Business Processes Modelling

MPB (6 cfu, 295AA)

Roberto Bruni
http://www.di.unipi.it/~bruni

19 - Workflow modules
Object

We study Workflow modules to model interaction between workflows

Ch.6 of Business Process Management: Concepts, Languages, Architectures
Problem

Not all tasks of a workflow net are automatic:

they can be triggered manually or by a message

they can be used to trigger other tasks

How do we represent this?
Implicit interaction

Separately developed processes

Some activities can input messages (symbol `?`)

Some activities can output messages (symbol `!`)

Diagram:

- Seller (receives a suggestion, sends a decision)
- `{?rec_reject, ?rec_accept}`
- `{!reject, !accept}`
Interface

Seller has an interface for interaction

It consists of some input places and some output places
From Workflow nets to Workflow modules

Assume the original workflow net has been validated:

it is a sound (and maybe safe) workflow net

When we add the (places in the) interface
it is no longer a workflow net!
It becomes a workflow module
Workflow Modules

Definition: A workflow module consists of a (sound) workflow net \((P, T, F)\) plus a set \(P^I\) of incoming places plus a set of incoming arcs \(F^I \subseteq (P^I \times T)\) plus a set \(P^O\) of outgoing places plus a set of outgoing arcs \(F^O \subseteq (T \times P^O)\) such that each transition in \(T\) has at most one arc to places in the interface \(P^I \cup P^O\)
Workflow module: example
Structural compatibility

A set of workflow modules is called **structurally compatible** if

for every message that can be sent there is exactly a module who can receive it,

and

for every message that can be received there is exactly a module who can send it

(formats of message data are assumed to match)
Compatibility

Auctioning Service

Seller

sending!

receiving?

sending!

receiving?
Interaction

Auctioning Service

Seller

!rec_accept  !rec_reject

?accept  ?reject

!reject  !accept

ra  ra  rr  rr  sa  sa  sr  sr
Problem

We have added places and arcs to single wf nets
We have joined places of different wf modules

How do we check that the system behaves well?

What has this check to do with WF net soundness?
Workflow systems
Workflow system
Workflow system

Definition: A workflow system is a wf net that consists of

a set of \( n \) structurally compatible wf modules
(initial places \( i_1, \ldots, i_n \), final places \( o_1, \ldots, o_n \))

plus an initial place \( i \)
and a transition \( t_i \) from \( i \) to \( i_1, \ldots, i_n \)

plus a final place \( o \)
and a transition \( t_o \) from \( o_1, \ldots, o_n \) to \( o \)

whose initial marking is \( i \)
Soundness of workflow systems

A workflow system is just an ordinary workflow net.

We can check its soundness as usual.
Example

Is the wf system sound?
Example

not sound!
Example

Is the wf system sound?
Example sound!
Exercise

Compose the two wf modules and check soundness
Exercise

Compose the two wf modules and check soundness
Weak soundness
Problem

When checking behavioural compatibility the soundness of the overall net is a too restrictive requirement

Workflow modules are designed separately, possibly reused in several systems
It is unlikely that every functionality they offer is involved in each system
Problem

**Definition:** A workflow net is **weak sound** if it satisfies “option to complete” and “proper completion”

(dead tasks are allowed)

Weak soundness can be checked on the RG

It guarantees deadlock freedom and proper termination of all modules
Sound + Sound = ?
Sound + Sound = not sound
Sound + Sound = not sound
Sound + Sound = not sound
Sound + Sound = not sound
Sound + Sound = not sound
Sound + Sound = not sound
Sound + Sound = not sound
Sound + Sound = not sound
Sound + Sound = not sound

Dead tasks!
Sound + Sound = not sound

Weak Sound!