

# Process algebras as a usable/used tool for the construction of correct systems

Jan Friso Groote



**TU** / **e**

Technische Universiteit  
**Eindhoven**  
University of Technology

**Where innovation starts**

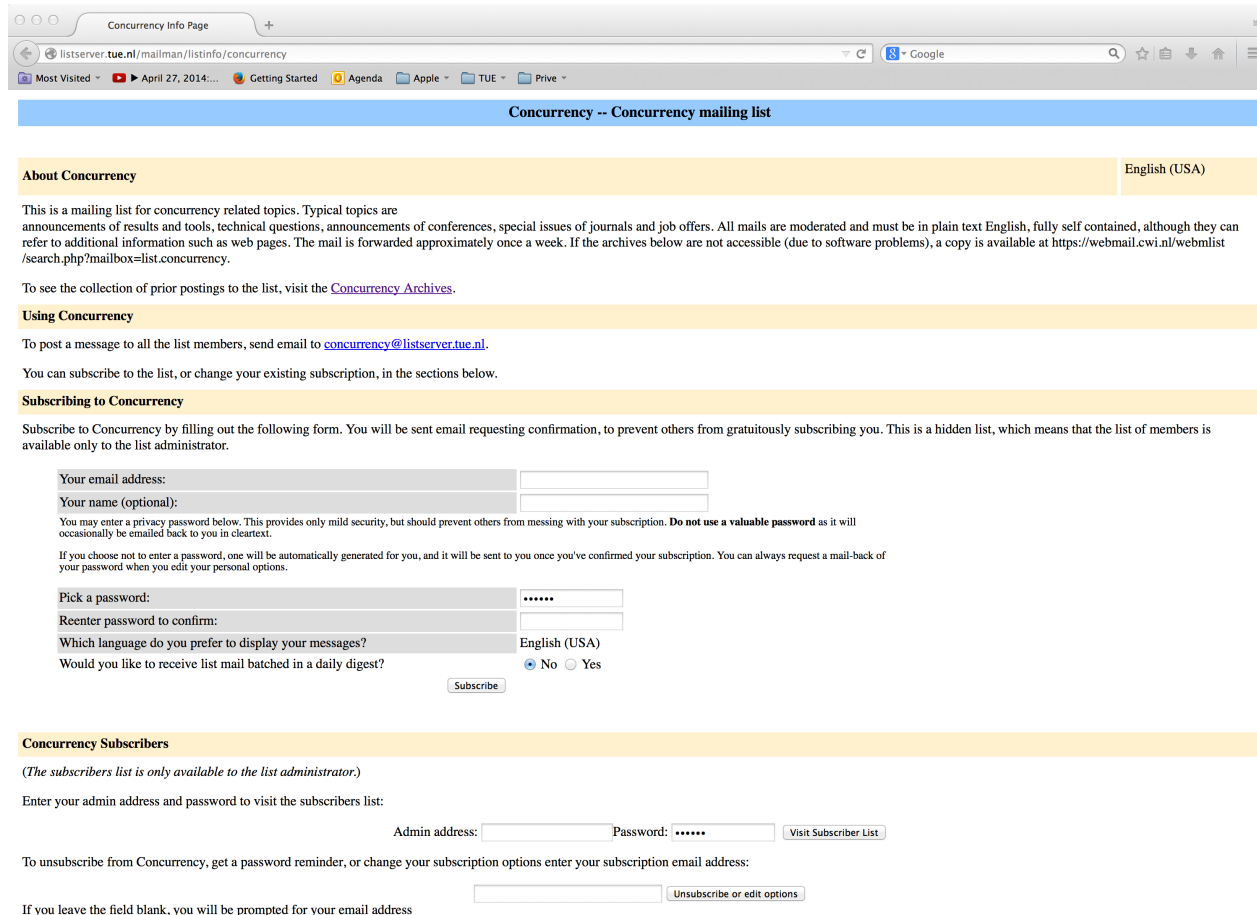
# Concurrency mailing list.

1171 members (slowly growing)

I am maintaining  
it since 1996.

Frits Vaandrager

Albert Meyer



Concurrency Info Page

listserv.tue.nl/mailman/listinfo/concurrency

Most Visited April 27, 2014... Getting Started Agenda Apple TUE Privé

## Concurrency -- Concurrency mailing list

English (USA)

### About Concurrency

This is a mailing list for concurrency related topics. Typical topics are announcements of results and tools, technical questions, announcements of conferences, special issues of journals and job offers. All mails are moderated and must be in plain text English, fully self contained, although they can refer to additional information such as web pages. The mail is forwarded approximately once a week. If the archives below are not accessible (due to software problems), a copy is available at <https://webmail.cwi.nl/webmlist/search.php?mailbox=list.concurrency>.

To see the collection of prior postings to the list, visit the [Concurrency Archives](#).

### Using Concurrency

To post a message to all the list members, send email to [concurrency@listserv.tue.nl](mailto:concurrency@listserv.tue.nl).

You can subscribe to the list, or change your existing subscription, in the sections below.

### Subscribing to Concurrency

Subscribe to Concurrency by filling out the following form. You will be sent email requesting confirmation, to prevent others from gratuitously subscribing you. This is a hidden list, which means that the list of members is available only to the list administrator.

Your email address:

Your name (optional):

You may enter a privacy password below. This provides only mild security, but should prevent others from messing with your subscription. **Do not use a valuable password** as it will occasionally be emailed back to you in cleartext.

If you choose not to enter a password, one will be automatically generated for you, and it will be sent to you once you've confirmed your subscription. You can always request a mail-back of your password when you edit your personal options.

Pick a password:

Reenter password to confirm:

Which language do you prefer to display your messages? English (USA)

Would you like to receive list mail batched in a daily digest?  No  Yes

### Concurrency Subscribers

*(The subscribers list is only available to the list administrator.)*

Enter your admin address and password to visit the subscribers list:

Admin address:  Password:

To unsubscribe from Concurrency, get a password reminder, or change your subscription options enter your subscription email address:

If you leave the field blank, you will be prompted for your email address

# PA used/usable for construction of systems?

“Usable”: no open question. The answer is definitely yes!

“Used”: ??

Hardly by us.

Hardly by colleagues.

Hardly by industry.

Why? Most of us are not interested in the application of concurrency.

We and our colleagues are not really aware of what can be done.

We do not use all possibilities to teach students, esp. non computer science students.

There is little industrial acceptance, also due to a perceived lack of stable support (startups) and doubt about usability.

# Verify the source code of the Algrabrug



Algrabrug:

Formulate requirements on good behaviour.  
Verify these on the source code.

Detected issues solved.

Concerns about software architecture.



Auteur  
Maikel Leemans  
m.leemans@student.tue.nl  
Ruud Koolen  
r.p.j.koolen@student.tue.nl  
Sjoerd Cranen  
scranen@gmail.com  
Sander Jurgens  
s.e.jurgens@student.tue.nl  
Jan Friso Groote  
j.f.groote@tue.nl

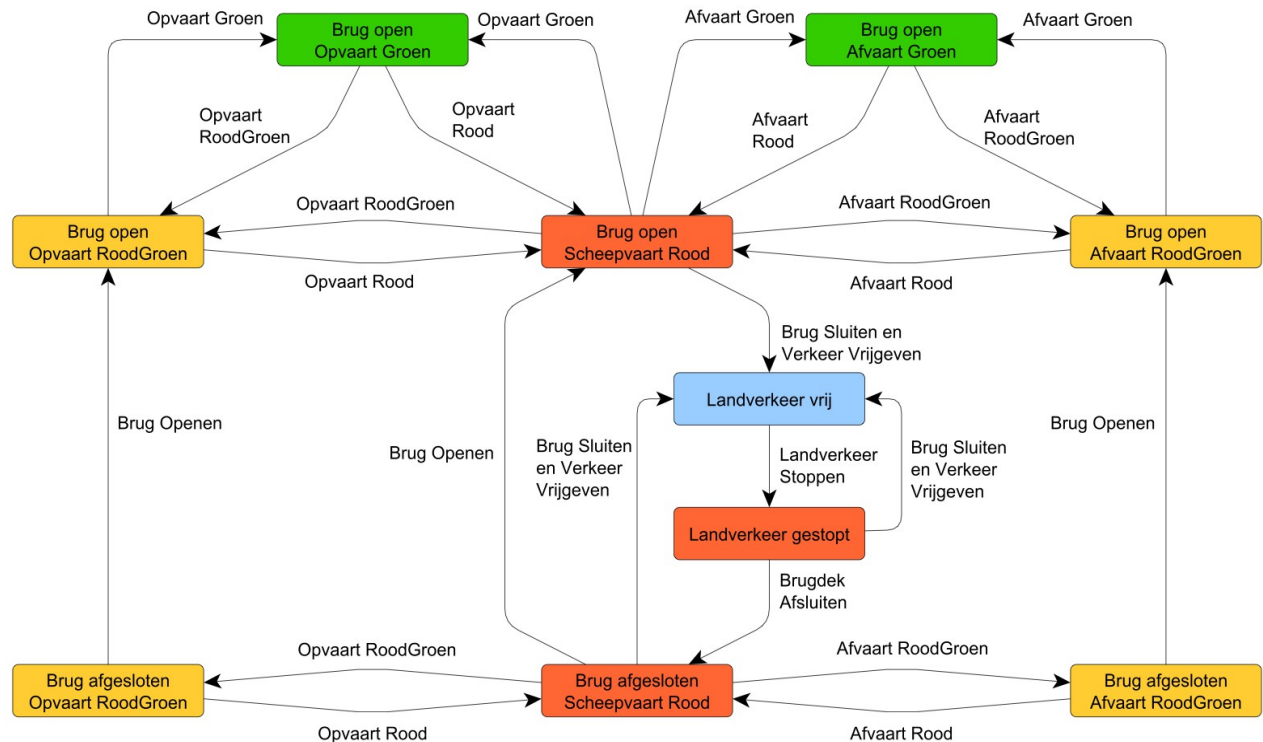
Opdrachtgever  
Rijkswaterstaat

Datum  
29 april 2014

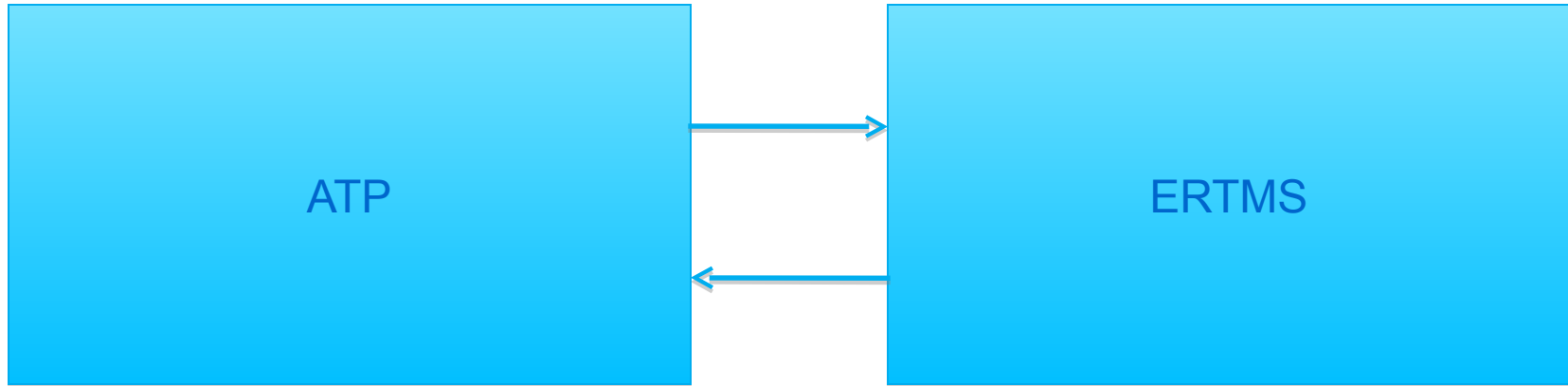
Versie  
1.3

# Analyse van besturingssystemen voor beweegbare bruggen

Systeemvalidatie Rapport in opdracht van Rijkswaterstaat



# Formal modelling of ATP in context of ERTMS



Model the future Automatic Train Protection System that must cooperate with the ERTMS /ETCS system.



# 10 times quality improvement/Design for verification.

At X-ray (Philips Healthcare) software is developed by staff using formal methods.

Is this better? Yes, up to **10** times less bugs, up to **3** times faster.

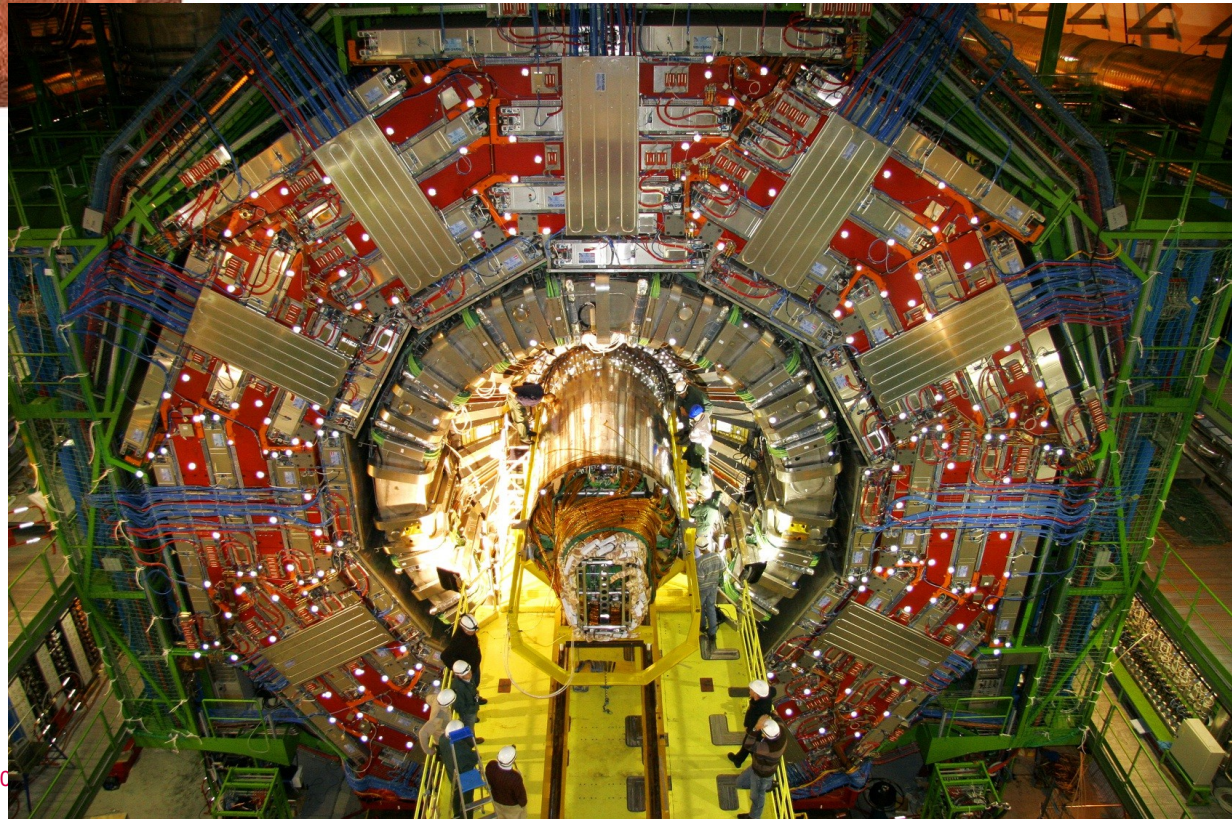
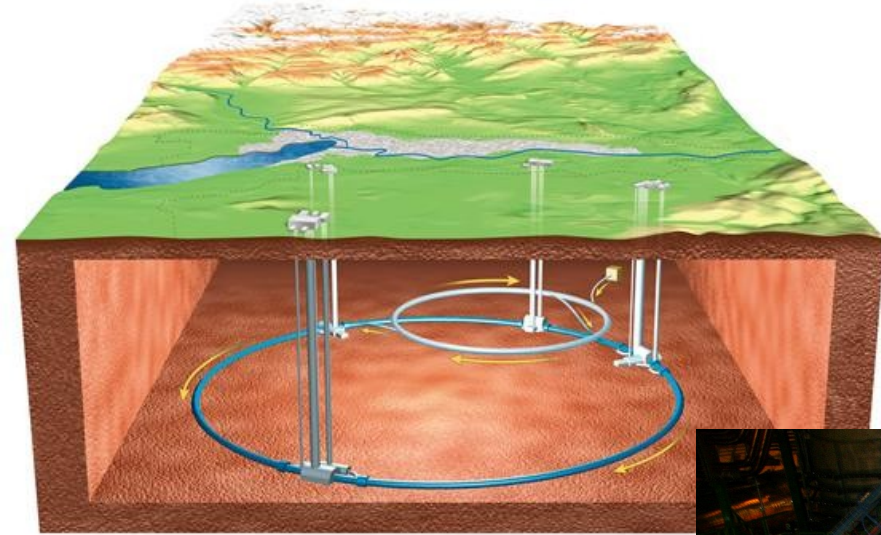
Industry standard **5-50** bugs/Kloc. Formal techniques **0.7** bugs/Kloc.

Required: design for verification.



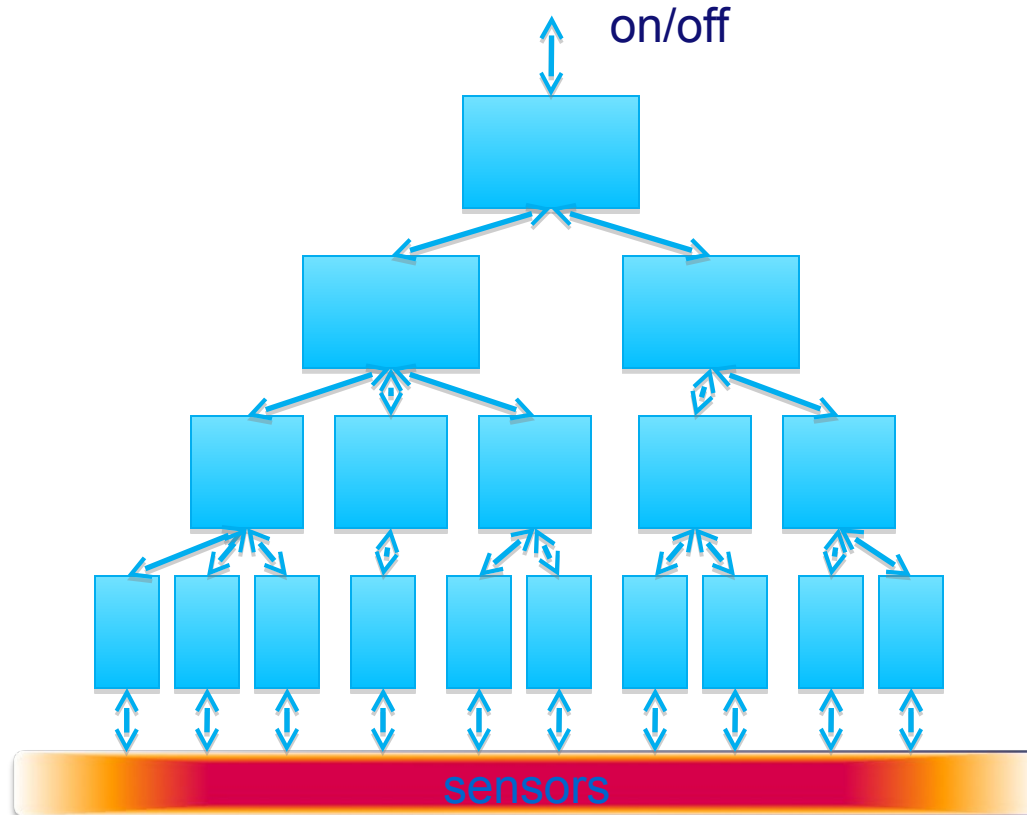
# CERN in Geneve: CMS Detector

Purpose: detection of the Higgs particle.






# Software control by finite state machines



60.000 software modules (Atlas)  
180 different types



Formal verification tools are now standard in the development of control software at CERN.

# PA used/usable for construction of systems?

“Usable”: no open question. The answer is definitely yes!

“Used”: ??

Hardly by us.

Hardly by colleagues.

Hardly by industry.

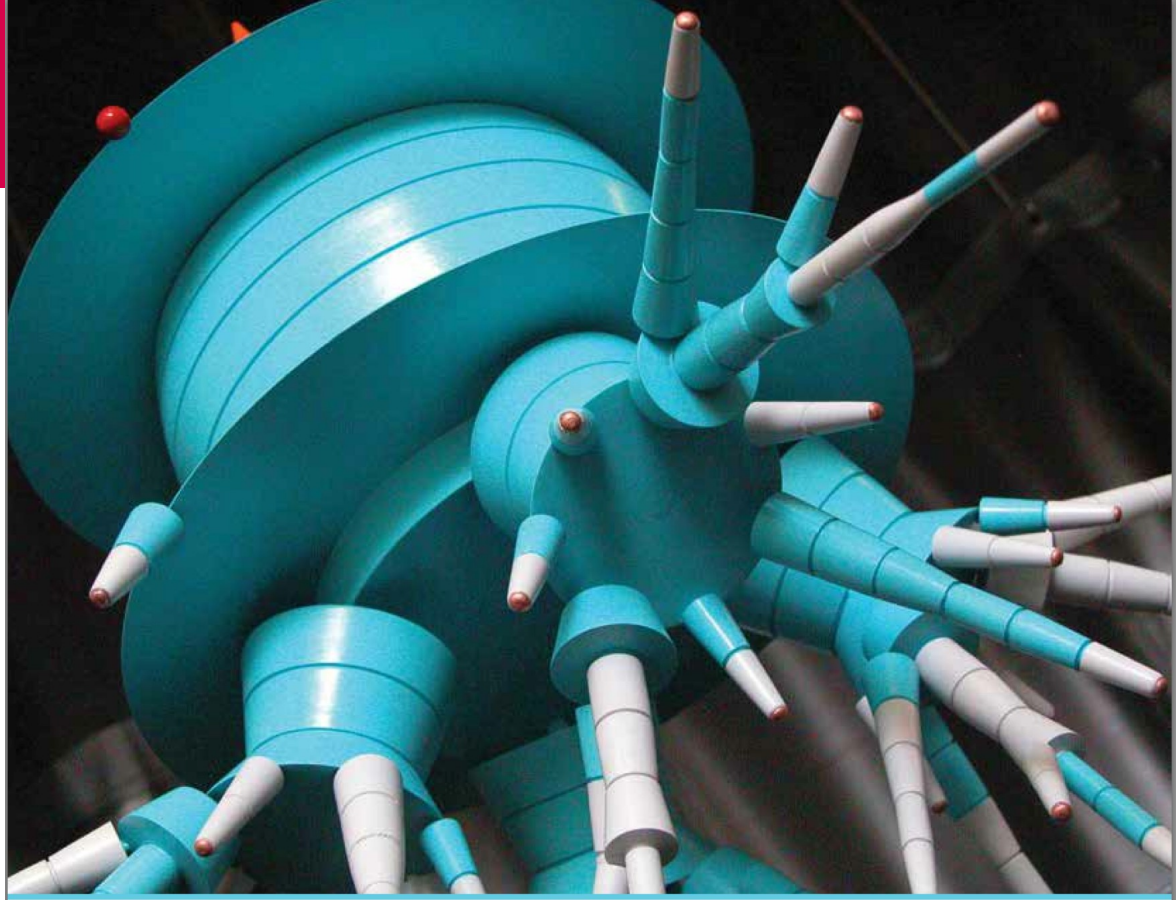
Why? Most of us are not interested in the application of concurrency.

We and our colleagues are not really aware of what can be done.

We do not use all possibilities to teach students, esp. non computer science students.

There is little industrial acceptance, also due to a perceived lack of stable support (startups) and doubt about usability.

MIT Press.  
Available for preorder.  
Will appear on August  
22, 2014.



## **MODELING AND ANALYSIS OF COMMUNICATING SYSTEMS**

Jan Friso Groote and Mohammad Reza Mousavi

# Why would we eat our own dogfood??

Muhammad Atif. Formal Modeling and Verification of Distributed Failure Detectors. PhD. Thesis. Eindhoven University of Technology. 2011.

26 distributed algorithms 23 not correct.

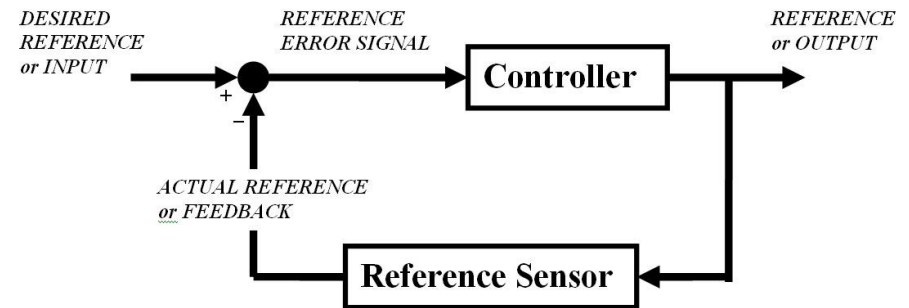
Positive example: Wan Fokkink. Free University, Amsterdam.

Model checking is highly effective. Deadly tool when doing refereeing. Far more efficient than computer assisted/controlled theorem proving, and of course the (my) human brain.

# Do we teach students Concurrency Theory and practice in such a way that they appreciate it?

Mechanical and electrical engineering:

Teach control theory as an essential design tool.



How many of us teach concurrency theory as an essential software/algorithm design and verification tool? How many of us have actually designed systems/algorithms in this way?

# We underestimate how hard system design is...

In case one might try to design a new system/algorithm only a small amount of the time is in formal modelling and analysis.

Most goes into understanding the problem and finding an appropriate solution.

It requires true interest in the application domain, quite some experience using the tools and a full set of tools to operate efficient in this domain.

How many of us do know somebody that would qualify as a concurrency theory based system engineer?

# Why is industry not embarking

There are really great tools available.

FDR3

CADP

mCRL2

Spin

Uppaal

nSMV

Prism

Employees lack understanding and appreciation for concurrency.

Quite often they are not even capable of abstractly understanding their own software, and they have many other concerns.

Number of contacts between industry and effective sellers of concurrency theory is relatively small.

Industry does not like academic tools.

There are a number of start-ups, but they all have a hard time.



# PA used/usable for construction of systems?

“Usable”: no open question. The answer is definitely yes!

“Used”: ??

Hardly by us.

Hardly by colleagues.

Hardly by industry.

Why? Most of us are not interested in the application of concurrency.

We and our colleagues are not really aware of what can be done.

We do not use all possibilities to teach students, esp. non computer science students.

There is little industrial acceptance, also due to a perceived lack of stable support (startups) and doubt about usability.

# Open problem:

How to transform the gems from concurrency theory into a common commodity in computer science/system engineering?

# Proposal (à la Joachim):



Assemble a group of researchers and model check all distributed algorithms appearing in:

DISC

ICDCN

ICDCS

ICPADS

OPODIS

PODC

PPoPP

SPAA

