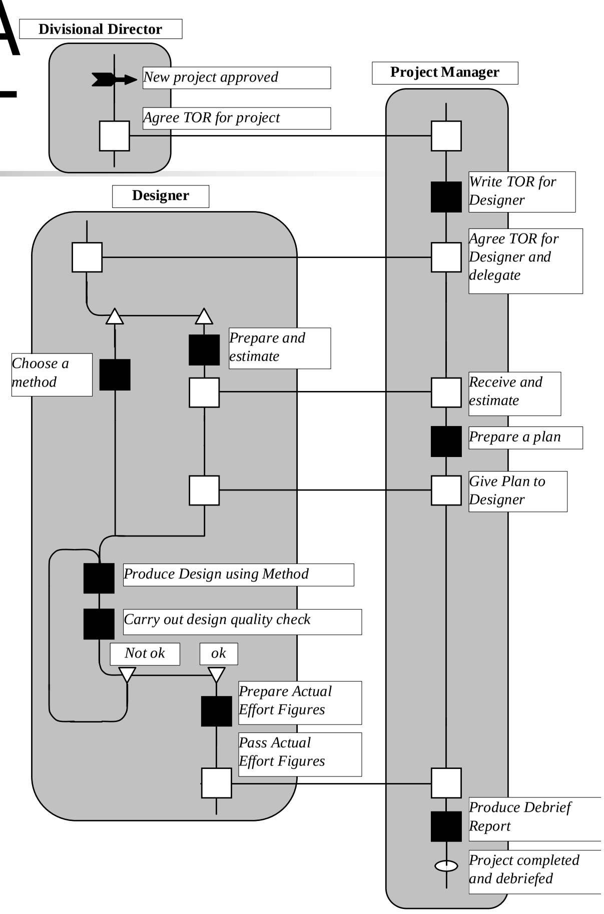
- *Roles* group together activities into units of responsibility. A role has one or more execution threads containing *case refinements* and *part refinements*.
- *Activities* are the basic blocks of a role. They can be carried out in isolation or may require coordination with activities in other roles. In this case they are called *interactions*.
- *External events* are points at which state changes occurring in the process environment influence on our process.

RAD NOTATION (II)



CARRYING OUT A DESIGN PROJECT

- Three roles:
  - Divisional Director
  - Project Manager
  - Designer



FSP PROCESS ALGEBRA

- FSP is an algebraic specification technique of finite state labeled transition systems.
- The tool Labeled Transition System Analyzer – LTSA that allows:
  - Writing FSP specifications
  - Generating the underlying LTS
  - Verification of safety and progress properties

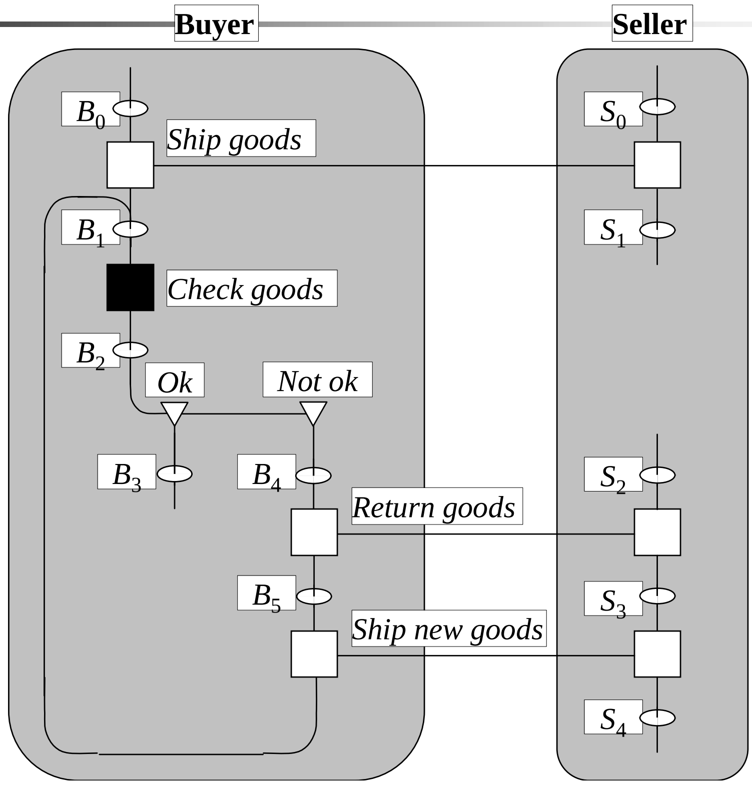
FSP OPERATIONS

- Notation: a is an action, P and Q are process terms, H is a set of actions and L is a re-labeling function
- Operations:
  - Prefix: $a \rightarrow P$
  - Choice: $P \mid Q$
  - Parallel composition: $P \mid\mid Q$
  - Hiding: $P \setminus H$
  - Re-labeling: P / L
  - Definition: $A = P_A$

FSP PROCESSES

- Primitive (sequential) processes
  - Is defined using a set of local definitions in which the process term is either END or choice.
- Composite processes
  - Is defined with a definition in which the process term is a parallel composition, a hiding or a re-labeling.
- Maintaining this distinction assures that the resulting models are finite.

INTERACTION BETWEEN A BUYER AND A SELLER



MODELLING RAD STATES

- State lines:

$$NL = L0,$$
$$L0 = (i \rightarrow L1|end \rightarrow END),$$
$$L1 = (s \rightarrow o \rightarrow L0).$$
$$||L = NL \setminus \{s\}.$$

- End states:

$$E = (i \rightarrow E|end \rightarrow END).$$

- Single instance start states:

$$SS = (o \rightarrow end \rightarrow END).$$

- Multiple instance start states

$$SM = (o \rightarrow SM|end \rightarrow END).$$

MODELLING THE BUYER

$$\| B_0 = SS/\{sg/o\}.$$
$$\| B_1 = L/\{\{sg, sng\}/i, cg/o\}.$$
$$\| B_2 = L/\{cg/i, \{n\_ok, ok\}/o\}.$$
$$\| B_3 = E/\{ok/i\}.$$
$$\| B_4 = L/\{n\_ok/i, rg/o\}.$$
$$\| B_5 = L/\{rg/i, sng/o\}.$$
$$\| Buyer = (B_0 \parallel B_1 \parallel B_2 \parallel B_3 \parallel B_4 \parallel B_5).$$

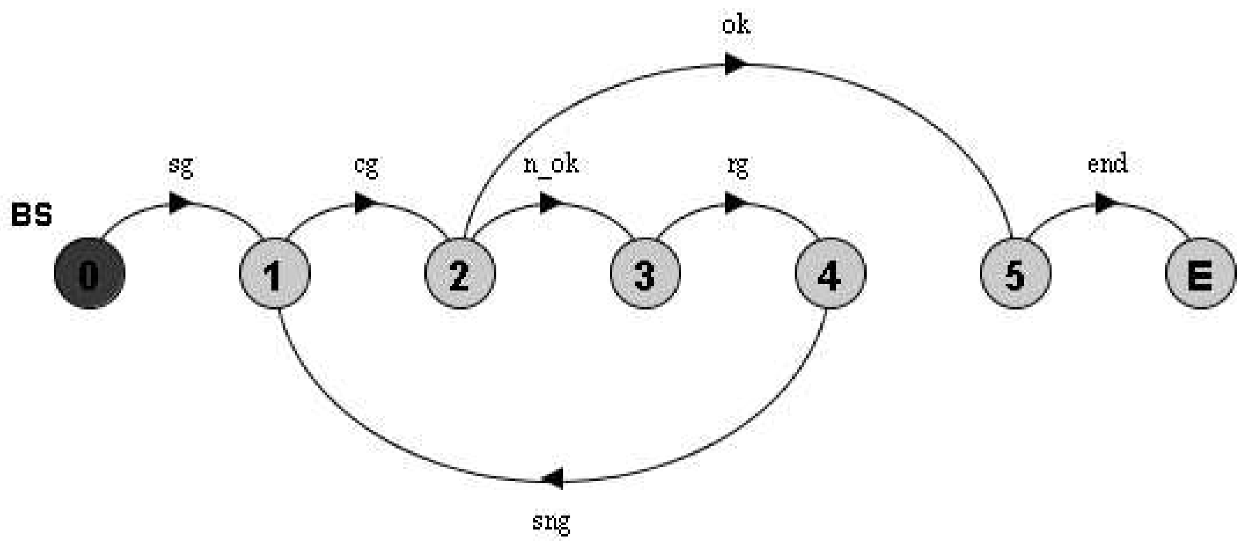
MODELLING THE SELLER

$$\| S_0 = SS/\{sg/o\}.$$
$$\| S_1 = E/\{sg/i\}.$$
$$\| S_2 = SM/\{rg/o\}.$$
$$\| S_3 = L/\{rg/i, sng/o\}.$$
$$\| S_4 = E/\{sng/i\}.$$
$$\| Seller = (S_0 \parallel S_1 \parallel S_2 \parallel S_3 \parallel S_4).$$

MODELLING THE INTERACTION BUYER-SELLER

$$\|System = (Buyer \parallel Seller).$$

LTS OF THE BUYER-SELLER EXAMPLE

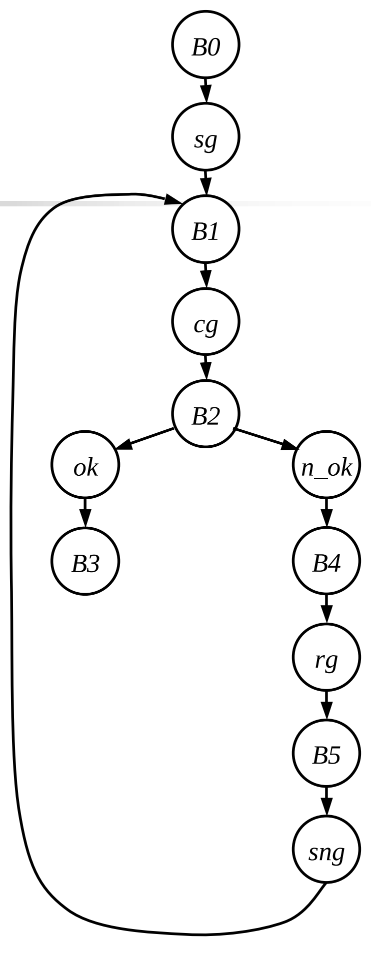


RAD MODELS AS GRAPHS

- *RAD model* = a set of role models
- *Role model* = a bipartite directed graph with nodes partitioned into states and actions.
  - States = states in a role
  - Actions = activities, external events, case refinements and part refinements
  - A case refinement with *n* alternatives ⇒ *n* actions
  - A part refinement ⇒ at least one action node. If two or more threads originating from the same part refinement recombine together ⇒ a new action is added.

EXAMPLE

- The graph for the buyer in the buyer-seller interaction example
- Uppercase nodes are *states* and lowercase nodes are *actions*.



CLASSIFYING STATES

- It is needed for the mapping algorithm
- States are partitioned into:
  - Labeled state lines
  - Unlabeled state lines
- Start states = states with no in-coming edges
- End states = states with no out-coming edges

MAPPING STATES

- An unlabeled state line s $\Rightarrow$ $P(s) = L/\{E^{-1}(s)/i, E(s)/o\}$
- A labeled state line s with label l $\Rightarrow$ $P(s) = NL/\{E^{-1}(s)/i, l/s, E(s)/o\}$
- A single instance start state s $\Rightarrow$ $P(s) = SS/\{E(s)/o\}$
- A multiple instance start state s $\Rightarrow$ $P(s) = SM/\{E(s)/o\}$
- An end state $s \Rightarrow$ $P(s) = E/\{E^{-1}(s)/i\}$

algorithm $RAD2FSP($
  input a RAD model $\mathcal{R}$
  output a FSP model Δ for $\mathcal{R})$
  $\Delta \leftarrow$ definitions of processes L, NL, E, SS, SM
  $PN \leftarrow$ a new composite process name
  $F_1 \leftarrow PN +' = ('$
  for each $R \in \mathcal{R}$ **do**
    $RN \leftarrow$ a new composite process name
    $F_2 \leftarrow RN +' = ('$
    for each $s \in S(R)$ **do**
      $(SN, F_3) \leftarrow S2FSP(s)$
      $\Delta \leftarrow \Delta \cup \{F_3\}$
      $F_2 \leftarrow F_2 \cup \{SN\}$
      if s is not the last in $S(R)$ **then** $F_2 \leftarrow F_2 +' \|'$
    $F_2 \leftarrow F_2 +').'$
    $\Delta \leftarrow \Delta \cup \{F_2\}$
    $F_1 \leftarrow F_1 + RN$
    if R is not the last in $\mathcal{R}$ **then** $F_1 \leftarrow F_1 +' \|'$
  $F_1 \leftarrow F_1 +').'$
  $\Delta \leftarrow \Delta \cup \{F_1\}$

POSSIBLE USES OF THE FSP MODEL

- Interactive simulation:
  - Trace into a process by executing it step by step.
- Static verification:
  - Check if the model has some desired correctness properties. This can be done using model-checking. LTSA has the possibility to check *safety* and *progress* properties.
- Dynamic simulation:
  - Estimate performance indicators of a business process: degree of utilization of resources, cycle times, waiting times, throughput, a.o.

RELATED WORK

- Mapping RAD to DES (Martinez-Garcia, Warboys, 1998)
- Modeling PIF using CCS (Schroeder, 1999)
- Modeling workflow processes using FSP (Karamanolis, Giannakapoulou, Magee, Wheater, 1999)
- Modeling UML AD with FSP (Rodrigues, 2000)
- Modeling UML AD with CSP (Bolton, Davies, 2000)
- Modeling RAD using PEPA (Badica et Fox 2001)

CONCLUSIONS

- RAD notation has a natural mapping to FSP.
- The mapping has been presented as a translation algorithm.
- The resulting FSP model can be used to derive a LTS computational model of a business process.
- The computational model is useful for checking the behavior of the business process before having actually to implement it in practice.