Declarative modeling

Dirk Fahland

based on slides by Christian Stahl
Programming paradigms

**Imperative**: How to compute/process/…?

**Declarative**: What should program accomplish?

- Includes computation paradigms

- Simplify implementation of parallel programs
Imperative modeling

- Explicit representation of steps in the process
- What is **allowed**?

\[ E = mc^2 \]
Example: ambulance

1. meet doctor
2. take photo
3. meet doctor again
4. do operation
5. get cast
When the imperative approach fails

- 20% standard, 80% exceptions

  ➔ Simpler to say what is not allowed

- High degree of variability
- Set of tasks not well-defined

  ➔ Hospital systems, case handling systems
Most natural choice regarding lucidity and modeling effort?
Declarative modeling: The idea

• Specify the **rules** of the process

• Flower model + additional (temporal) constraints
Example revised: informal

Preceded by meeting doctor

No operation has been performed

Not eaten, no cast, photo taken

Preceded by meeting doctor

After taking photo(s), meet the doctor again

No operation has been performed

Not eaten, no cast, photo taken

Preceded by meeting doctor
Toward a declarative model

Tasks

- meet doctor
- take a photo
- do operation
- get a cast
- eat

Constraints

- Before a photo is taken, the doctor has to be met
- After taking photo(s), meet the doctor again
- Perform an operation only if the patient has not eaten, is not in a cast, and at least one photo has been taken
- Eat only if no operation has been performed
- Before a cast is put on, the doctor has to be met
Specifying constraints

- Formal semantics (e.g., LTL)
- Graphical notation

Expressiveness

- As expressive as workflow nets (or business process models in Protos)
Outline

• Part I: The declarative modeling paradigm

• Part II: Declarative modeling
  • Introduction to a concrete syntax
  • Assignment
  • Evaluation

• Part III: Closing
Existence constraints

**Init**

start activity, for all instances

**Last**

final activity, for all instances
Order constraints

Response(A, B)

Whenever A is executed, B has to be eventually executed afterward.

Examples: CDACAB ; CDC ; CDADD

Precedence(A, B)

B has to be preceded by A.

Examples: CDA ; CAACBB ; CCB
Alternate Order constraints

**AlternateResponse**(A, B)

After each A is executed at least one B is executed.

Examples: CDACBBBAB ; CADABD

**AlternatePrecedence**(A, B)

B cannot happen before A. After it happens, it can not happen before the next A again.

Examples: CADABD ; CCB ; CABB
Choice Constraints

ExclusiveChoice(A, B)

Either A or B must be executed.
Example: CAD ; CBC ; CCD ; CAB
Negative relation constraints

**NotSuccession(A, B)**

Before B there cannot be A and after A there cannot be B.

Examples: CBA ; ABD

**NotCoexistence(A, B)**

Either A or B can happen.

Examples: CDA ; CBDB ; CDE ; CCBA
Tool support: Declare
Example revised: Tasks

Tasks

- meet doctor
- take a photo
- do operation
- get a cast
- eat
Before a photo is taken, the doctor has to be met.
After taking photo(s), meet the doctor again
Example revised: Add Constraints 3 and 4

Eat only if no operation has been performed

Perform an operation only if the patient has not eaten, is not in a cast, and at least one photo has been taken
Before a cast is put on, the doctor has to be met.
Determining the initial state

No init

- get a cast
- meet doctor
- do operation
- take a photo
- eat

Diagram shows relationships between these states with arrows indicating precedence and relationships such as not succession and not co-existence.
Executing the model (1)
Executing the model (2)

meet doctor

get a cast

meet doctor

response

take a photo

do operation

not succession

not co-existence

eat
Executing the model (3)

meet doctor

take a photo

got a cast

meet doctor

precedence

response

take a photo

not succession

not co-existence

do operation

eat
Executing the model (4)

- meet doctor
- take a photo
- do operation

Diagram:

- meet doctor -> precedence -> get a cast -> not succession -> not co-existence -> eat
- meet doctor -> precedence -> take a photo
- take a photo -> precedence -> do operation
- do operation -> precedence -> meet doctor
Executing the model (5)

- meet doctor
- take a photo
- do operation
- get a cast

Diagram:
- Box labeled "get a cast" with a green checkmark.
- Arrows indicating precedence from "meet doctor" to "take a photo" and from "take a photo" to "do operation".
- Arrows indicating "not succession" from "get a cast" to "do operation".
- Arrows indicating "not co-existence" from "do operation" to "eat".
Executing the model (6)

- meet doctor
- take a photo
- do operation
- get a cast
- get a cast
Executing the model (7)

- meet doctor
- take a photo
- do operation
- get a cast
- get a cast
- meet doctor

- get a cast (precedence)
- meet doctor (response)
- take a photo (precedence)
- do operation (not succession)
- eat (not co-existence)
executing the model (8)

- meet doctor
- take a photo
- do operation
- get a cast
- meet doctor
- get a cast

Diagram:
- get a cast (precedence)
- do operation (not succession)
- eat (not co-existence)
- take a photo (precedence)
- meet doctor (precedence)
Outline

• Part I: The declarative modeling paradigm
• Part II: Declarative modeling
  • Introduction to a concrete syntax
  • Assignment
  • Evaluation
• Part III: Closing
Open issues

• When is the declarative approach beneficial?

• How to use declarative modeling effectively?
  • How to specify constraints?
  • Which constraints are needed?
  • How to model in a declarative way?

• What are the benefits?
Future work

• Investigate how declarative modeling can be used

• Consider link with Perceptive models

• Expand the scope of declarative modeling
Vision: Hybrid approach

- What **must** happen?
- What **may** happen?
- What **must not** happen?
Take home points