# Ontological Analysis of communication-bus behavior

## WBA & CausalML User Group
## Bieleschweig v5.5

Jörn Stuphorn
stuphorn@rvs.uni-bielefeld.de

Universität Bielefeld
Technische Fakultät



R|V|S
Rechnernetze und
Verteilte Systeme

# Ontological Analysis

- Method for requirement development
- Based on ontological system description
  - Objects
  - Relations
    - un-ary (properties)
    - n-ary (relations)
- Iterative expansion of ontology
  - beginning with simple system
    - extreme case:
      - one object
      - only unary relations
    - typical case:
      - objects and relations based on experience and knowledge of an abstract system description

Bielefeld, 7 June 2005

# Ontological Analysis

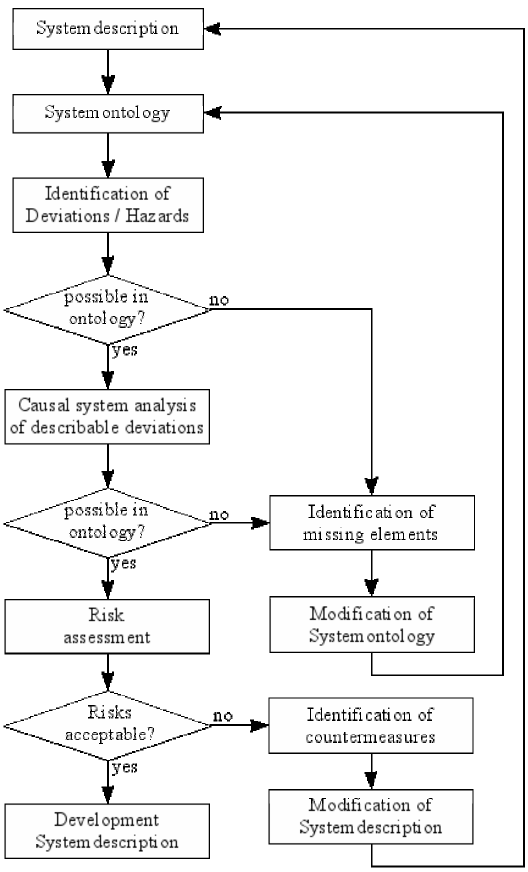
R|V|S Rechnernetze und Verteilte Systeme

- Failures of the system are described as causal relations in the ontology
  - Failures have to be present
    - inserted from outside „expert knowledge“
    - systematically developed
- Risks inherited by the system are determined
- Based on results
  - ontology is extended



# Ontological Analysis

R|V|S Rechnernetze und Verteilte Systeme




Bielefeld, 7 June 2005

# The analysed System

RVS Rechnernetze und Verteilte Systeme

- Communication Network for future automotive use
- Transmission modes
  - time-triggered
  - event-triggered
- Currently deployed communication systems use event-triggered transmission
  - CAN
  - J1850
  - LIN
- Future Communication systems will probably use time-triggered transmission to facilitate
  - X-by-wire
  - Powertrain



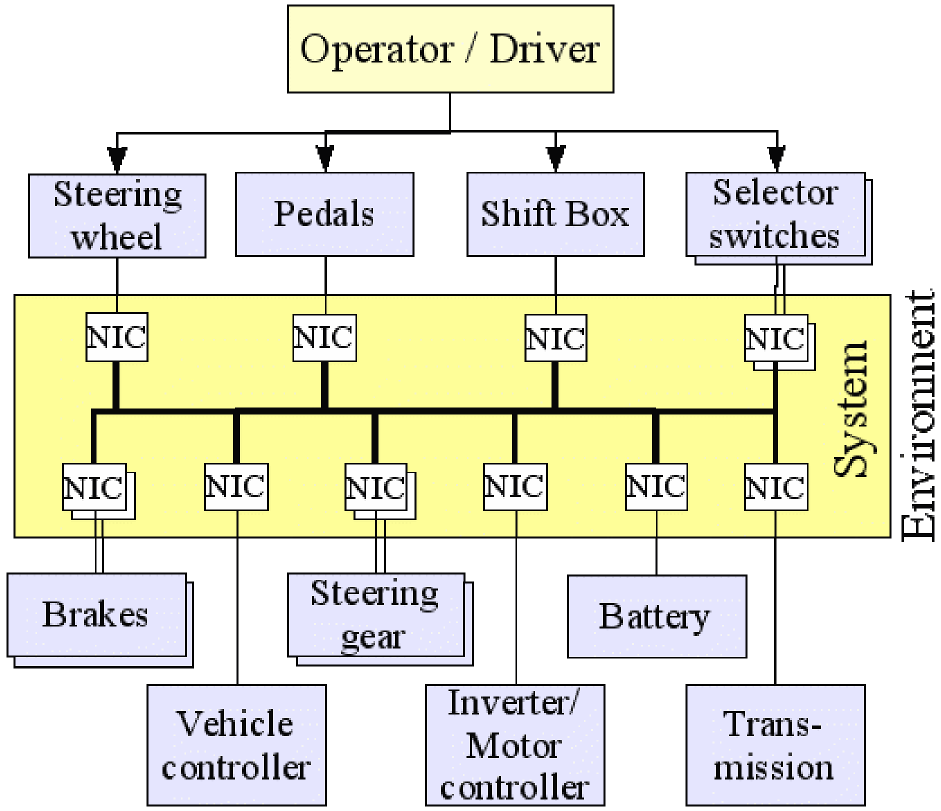
# X-by-Wire

RVS Rechnernetze und Verteilte Systeme

- currently developed system
  - like Fly-by-Wire in aircraft construction
  - Interconnection of automotive systems without mechanical fallback solution
- Motivation
  - weight reduction
  - simpler integration of drive assistance programs
- Problem
  - System has to be very reliable
  - existing systems for aircraft very expensive
  - very high number of units in automotive industry
  - existing technology should be integrable

# System Schematics

R|V|S Rechnernetze und Verteilte Systeme






# Initial System Ontology

R|V|S Rechnernetze und Verteilte Systeme

- Objects
  - NIC
  - Wiring
  - Transmission
- Relations
  - Connection (Wiring, NIC)
- Properties (unary relations)
  - Input(NIC)
  - Output(NIC)
  - Intact(NIC)
  - Intact(Wiring)
  - Size(Transmission)
  - Deadline(Transmission)
  - Period(Transmission)
  - Mode(Transmission)
  - Latency(Transmission)
  - Jitter(Transmission)
- Every element of the ontology has to be accurately defined!

# HAZOP



- **HAZ**ard and **OP**erability Study
  - Group of experts
  - „what would happen, if a component would operate outside its normal design mode“
- Guide-words
  - Group agrees on a set of guide-words
  - Typical sets developed by
    - Royal Society of Chemistry (CISHEC)
      *A Guide to Hazard and Operability Studies, 1977*
    - Redmill, Chudleigh, Catmur
      *System Safety: HAZOP and Software HAZOP, 1999*



# HAZOP - „sentences“



- Guide-words applied to components form sentences

| ATTRIBUTE: | Intact(NIC) |
|---|---|
| GUIDE WORD | INTERPRETATION |
| No | The NIC is not intact |
| More | The NIC is more than intact |
| Less | The NIC is less than intact |
| As well as | The NIC is more than intact |
| Part of | The NIC is only in part intact |
| Reverse | The concept of intact is reversed |
| Other than | The concept of intact is replaced |
| Early | The concept of intact happens early |
| Late | The concept of intact happens late |
| Before | The concept of intact happens before something |
| After | The concept of intact happens after something |
| Faster | The concept of intact is faster than intended |
| Slower | The concept of intact is slower than intended |

# HAZOP - „deviations“



- Sentences have to be analysed in respect of failures
- Usage of assumptions can influence analysis

**Comments on the formed HAZOP sentences**

**More, Less, As well as, Part of** An object can be either intact or not but cannot be in between these states.

**Reverse** The concept of intact aims at assessing the functionality of the NIC. Reversing the concept would only result in switching the labels. The assessment of the functionality would not be limited.

**Other than** A NIC has to be intact to be able to operate. Because of this Intact(NIC) is irreplaceable.

**Early, Faster** If the NIC is intact early or faster than needed, it will be intact any time afterwards until it breaks. This is the expected state of Intact(NIC) and no threat.

**Late, Slower** If the NIC would achieve its functionality later than needed, it would not be intact at the required time. This deviation is identified with "No Intact(NIC)".

**Before, After** If the NIC is intact before an event it can be assumed that it will be intact at the time of the event until stated otherwise. Likewise it can be assumed that if the NIC is intact after an event it was intact before the event if not stated otherwise. These are the expected states of Intact(NIC) and no threats.



# Results of HAZOP

R|V|S Rechnernetze und Verteilte Systeme

- List of deviations
  - Deviations are causally analysed using CIDs
- List of assumptions made in the interpretations
  - Assumptions have to be ascertained
  - If this cannot be done
    Countermeasures must be integrated

*WBA & CausalML User Group (Bieleschweig v5.5)* Bielefeld, 7 June 2005

# Translation of deviations

R|V|S Rechnernetze und Verteilte Systeme

- Deviations are expressed using the ontology as a kind of language
- Deviations not translatable lead to
  - extension of ontology
  - These deviations are analysed in later iterations

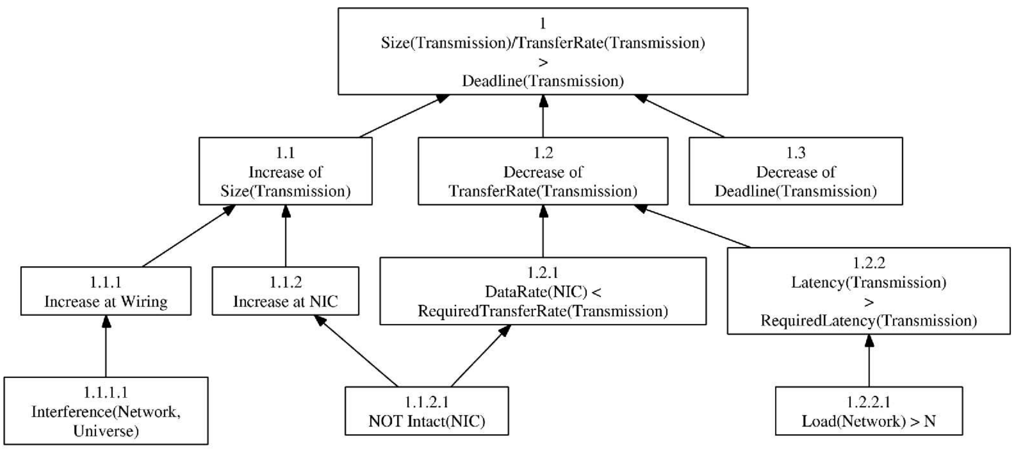
| | Deviation | | Ontological analogue |
|---|---|---|---|
| 1.a | More NICs in system than expected | 1.a | NodeCount(Network) > DesignNodeCount(Network) |
| 1.b | Less NICs in system than expected | 1.b | NodeCount(Network) < DesignNodeCount(Network) |
| 1.c | A NIC is fragmented | 1.c | Intact(NIC) = False |
| 2.a | Wiring too long | 2.a | Length(Wiring) > RequiredDeadline(Transmission) $*2.0*10^8\frac{m}{s}$ |
| 2.b | Wiring too small | 2.b | $\{\exists(NIC\ i)\|(Connection(Wiring, i) = FALSE)\}$ |
| 2.c | Other medium in addition to wiring present | 2.c | Wiring a $\wedge$ Wiring b |
| 2.d | Wiring meets design intention only in part | 2.d | *needed: Design(Wiring)* |



# CID

R|V|S Rechnernetze und Verteilte Systeme

- Causal factors of Analogues are identified
- Analysis is stopped, if
  - Elements not in ontology are needed
  - Relation of an element was identified



# Risk Assessment

RVS Rechnernetze und Verteilte Systeme

- Risks described in ontology:
  - Risk of Relations not being met
  - Risk of Assumptions not being fulfilled
- Risk of Analysis process:
  - Risk of Analysis not being complete



# Unfulfilled Relations

RVS Rechnernetze und Verteilte Systeme

- Bayesian Belief Network can estimate risk of a CID (given no circular influences occur)
- Risk of unfulfilled relations:
  - Knowledge
  - Resulting Risk from other Deviations

## Assumptions not fulfilled



- Assumption can be
  - trivial
    - *value can be computed instantaneously*
    - *an Element will be present*
  - complex
    - *computation of a value is done without systematic mistake*
    - *Attributes will not interfere with other attributes*



## Assumptions not fulfilled



- *Risk in trivial Assumptions*
  - estimated by Knowledge
  - Requirements towards design process
- Risk in complex Assumptions
  - Requirements towards design process
  - Risk of assumption not fulfilled guarded by countermeasures
- Countermeasures
  - Countermeasures extend Ontology
  - Impact of countermeasures on the system must be analysed in following iterations

# Incomplete Analysis

RVS Rechnernetze und Verteilte Systeme

- Omitted Elements in Analysis can be problematic
  - Elements of ontology
  - Deviations
- **Omitted elements of ontology**
  - Ontological analysis is iterative
  - Starting with simple system description
  - Refining system description with each iteration
  - Statements made for ontology in one iteration is valid in all following iterations
- Elements of ontology can only be omitted if the ontology development is interrupted prematurely



# Incomplete Analysis

RVS Rechnernetze und Verteilte Systeme

- Omitted Elements in Analysis can be problematic
  - Elements of ontology
  - Deviations
- **Omitted deviations**
  - HAZOP process identifies possible dangers in system operation
  - HAZOP is a systematic approach
- If the guide-words are complete all sentences leading to deviations will be identified
- If the group identifies all deviations posed by sentences this will be complete if the set of guide-words was complete

# Results

RVS Rechnernetze und Verteilte Systeme

- The HAZOP method led to large number of deviations
- The translation into ontological analogues identified identical deviations

- Ontological description in combination with HAZOP leads to refinement of system description
  - System description only describes dependencies within the sytem
  - Assumptions can be used to control refinement of system

- Countermeasures are not automatically identified
  - Assumptions can lead to countermeasures



# Results

RVS Rechnernetze und Verteilte Systeme

- $1^{st}$ iteration
  - 3 objects
  - 10 properties (unary relations)
  - 1 relation

- $2^{nd}$ iteration
  - 6 objects
  - 31 properties (unary relations)
  - 2 relations

- $3^{rd}$ iteration
  - 6 objects
  - 45 properties (unary relations)
  - 3 relations

# Results

R|V|S Rechnernetze und Verteilte Systeme

- 1st iteration
  - 59 deviations
  - 19 describable in ontology
  - quota: 32.2%
- 2nd iteration
  - 146 deviations
  - 123 describable in ontology
  - quota: 82.2%
- 3rd iteration
  - 181 deviations
  - 172 describable in ontology
  - quota: 95.0%



# Results

R|V|S Rechnernetze und Verteilte Systeme

- The ontological analysis produces very much documentation
  - The meaning of every element in the system description / ontology has to be defined
- It can be used for justification of decisions made in the development process
  - „I may be wrong, but my decision was based on these assumptions“
- Size of group leads to bigger reliability in the number of deviations identified
  - Even a small group (e.g. one „expert“) develops fine system description using iteration process
  - Examples for identified elements after 3 iterations:
    - Shielding(Network)
    - EmissionRegulation(Transmission)

# Thank you
# for your attentiveness.