University of Paderborn

„Agent Dependability in Open Architectures“

- Safety is different! -

Holger Giese
Software Engineering Group
University of Paderborn
Warburger Str. 100
D-33098 Paderborn, Germany
hg@upb.de

What is Dependability?

Dependability is that property of a computer system such that ***reliance*** can justifiably be placed on the service it delivers.

[J.C. Laprie]

Attributed of dependability:

- Reliability/Availability
- Safety
- Confidentiality
- Integrity
- Maintainability

Most Attributes fits well to MAS

- **Reliability/Availability:** MAS naturally lead to heterogeneous and redundant processing of tasks
  ⇨ MAS ***can*** result in high reliability and availability
- **Confidentiality:** The design of open agent systems naturally includes the question whether agents can trust each other or not.
- **Integrity:** The strong encapsulation provided by agents builds a natural basis for integrity.
- **Maintainability:** The flexibility and adaptability of open agent architectures as well as the strong encapsulation provided by agents simplifies maintenance to a great extend.

⇨ MAS **can** be made dependable (according to these attributes)

But this is still a challenging task!

But not Safety …

> **Safety** is a property of a system that it will not endanger human life or the environment. (Storey1996)

Problems for safe MAS:

- Open MAS result in
  - More complex or even open systems
  - Interaction and emergence
  - Adaptation/learning

  ⇨ **non-predictable** behavior
- Safety is a system property!
  - Safe agents do **not** result in a safe MAS
  - A MAS is only safe w.r.t. a given **environment**

⇨ Open MAS and safety do ***not*** fit! Why?

Why? Evolution of Social System

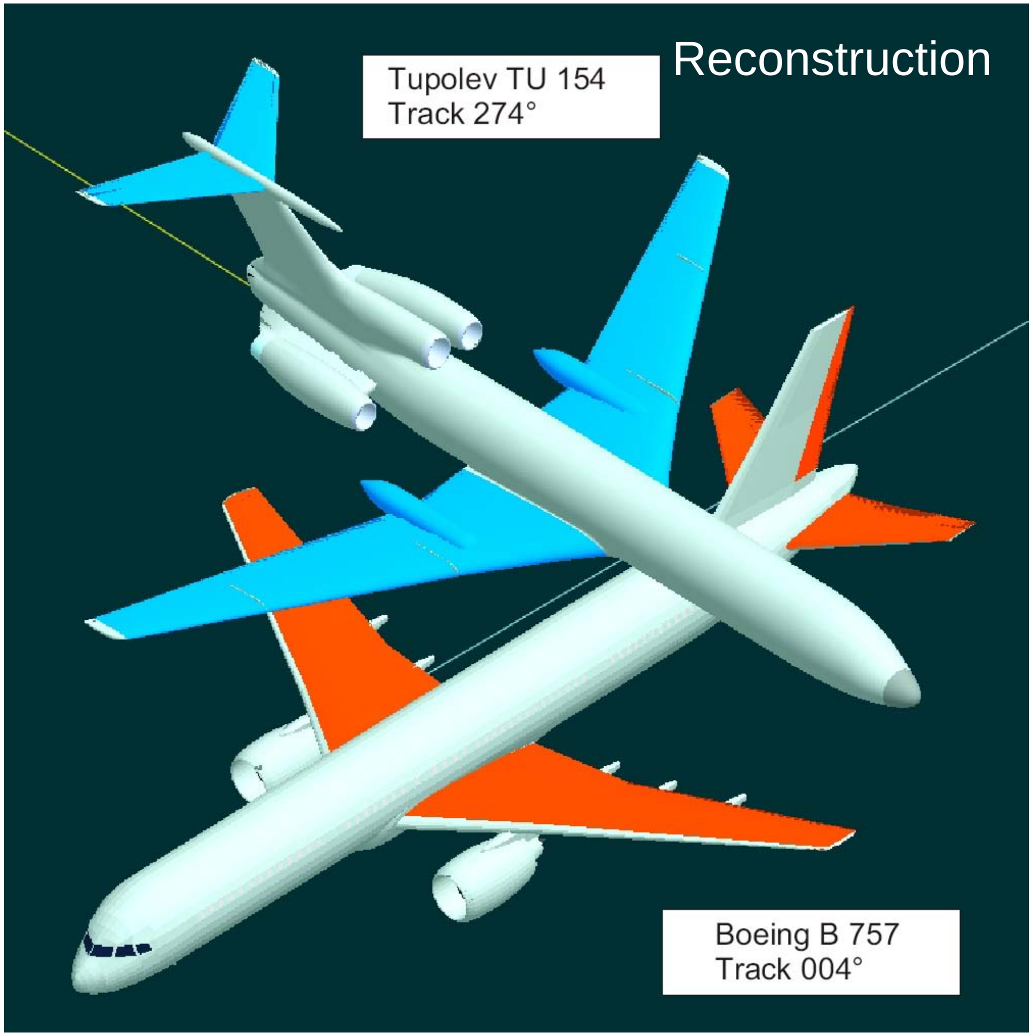
- MAS are inspired by the intelligent social behavior observed for humans and other natural species which have resulted from (cultural) evolution.
  - □ Social system “designs” **must ensure** some degree of reliability, availability, confidentiality, integrity, and maintainability to survive
    ⇨ the general architectural principles reflect them
  - □ A social system “design” only survives evolution when it protects enough members from server undesired consequences
    ⇨ **one specific design** is “safe” for **one specific context**
  - □ The environment will only be protected by a natural social system when destroying the environment would have server consequences for the social system itself
    ⇨ **safety** for the environment occurs only in rare “altruistic” cases
  - □ Natural social systems tend to continue operation (being reliable or at least available) under all circumstances while safety-critical systems can behave **fail-safe** assuming that the operators will restart them.

Safety is different!

- Safety always includes the **members** of the society as well as **environment**
- Safety has usually higher demands than only enough surviving society members (ethic)!
- In a safety-critical system catastrophic accidents have to be limited to a probability of at most 10^{-4} per year (SIL 4), while in natural social systems much lower probability bounds are sufficient to survive.

Safety cannot simply be programmed into MAS using the common architectural principles!

Aeroplane Crash Bodensee July 2002



- Traffic alert and Collision Avoidance System (TCAS) works in both machines
- Short Term Conflict Alert (STCA) was not available at ACC Zurich due to maintenance
- The STCA of the Upper Area Control Center warns two minutes prior to collision, but no connection via the direct line could be established

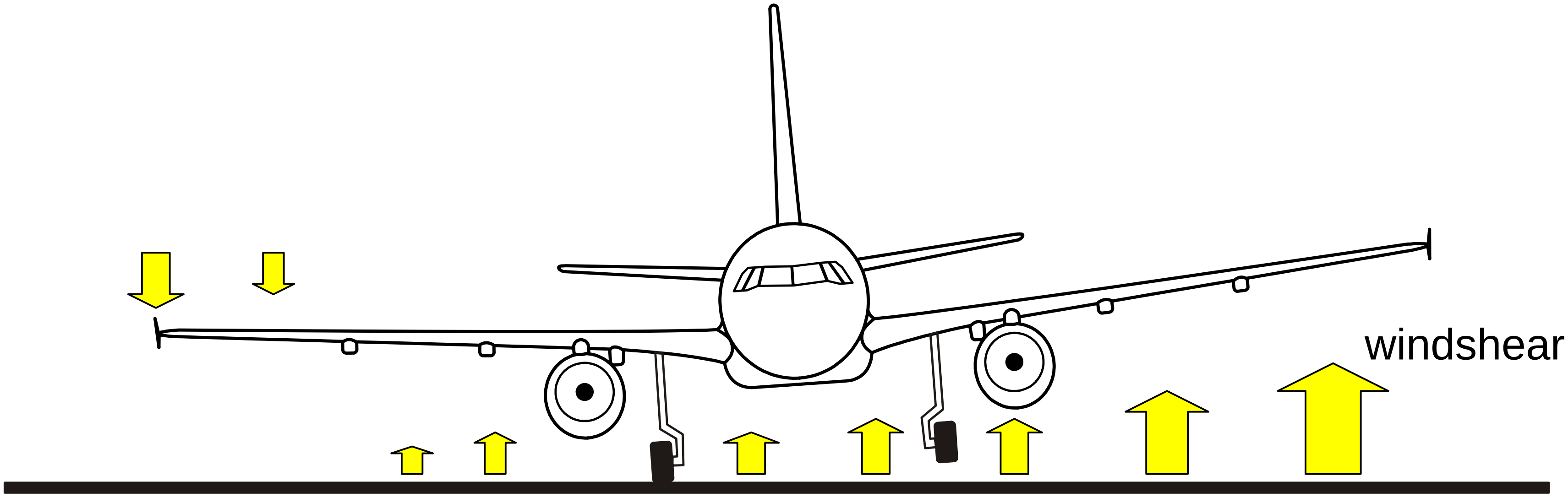
⇒Interaction (MAS) can improve safety!

A320-211 Accident in Warsaw

A high-tech aeroplane with special safety features happens to be ***too*** intelligent.

A320-211 Accident in Warsaw

- The aircraft was cleared for a Warsaw runway 11 approach and were told of the existence of windshear on the approach. The Airbus' right gear touched down 770m from the runway 11 threshold. The left gear touched down ***9 seconds later***, 1525m from the threshold.

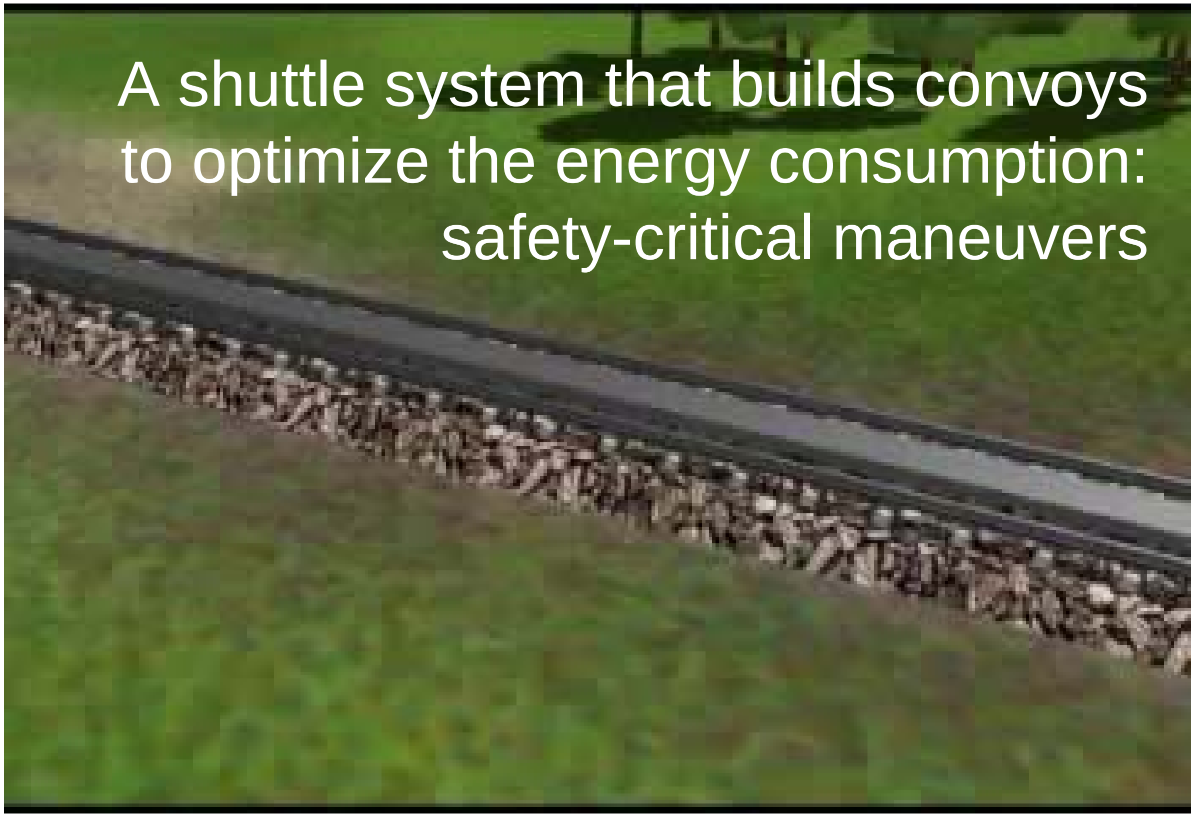


Example: “Intelligent” Shuttles

Intelligent mechatronic agents:

- System of systems
- networked,
- hard real-time,
- safety-critical,
- embedded, and
- will contain complex software.

How to ensure safety?

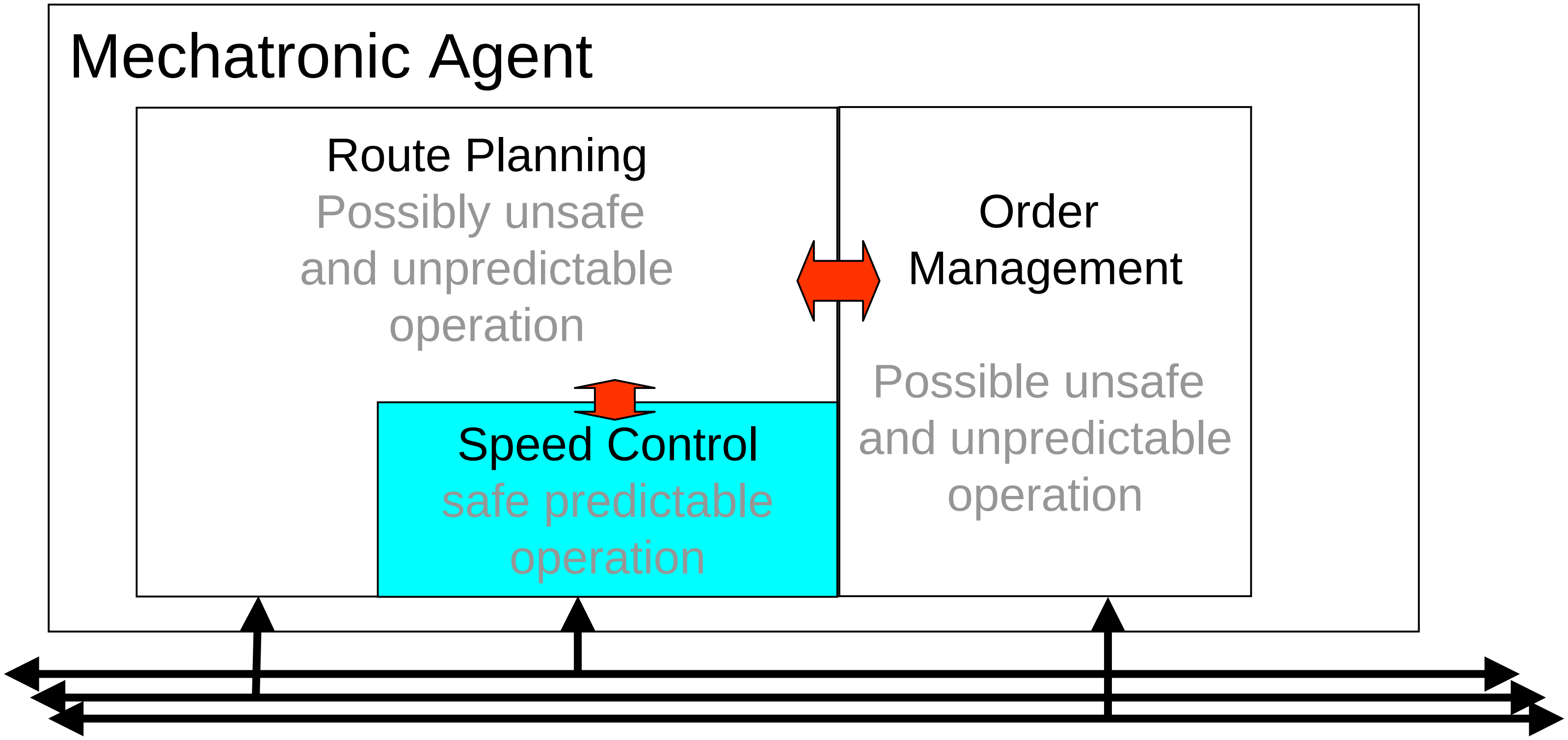


A shuttle system that builds convoys to optimize the energy consumption: safety-critical maneuvers

New Railway Technology Paderborn: http://nbp-www.upb.de/en/index.php
SFB 614: http://www.sfb614.de/eng/index.htm

Separate Safety-Related Concerns

Restrict at least the safety-critical concerns of an open MAS architecture to design principles which enable their independent analysis w.r.t. safety.


Mechatronic Agent
Route Planning
Possibly unsafe and unpredictable operation
Speed Control
safe predictable operation
Order Management
Possible unsafe and unpredictable operation

Conclusion

- Open MAS architectures fit very well to all dependability attributes except safety
- safety could only be “borrowed” from nature in rare altruistic cases and when the context/design is exactly the same
  ⇨ usually not!
- The requirements for safety-critical systems are much more demanding than the ones in nature!

Our proposal for intelligent mechatronic agents:

- Separation of concerns, design for safety, and formal methods can enable us to build safe open MAS systems, if we restrict the relevant concerns in such a manner that we can ensure safety.