

General and Specific Issues in Teaching Computer Science/ TLA+

Friedrich Vogt/Paul Tavolato

University of Technology Hamburg-Harburg and
University of Applied Science, St. Pölten, Austria

The paper

Integrating Formal Methods into Computer Science Curricula at a UAS (St. Pölten, Austria)

describes the universe of discourse.

My presentation concentrates on two parts:

- Present challenges on education and
- My experience in teaching Model Checking using **TLA+** (since 2009) at UAS

Student(s), TLA+ and real problems?

“I missed the reference to the practice: for example, in which projects or which companies do really use TLA+ and what for? Maybe I’ll thinking about the features or the possibilities of TLA+ in some years, when I’m in some projects and I’m missing something and TLA+ could be the solution for that. But at the moment I am really not able to answer that question.”

... tells us, to distinguish between two kinds of knowledge (Marotzki , U of Magdeburg):

- **available** (or ready to use!) knowledge
and
- **orientation** (direction of progress)
knowledge

But how can we assure ...

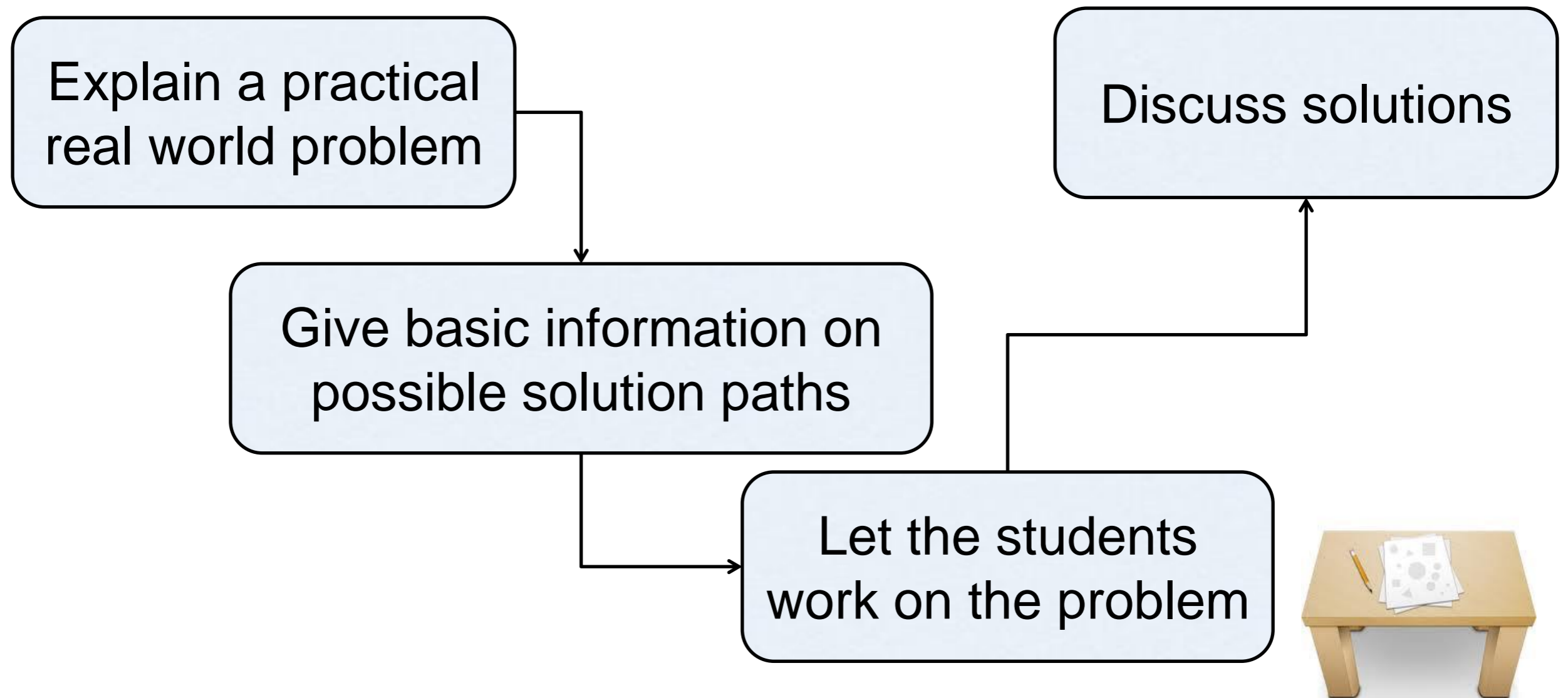
... that available knowledge does not come as „passive“ knowledge (A. N. Whitehead)

by

- Learning from projects in industry – right from the beginning (this idea is currently successfully implemented at UAS) and
- Learning from scientists/engineers working in industry (e.g. Amazon, Motorola, etc.)?

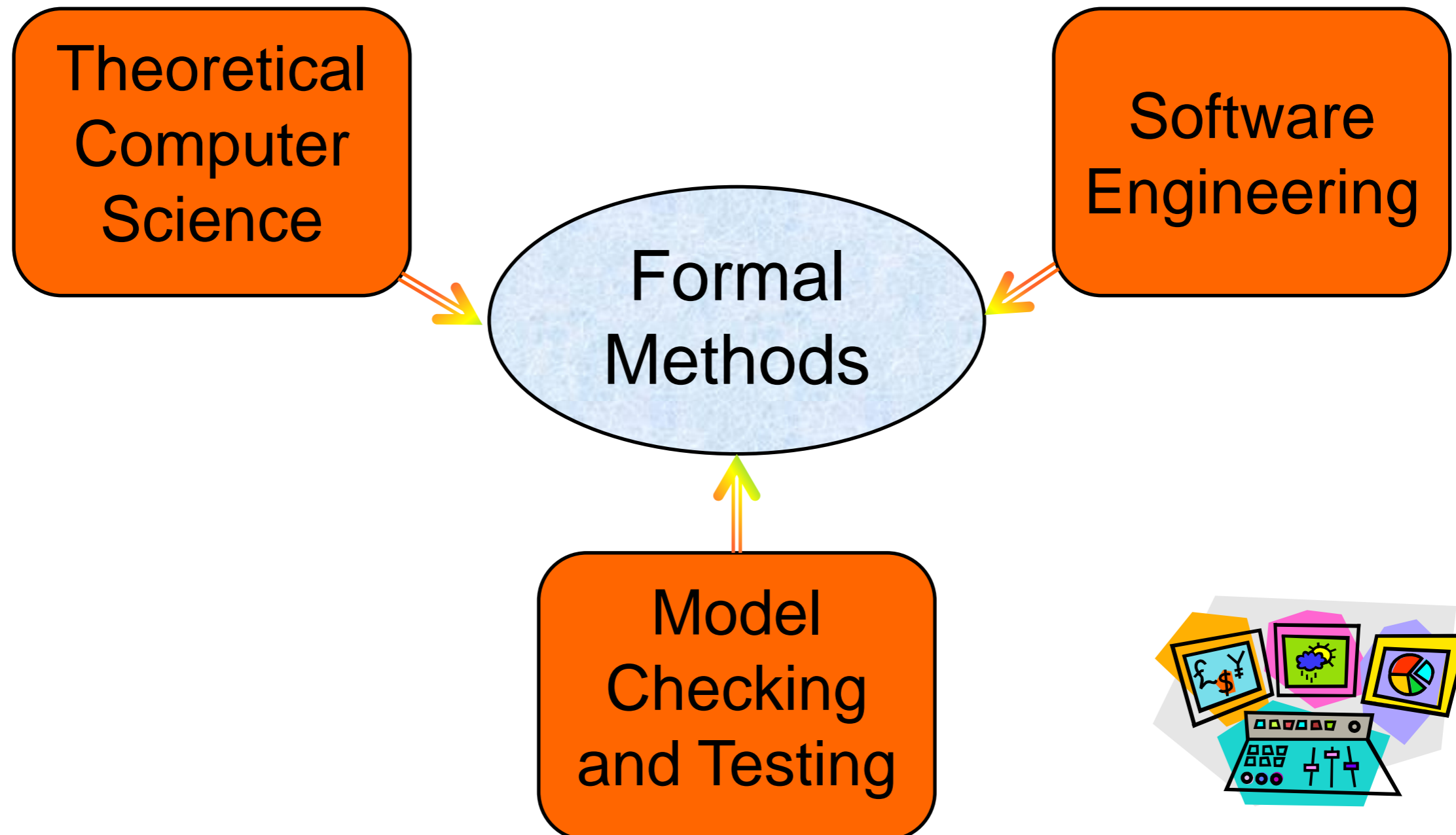
At UAS ...

a problem oriented teaching approach is used:
(teaching is organized blockwise)



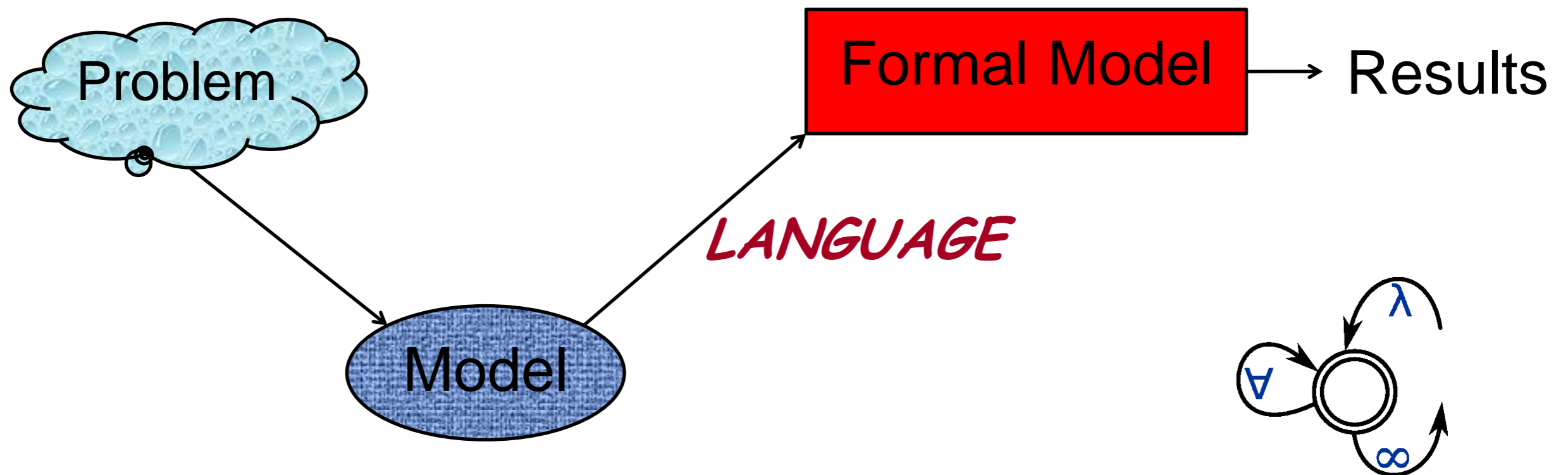
And Implemented for CS by ...

the following three courses:



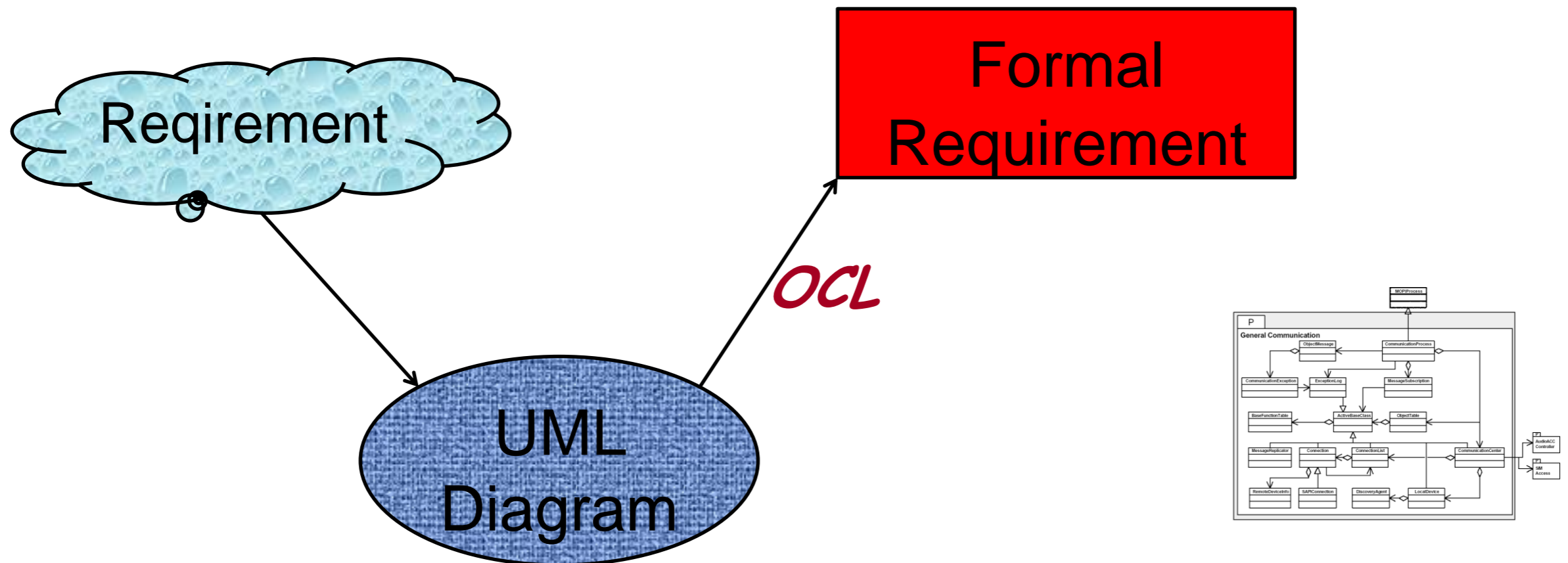
Theoretical Computer Science,

- Model building in general
- Complexity of algorithms
- Formal language theory and automata
- Introduction to logics



Software Engineering, and

- Requirements engineering
- Software design
- UML
- Software project management



Model Checking, being Aware that:

“The task of science is not to explain nature through an appeal to a conceptual representation of it, but rather to explain a conceptual representation through an appeal to experience”
and

“abstraction is foremost a tool of science”.

(as of A. N. Whitehead)

Trying to answer Questions like ...

- What has to be provided before learning can take place ?
- How can we avoid the impression students may have, that teaching theory is mostly related to obtain „passive“ knowledge?
- Is it possible to install learning prerequisites for those who want to be part of a learning community?

Not Ignoring the Current State in CS Education, which to my opinion ...

... is to a large extent characterized by the fact that:

- The so called „available knowledge“ in CS is quite low and seems to become lower every year (i-generation?).
- The demand on „orientation knowledge“ is increasing sharply, since the job perspectives are in flux creating severe uncertainty.

Can we meet this additional Challenge by Assuming that Learning/Teaching

- is a project oriented tool based process with strong emphasis on self learning, and
- is a social process (continuous exchange and presentation opportunities are part of the learning process)?

In my Opinion yes, if we

are successful in:

- Providing the flavor/feeling about the subject
- Distinguishing between **available** and **orientation** knowledge
- Interpreting Questions as a coming up reflection of a difference
- Viewing Answers as tools backed by actual industry-examples

Back to Model Checking at UAS

- What worked well?
- What didn't work well?
- What additional tools may help?
- Can students apply TLA+ to real problems when they get into industry?

What worked well?

- The strong emphasis on simple examples lead to a rather good understanding of the basic concepts of TLA+
- The use of the Toolbox was quite well practiced and where used heavily by all students
- Hyperbook was used as well, and new concepts got understood in selfstudy (PlusCal)

What didn't work well?

- The focus on simple examples seemed to prohibit the understanding of the general applicability of the TLA concepts
- To what extent the understanding of the principles of the TLC is required for a more advanced use of the Toolbox is left unclear
- Most of the Hyperbook examples are considered as too „mathematical“ and less „system oriented“

What additional Tools may help?

Three directions may be considered:

1. Round up the Hyperbook to be a perfect basis for all theories and concepts relevant to apply TLA+/PlusCal for novice users
2. Add examples which make the systemic and process views for various application areas more obvious
3. Provide for all cross links to enable an uniform view of the whole

Conclusion

- Focus should be directed to widen the community by attracting more educational institutions
- Systems examples from industry and research should widely be available and communicated

and Future Work

- Within UAS we will try to apply the concepts to fit TLA+ in with the other topics, such as theory of computation and UML (is emphasized, but not yet done completely)