Dependability of Autonomous Robots

Gerald Steinbauer
Institute for Software Technology
Graz University of Technology
Motivating Example - Officially
Motivating Example - Unofficially
The natural enemies of dependability

- Environment
- Decision Making
- Perception

Failures:
- Hardware
- Execution
- Incomplete/Wrong Knowledge
- Sensing
- Exogenous Events
- Software
- Algorithm
Simple Model-Based Reasoning Example

**Model**

- $\neg AB(B) \land \neg AB(L_1) \rightarrow light(L_1)$
- $\neg AB(B) \land \neg AB(L_2) \rightarrow light(L_2)$
- $\neg AB(L_1) \land light(L_2) \rightarrow light(L_1)$
- $\neg AB(L_2) \land light(L_1) \rightarrow light(L_2)$

**Assumption**

- $\rightarrow \neg AB(B) \land \neg AB(L_1) \land \neg AB(L_2)$

**Observation**

- $\rightarrow light(L_1) \land \neg light(L_2)$

**System**

- $B$
- $L_1$
- $L_2$

**Contradiction**

**Root Cause**
Software Failures

- independent software modules
- diagnosis is based on the communication patterns between modules

\[ \neg AB(MOTION) \implies ok(\text{odometry}) \]
\[ \neg AB(\text{TRACKER}) \land ok(\text{pose}) \land ok(\text{local_objects}) \implies ok(\text{global_objects}) \]
\[ ok(\text{pose}) \implies \neg AB(\text{SELF\_LOC}) \]

[Steinbauer and Wotawa, IJCAI, 2005]
Monitoring, Diagnosis & Repair

• monitoring connections by observers
  – periodic event production
  – conditional event production
  – periodic method call
  – observer generate the observations

• diagnosis
  – triggered if a observer recognized a violation
  – model-based diagnosis (Reiter + LTUR [Minoux 1988])

• repair
  – planned restart of the effected modules (direct or indirect)

• experiments
  – successful automated recover from deadlocks and crashes
Hardware Failures

[Brandstötter, Hofbaur, Steinbauer and Wotawa, IROS, 2007]
Diagnosis/Repair Omni-Drive
Combine both Worlds for ROS

[Lepej et.al, DX 2012]
Perception-Execution-Failures

Task: bring the letter to room B and the folder to C

Actions: goto(l), grab(o), release(o)
Multiple Explanations/Beliefs

**Background Model**

\[
\text{at}(l) \land \text{see}(o) \rightarrow \text{isat}(o,l) \\
\text{isat}(o,l_1) \land \text{isat}(o,l_2) \land l_1 \neq l_2 \rightarrow \bot
\]

[goto(A), grab(Letter), goto(B), release(Letter), goto(D), grab(Folder), goto(C), see(Calculator)]

[goto(A), grab(Letter), goto(B), release(Letter), goto(D), grab(Folder), goto(A), see(Calculator)]
Belief Management

- idea “to explain what **went** wrong rather then what is faulty”
- a situation calculus history \( h \) is **sufficient** to describe the world
- if an observation \( \phi \) **contradicts** a history \( h \) we look for an alternative **consistent** history \( h' \) [Iwan, AlCom 02]
- alternative hypotheses are generated by
  - variation of actions in \( s \), e.g. \( goto(A) \) instead of \( goto(C) \), \( sense(\text{none}) \) instead of \( sense(calc) \)
  - insertion of exogenous events, e.g. \( teleport(calc,C) \)
- integration of the system into **IndiGolog** agent programming framework [Giacomo et al., MAP 2009]
Example – Irritated Delivery Robot

[Gspandl et.al, IJCAI, 2011], [Gspandl et.al, ICRA, 2012]
Evaluating Dependability of Autonomous Systems

• we want to evaluate how dependable a system is
• compare how “good” systems are able to cope with the real world – no classical software testing
• but reuse concepts from software testing

• in particular industry needs and ask for certification
• most “intelligent” robot use some kind of non-deterministic algorithms
• non-determinism not allowed in higher SIL
Evaluation Components

- **Mission/Task**
  - clear definition
  - initial and goal state
  - restrictions

- **Quality Criteria**
  - objective and measurable
  - fair
  - reflects the domain properties

- **Fault Injection**
  - set of important faults
  - various natures
  - injection process

- **Test Framework**
  - setup
  - orchestration of the evaluation
  - evaluation support and automation
But the Target System matters …
Conclusion

• **dependability of autonomous systems**
  • is an important property in the **real world**
  • touches a wide field of **problems**
  • requires diagnosis-repair-reasoning **capabilities**
  • can be improved by model-based **reasoning** approaches

• **evaluation of dependability**
  • interaction with the real world
  • important for research – **performance** evaluation of methods
  • important for industry – **certification**
Future Work

- combination/interoperation of methods – complexity
- domain analysis - properties
- formal definition of domain complexity classes
- cooperation with social sciences and psychology - well established methods for skill evaluation, e.g. Item Response Theory
Thank you!