University of Aarhus
in Computer Science
Basic Research
BRICS

Joint work with Michael I. Schwartzbach
Anders Sandholm

FASE, March 31, 1998

Web Services
Distributed Safety Controllers for Web Services
Problem: Need concurrency control.

```java
{
    { 0
        Reset { counter } =
        { }
        show ThePage[num \rightarrow counter]
        counter = counter + 1;
        session Readme { }
        <;
        } You are visitor number "var name=\"num\"
        } document ThePage
dl
        } = 0;
    service
}

A Simple Web Service
Suffices for simple services — what about complex services?

```hs
{
  session Reset counter:=0
  {
    show ThePage[num -> counter]
      {
        counter:=counter+1
        take counter
      }
    session Readme
      {
        you are visitor number `var name="num"
      }
    document ThePage
    global counter: int = 0
  }
  service
}
```

Low-Level Semaphore Programming
Distributed Safety Controllers for Web Services, FASE’98, Anders Sandholm.

Hard to use and maintain.
Lack of expressiveness.

Disadvantages:

Well-known concepts.

Advantages:

(e.g., monitors and semaphores).
Language constructs for concurrency control

The Standard Solution
Alternative Idea

- Runtime system without performance loss.
- Synthesize controller.
- Allow high-level specification.
General Setup
Plan

- Conclusion
- Implementation
- Improvements on the Runtime System
- Runtime System
- Labeled Services and Safety Requirements
{ session Reset counter:0 }

{ show ThePage [num → counter]
  give-counter:
  counter=counter+1;
  take-counter:
} readme}{

You are visitor number "var name=num".
}

document ThePage
global counter: int = 0;
}

Labeled Services
At most one thread at a time.

\[ \text{Atime} \left( t, t' \right) \Leftrightarrow \left( t > t' \land \text{end-A}(t) \right) \land \left( t > t' \land \text{start-A}(t') \right) \]

Safety Requirements — Example
Mon: Safety Requirements → DFA

\[ \phi \] (How?)

where \( \text{pre}(T) \) is the prefix closure of \( T \).

\[ \text{pre}(T) = \{ 0 < u | \omega \in 1^{-u} \rho \cdots 0 \rho \} \]

Allow only runs \( \rho = 0 \rho 1 \rho 2 \cdots \) for which

Synthesizing the Controller
Example — Revised

\[ \text{Time } t': (t' \land \text{start-A}(t') \land \text{end-A}(t')) \]

Mona

\[ \text{Time } t': t' < t' \land \text{end-A}(t') \]
Plan

Conclusion

Implementation

Improvements on the Runtime System

Runtime System

Labeled Services and Safety Requirements
Safety Requirements

Combining Labeled Services and

\[ \text{We want } T(S) \cup \{A'\} \cdot \]

- a controller \( A^c \)
- a service \( S \) and

Given

Distibuted Safety Controllers for Web Services, FASE'98, Anders Sandholm.
Runtime System
Plan

- Implementation
- Improvements on the Runtime System
- Runtime System
- Labeled Services and Safety Requirements
- Conclusion
Size of state space: \(2^n + 1\)

\[
\text{Atime } t, t' \geq t'' \implies \text{give-} X^{(t)} \Rightarrow

\text{Atime } t, t' \geq t'' \Rightarrow \text{take-} X^{(t)} \Rightarrow

\text{Atime } t, t' \geq t'' \Rightarrow \text{take-} X^{(t)} \Rightarrow

\text{Atime } t, t' \geq t'' \Rightarrow \text{take-} X^{(t)} \Rightarrow

\text{Safety requirements}

\text{Globals } X^1, \ldots, X^n

\text{The State Explosion Problem}
\[ \frac{\mu^b = \nu^b}{\exists \alpha \not\in \sigma} \]

\text{and} \quad \frac{\mu^b \equiv \nu^b}{\exists \alpha \in \sigma} \quad \text{if and only if}

where

\[ (\forall x_1, \ldots, x_n) \left( \mu^b \leftarrow (\nu^b, \ldots, \nu^b) \right) \leftarrow (\mu^b \leftarrow (\nu^b, \ldots, \nu^b)) \times \cdots \times \emptyset \) \]

\text{Behavior of} \quad \forall \quad \text{•}

\[ (\forall x_1, \ldots, x_n) \leftarrow (\mu^b \leftarrow (\nu^b, \ldots, \nu^b)) = \emptyset \]

\text{Distributed automatlon} \quad \text{•}

\[ (\exists x_1 \in \cdots \exists x_n) \left( \mu^b \leftarrow (\nu^b, \ldots, \nu^b) \right) = \emptyset \left( \exists x_1 \cdots \exists x_n \right) \]

\text{Distributed alphabet} \quad \text{•}

\text{Distributed Automaton}
Example — Revised
Transitions.

Observation: It is OK to remove locally dead labels and

A \( o \)-transition is dead if \( o \) is dead.

\( o \)-transitions are state preserving.

\( h \leftarrow o \) is state preserving if \( h = \).

Inference of Independence Information
Connected components can be run completely in parallel.

**Observation:**

\[
\{ \emptyset \neq f \cup \exists \; \forall' \forall \} = \mathcal{E}
\]

and edges

\[
\{ \forall' \forall' \cdot \cdot \cdot \forall' \forall \} = \Lambda
\]

Consider \( \mathcal{E}, \Lambda \) with nodes

**Dividing the Central Controller**
Example — Re-revisited
Plan

Implementation

- Improvements on the Runtime System
- Runtime System
- Labelled Services and Safety Requirements

Conclusion
(See http://www.brix.dk/~mbs/wf/index.htm).

- Enhanced input forms, etc.
- Support for work-flow
- Concept of user state
- Relational database
- Typed dynamic HTML documents
- WIG (Web Interface Generator),

Central part of the domain specific language

Implementation
Conclusion

Implementation

Improvements on the Runtime System

Runtime System

Labeled Services and Safety Requirements

Plan
No performance loss

Synthesize controller automatically

High-level specification of concurrency constraints

Therefore runtime controller is ideal

Slow networks

Fast machines

Rapidly changing code

Web programming environment

Conclusion
\[
\{ (S) \mid session \} \cap \{ (S) \mid A \ \text{end} \} \cap \{ (S) \mid A \ \text{start} \} \\
\cap \{ (S) \mid X \ \text{give} \} \cap \{ (S) \mid X \ \text{take} \} \\
\cap (S) \text{labels} = \sum \\
\text{We get} \\
\text{session } A: \text{start-A and end-A} \\
\text{Global } X: \text{take-X and give-X} \\
\text{Standard Alphabet}
\]
Service $s = s_H$ and label, and where transitions

$$\text{Service } s = s_H$$

acceptance states.

$$\text{auto}$$
\{ \ \phi \models m \ | \ \exists \ \mathcal{G} \in m \ \} = (\phi)_{T}

- Language of \( \phi \) if \( \phi \) is true over \( m \).
- \( \cdots \), \( (t) \in \mathcal{T} \), \( t \in \mathcal{T} \cap \mathcal{T}' \) + \( t \).
- Boolean connectives: \( \land \), \( \lor \), \( \lnot \), \( \Leftarrow \), \( \rightarrow \).
- \( \exists \), \( \forall \) over positions and position sets.

\[ \text{M2L-SfT} \]

Monadic Second-Order Logic on Strings
\[(\mathcal{V}\times \mathcal{V}) I \cup (\mathcal{I}\times \mathcal{I}) I = (\mathcal{V} \times \mathcal{V}) I\]  

Observe that \( I \) is the identity relation on \( \mathcal{V} \times \mathcal{V} \).

Where

\[(\mathcal{I}\times \mathcal{I}) I \]

is the identity relation on \( \mathcal{I}\times \mathcal{I} \).

We define the product

\[(\mathcal{V}\times \mathcal{V}) I \cup (\mathcal{I}\times \mathcal{I}) I = (\mathcal{V} \times \mathcal{V}) I\]

Given \( \mathcal{V}\times \mathcal{V} \) and \( \mathcal{I}\times \mathcal{I} \), we have

Product Automaton
• Wait for permission to continue.
• Push thread id onto reg(0).

Passing Label 0:

Compiling Labels
Runtime System now behaves as $A S \times A^e$.

- Wake up thread corresponding to $id$.
- Make $o$-transition, and
- Remove thread $id$ from $Rt(q(o))$.

If $Rt(q(o))$ is nonempty and $o$ is enabled then

The Safety Controller