Semantics and Verification of Software

Lecture 26: Wrap-Up

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)

RWTH Aachen University

noll@cs.rwth-aachen.de

http://www-i2.informatik.rwth-aachen.de/i2/svsw08/

Winter semester 2008/09

Outline

Semantics of Functional Languages I

- Program = list of function definitions

Semantics of Functional Languages I

- Program = list of function definitions
- Simplest setting: first-order function definitions of the form
$$f(x_1, \ldots, x_n) = t$$
  - function name f
  - formal parameters $x_1, \ldots, x_n$
  - term t over (base and defined) function calls and $x_1, \ldots, x_n$

Semantics of Functional Languages I

- Program = list of function definitions
- Simplest setting: first-order function definitions of the form

$$f(x_1, \dots, x_n) = t$$

  - function name f
  - formal parameters $x_1, \dots, x_n$
  - term t over (base and defined) function calls and $x_1, \dots, x_n$
- Operational semantics (only function calls)
  - call-by-value case:

$$\frac{t_1 \to z_1 \quad \dots \quad t_n \to z_n \quad t[x_1 \mapsto z_1, \dots, x_n \mapsto z_n] \to z}{f(t_1, \dots, t_n) \to z}$$

  - call-by-name case:

$$\frac{t[x_1 \mapsto t_1, \dots, x_n \mapsto t_n] \to z}{f(t_1, \dots, t_n) \to z}$$

Semantics of Functional Languages II

- Denotational semantics
  - program = equation system (for functions)
  - induces call-by-value and call-by-name functional
  - monotonic and continuous w.r.t. graph inclusion
  - semantics := least fixpoint (Tarski/Knaster Theorem)
  - coincides with operational semantics

Semantics of Functional Languages II

- Denotational semantics
  - program = equation system (for functions)
  - induces call-by-value and call-by-name functional
  - monotonic and continuous w.r.t. graph inclusion
  - semantics := least fixpoint (Tarski/Knaster Theorem)
  - coincides with operational semantics
- Extensions: higher-order types, data types, ...

Semantics of Functional Languages II

- Denotational semantics
  - program = equation system (for functions)
  - induces call-by-value and call-by-name functional
  - monotonic and continuous w.r.t. graph inclusion
  - semantics := least fixpoint (Tarski/Knaster Theorem)
  - coincides with operational semantics
- Extensions: higher-order types, data types, ...
- see [Winskel 1996, Sct. 9] and *Functional Programming* course [Giesl]

Semantics of Concurrent Languages

- **Problem:** "classical" view of sequential systems

$$\text{Program} : \text{Input} \rightarrow \text{Output}$$

not adequate for concurrent settings

- **Problem:** "classical" view of sequential systems

  $$\text{Program} : \text{Input} \rightarrow \text{Output}$$

  not adequate for concurrent settings
- Missing: aspect of interaction

Semantics of Concurrent Languages

- **Problem:** "classical" view of sequential systems

$$\text{Program} : \text{Input} \rightarrow \text{Output}$$

  not adequate for concurrent settings
- Missing: aspect of interaction
- Typical approach:
  - concurrency modelled by interleaving
  - interaction modelled by (explicit) communication

Semantics of Concurrent Languages

- **Problem:** "classical" view of sequential systems

$$\text{Program} : \text{Input} \rightarrow \text{Output}$$

  not adequate for concurrent settings
- Missing: aspect of interaction
- Typical approach:
  - concurrency modelled by interleaving
  - interaction modelled by (explicit) communication
- Example: Milner's *Calculus of Communicating Systems (CCS)*

Semantics of Concurrent Languages

- **Problem:** "classical" view of sequential systems

$$\text{Program} : \text{Input} \rightarrow \text{Output}$$

  not adequate for concurrent settings
- Missing: aspect of interaction
- Typical approach:
  - concurrency modelled by interleaving
  - interaction modelled by (explicit) communication
- Example: Milner's *Calculus of Communicating Systems (CCS)*
- Syntax: $P ::= 0 \mid \alpha.P \mid P_1 + P_2 \mid P_1 \parallel P_2 \mid \ldots$

Semantics of Concurrent Languages

- **Problem:** "classical" view of sequential systems

$$\text{Program} : \text{Input} \longrightarrow \text{Output}$$

  not adequate for concurrent settings
- Missing: aspect of interaction
- Typical approach:
  - concurrency modelled by interleaving
  - interaction modelled by (explicit) communication
- Example: Milner's *Calculus of Communicating Systems (CCS)*
- Syntax: $P ::= 0 \mid \alpha.P \mid P_1 + P_2 \mid P_1 \parallel P_2 \mid \ldots$
- (Operational) Semantics: labelled transition systems defined by transition rules of the form

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \quad \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \quad \frac{P \xrightarrow{\alpha} P'}{P \parallel Q \xrightarrow{\alpha} P' \parallel Q} \quad \frac{P \xrightarrow{\alpha} P' \quad Q \xrightarrow{\bar{\alpha}} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'} \quad \ldots$$

Semantics of Concurrent Languages

- **Problem:** "classical" view of sequential systems

  $$\text{Program} : \text{Input} \to \text{Output}$$

  not adequate for concurrent settings
- Missing: aspect of interaction
- Typical approach:
  - concurrency modelled by interleaving
  - interaction modelled by (explicit) communication
- Example: Milner's *Calculus of Communicating Systems (CCS)*
- Syntax: $P ::= 0 \mid \alpha.P \mid P_1 + P_2 \mid P_1 \parallel P_2 \mid \ldots$
- (Operational) Semantics: labelled transition systems defined by transition rules of the form

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \quad \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \quad \frac{P \xrightarrow{\alpha} P'}{P \parallel Q \xrightarrow{\alpha} P' \parallel Q} \quad \frac{P \xrightarrow{\alpha} P' \quad Q \xrightarrow{\bar{\alpha}} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'} \quad \ldots$$

- see course on *Modelling Concurrent and Probabilistic Systems* in Summer 2009 [Katoen, Noll] and [Winskel 1996, Sct. 14]

Outline

1. Further Topics in Formal Semantics
2. Topics for Diploma and Master's Theses
3. Upcoming Courses and Seminars
4. Evaluation of the Course

The COMPASS Project

COMPASS

Correctness, Modelling and Performability of Aerospace Systems

The COMPASS Project

COMPASS

Correctness, Modelling and Performability of Aerospace Systems

Current Situation

Yes, formal methods are applied for aerospace systems, but not in a coherent manner at the systems engineering level

The COMPASS Project

COMPASS

Correctness, Modelling and Performability of Aerospace Systems

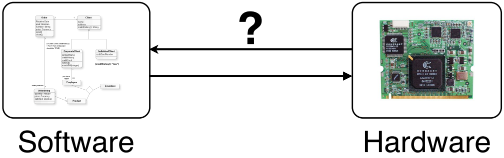
Current Situation

Yes, formal methods are applied for aerospace systems, but not in a coherent manner at the systems engineering level

Systems Engineering

"Identification and quantification of system goals, creation of alternative system design concepts, performance of design trades, selection and implementation of the best design, verification that the design is properly built and integrated, and post-implementation assessment of how well the system meets (or met) the goals."

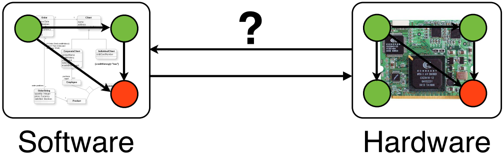
- NASA's Systems Engineering Handbook, 1995



Coherency issues

- Co-engineering

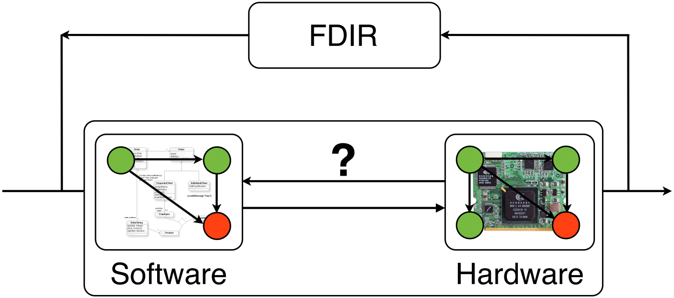
Evaluating Critical Embedded Systems is Hard



Coherency issues

- Co-engineering
- Analysis of degraded modes of operation

Evaluating Critical Embedded Systems is Hard



Coherency issues

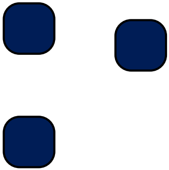
- Co-engineering
- Analysis of degraded modes of operation
- Assessment of Fault Detection, Isolation and Recovery analysis

ESA's Wish

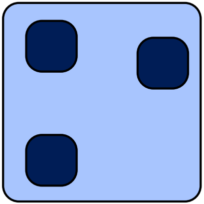
"Provide a unified modelling language that is amenable for validation and verification"

Three Components

- Modelling language
- Verification and validation activities
- Toolset

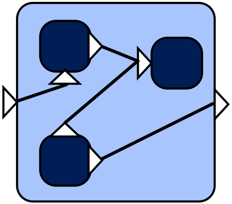


- Component-oriented



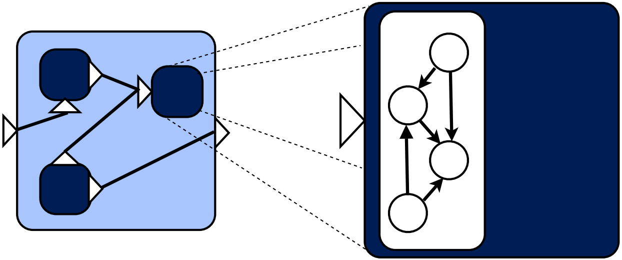
- Component-oriented
- Sub/supercomponents

System-Level Integrated Modelling Language



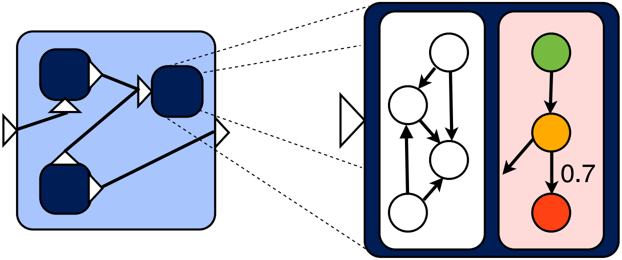
- Component-oriented
- Sub/supercomponents
- Event/data ports

System-Level Integrated Modelling Language



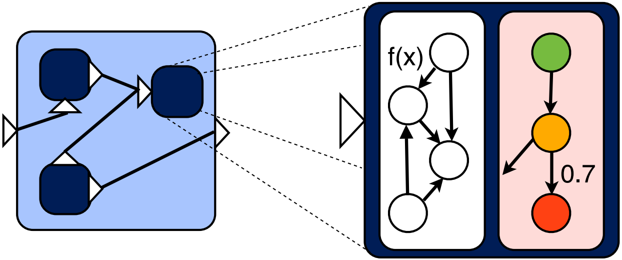
- Component-oriented
- Sub/supercomponents
- Event/data ports
- (Functional) nominal behaviour

System-Level Integrated Modelling Language



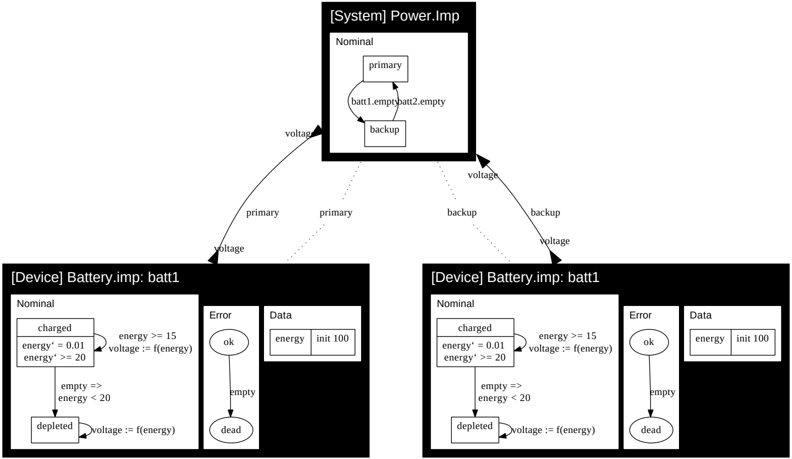
- Component-oriented
- Sub/supercomponents
- Event/data ports
- (Functional) nominal behaviour
- (Probabilistic) error behaviour

System-Level Integrated Modelling Language



- Component-oriented
- Sub/supercomponents
- Event/data ports
- (Functional) nominal behaviour
- (Probabilistic) error behaviour
- Hybrid behaviour

SLIM Example: Battery



SLIM Example: Battery

```
device type Battery
  features
    empty: out event port;
    voltage: out data port real;
end Battery;

device implementation Battery.Imp
  subcomponents
    energy: data continuous initially 100;
  modes
    charged: initial mode
             while energy'=-0.01 and energy>=20;
    depleted: mode
              while energy'=-0.015;
  transitions
    charged -[when energy>=15
              then voltage:=f(energy)]->
      charged;
    charged -[empty when energy<20]->
      depleted;
    depleted -[then voltage:=f(energy)]->
      depleted;
end Battery.Imp;
```

SLIM Example: Redundant Battery System

```
system Power
  features
    voltage: out data port real;
end Power;

system implementation Power.Imp
  subcomponents
    batt1: device Battery.Imp in modes (primary);
    batt2: device Battery.Imp in modes (backup);
  connections
    data port batt1.voltage -> voltage
      in modes (primary);
    data port batt2.voltage -> voltage
     in modes (backup);
  modes
    primary: initial mode;
    backup: mode;
  transitions
    primary -[batt1.empty]-> backup;
    backup -[batt2.empty]-> primary;
end Power.Imp;
```

SLIM Example: Specifying Fault Behaviour

```
error model BatteryFailure
  features
    normal: initial state;
    dead: error state;
end BatteryFailure;

error model implementation BatteryFailure.Imp
  events
    fault: error event occurrence poisson 5;
  transitions
    normal -[fault]-> dead;
end BatteryFailure.Imp;
```

SLIM Example: Specifying Fault Behaviour

```
error model BatteryFailure
  features
    normal: initial state;
    dead: error state;
end BatteryFailure;

error model implementation BatteryFailure.Imp
  events
    fault: error event occurrence poisson 5;
  transitions
    normal -[fault]-> dead;
end BatteryFailure.Imp;
```

Fault injection: in error state `dead`, `voltage:=0`

Topics for Theses

1. Visualization of SLIM Specifications
    - tool: SLIM specification $\rightarrow$ graphical representation
    - visualization of hierarchical system structure and component interconnections
    - challenge: support dynamic reconfiguration

Topics for Theses

1. Visualization of SLIM Specifications
2. Translation of SLIM into PRO[B]MELA
    - PROMELA: input language of SPIN model checker
    - re-use for validating SLIM specifications
    - probabilistic extension called PROBMELA

Topics for Theses

1. Visualization of SLIM Specifications
2. Translation of SLIM into PRO[B]MELA
3. Translation of SLIM into UPPAAL
    - UPPAAL: tool for modeling, validation and verification of real-time systems
    - modeled: networks of timed automata, extended with data types
    - re-use for validating SLIM specifications

Topics for Theses

1. Visualization of SLIM Specifications
2. Translation of SLIM into PRO[B]MELA
3. Translation of SLIM into UPPAAL
4. TERMA Case Study
   - case study from Quasimodo project
   - goal: model (abstraction of) HW and SW of Attitude and Orbit Control System (AOCS) of Herschel and Planck satellites

Topics for Theses

1. Visualization of SLIM Specifications
2. Translation of SLIM into PRO[B]MELA
3. Translation of SLIM into UPPAAL
4. TERMA Case Study
5. Formal Semantics of SLIM Language
    - hierarchical semantics has been developed
    - to be done: "flat" semantics and hybridity

Topics for Theses

1. Visualization of SLIM Specifications
2. Translation of SLIM into PRO[B]MELA
3. Translation of SLIM into UPPAAL
4. TERMA Case Study
5. Formal Semantics of SLIM Language
6. Minimization of SLIM Models
    - problem: state-space explosion
    - solution: abstraction techniques (bisimulation minimization, ...)
    - desirable: compositionality

Outline

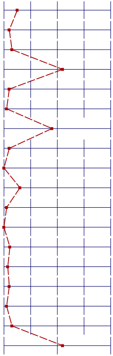
Courses and Seminars in Summer 2009

- Course Advanced Model Checking [Katoen]
- Course Modeling Concurrent and Probabilistic Systems [Katoen/Noll] ("Hiwi" jobs available!)
- Course Testing of Reactive Systems [Bohnenkamp]
- Seminar Applying Formal Verification Methods to Embedded Systems [Noll/Schlich]

Outline

Profillinie

Teilbereich: Informatik
Name der/des Lehrenden: Priv.doz. Dr.rer.nat. Thomas Noll
Titel der Lehrveranstaltung: (Name der Umfrage) Semantik und Verifikation von Software (1033682)



I realise what the lecture is good for.	fully applies	does not apply	mw=1.5
The lecture is clearly structured.	fully applies	does not apply	mw=1.2
The lecture can be followed-up well with the material (script, textbook, handouts, ...) available.	fully applies	does not apply	mw=1.3
I have prior knowledge for this lecture.	fully applies	does not apply	mw=3.2
The examples chosen facilitate understanding the lecture's contents.	fully applies	does not apply	mw=1.2
The lecturer sums up the contents in appropriate intervals.	fully applies	does not apply	mw=1.1
The level of difficulty is ...	too high	too low	mw=1.9
... imparts the contents in an intelligible manner.	fully applies	does not apply	mw=1.2
... responds with great care to questions of understanding.	fully applies	does not apply	mw=1
... considers the different levels of knowledge of the students.	fully applies	does not apply	mw=1.6
... speaks loudly and clearly in an appropriate manner.	fully applies	does not apply	mw=1.1
... speaks proper, comprehensible English.	fully applies	does not apply	mw=1
... is open to improvement suggestions.	fully applies	does not apply	mw=1.2
... is available outside lecture times, e.g. during business hours or by email.	fully applies	does not apply	mw=1.1
The employment of auxiliary materials (blackboard, overhead projector, projector, demonstrations, ...) is appropriate.	fully applies	does not apply	mw=1.2
Handwriting and drawings are legible.	fully applies	does not apply	mw=1.1
Blackboard texts and transparencies are clearly arranged.	fully applies	does not apply	mw=1.3
The pace is ...	too fast	too slow	mw=2.1

Auswertungsteil der offenen Fragen

In your opinion, what makes the lecture especially bad or good? How can the lecture be improved (presentation, media, equipment, …)? Please note that your handwritten comments may possibly lead back to you. We therefore suggest that you make your handwritten comments in block letters. Comments made outside the text box will not ...

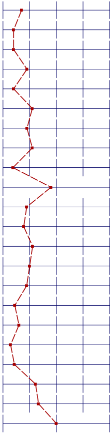
Slides shoud have more details about the topic. Current slides are very dense. There should be projectors fixed in lecture rooms. Every teacher has to bring his own projector that is not good.

Mr. Noll was always well prepared for the lectures and presented the contents in a very structured way, which I enjoyed much.
The slides and the handouts could be more detailed (sometimes)

Profillinie

Teilbereich: Informatik
Name der/des Lehrenden: Priv.doz. Dr.rer.nat. Thomas Noll
Titel der Lehrveranstaltung: (Name der Umfrage) Semantik und Verifikation von Software (1033382)

Lecture and exercise harmonise with regard to contents.	fully applies	does not apply	mw=1.7
Lecture and exercise harmonise with regard to time planning.	fully applies	does not apply	mw=1.4
I realise what the exercise course is good for.	fully applies	does not apply	mw=1.4
The process of the exercise course is well-structured.	fully applies	does not apply	mw=1.9
The exercises chosen facilitate understanding the course content.	fully applies	does not apply	mw=1.4
The exercise tasks are comprehensible.	fully applies	does not apply	mw=2.1
The exercise tasks have a reasonable scope.	fully applies	does not apply	mw=1.9
The solutions presented are comprehensible.	fully applies	does not apply	mw=2.1
In case you could deliver your solution: was it controlled in an appropriate manner?	fully applies	does not apply	mw=1.4
The level of difficulty is ...	too high	too low	mw=1.9
... imparts the contents in an intelligible manner.	fully applies	does not apply	mw=1.9
... responds with great care the questions of understanding.	fully applies	does not apply	mw=1.8
... considers the different levels of knowledge of the students.	fully applies	does not apply	mw=2.1
... speaks proper, comprehensible English.	fully applies	does not apply	mw=2
... speaks loudly and clearly in an appropriate manner.	fully applies	does not apply	mw=1.9
... is open to improvement suggestions.	fully applies	does not apply	mw=1.4
... prepared for this exercise course adequate.	fully applies	does not apply	mw=1.6
... is available outside exercise course times, e.g. during business hours or by email.	fully applies	does not apply	mw=1.3
The employment of auxiliary materials (blackboard, overhead projector, projector, ...) is appropriate.	fully applies	does not apply	mw=1.4
Handwriting and drawings are legible.	fully applies	does not apply	mw=2.2
Blackboard texts and transparencies are clearly arranged.	fully applies	does not apply	mw=2.3
The pace is ...	too fast	too slow	mw=2



Auswertungsteil der offenen Fragen

In your opinion, what makes this exercise course especially bad or good? How can this exercise course be improved (presentation, media, equipment, …)? Please note that your handwritten comments may possibly lead back to you. We therefore suggest that you make your handwritten comments in block letters. Comments made outside the te...

More detailled exercices (description, and some applications for such theory..)

Ting Ting was well prepared for the exercise courses and presented the solutions very detailed.
Critics on Alexandru: You have to speak up and try better on your handwriting. Thanks for ~~sorting~~ putting the solutions on the website!