Three studies on Model Transformations

- Parsing, Generation and Ease of Use

Håkan Burden

Model

- Description
- Abstraction
- Usable

Ceci n'est pas une pipe.

- C code: 1001110…
- Grammar: ”one bottle of milk,
  two bottles of milk, …”

Model Transformations

Source → Target

- Rules
- Algorithm
- Manual … Automatic
- Endogenuous or Exogenuous
- Abstraction level

Examples

		Level of abstraction	
	S<T	S=T	S>T
Endo-genuous	Abstracts in publications	Re-factoring/ re-ordering	Optimisations **Parsing**
Exo-genuous	Subtitles **Generation**	Google Translate	Code generation **Ease of Use**

Parsing Linear Context-Free Rewriting Systems

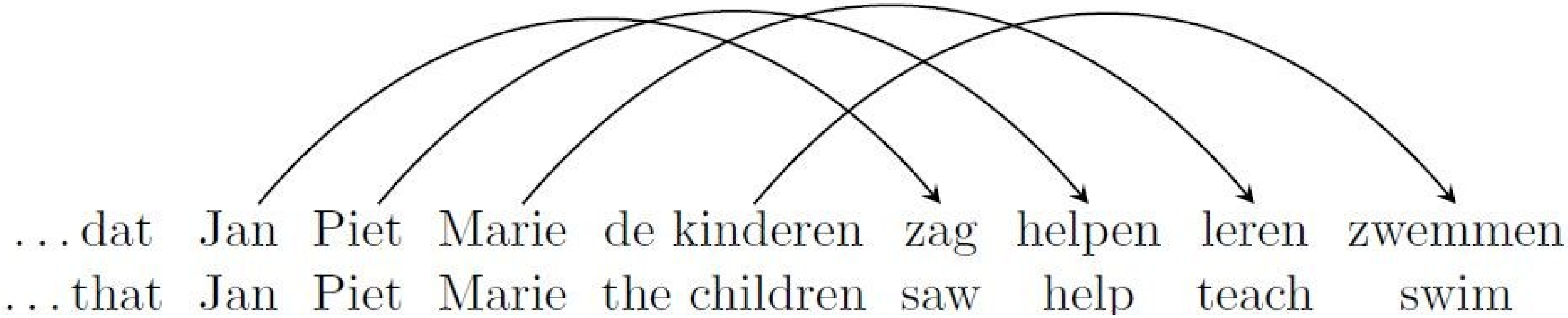
Håkan Burden and Peter Ljunglöf

IWPT´05

International Workshop on Parsing Technologies

2005

Dutch Subordinate Clauses



…dat	Jan	Piet	Marie	de kinderen	zag	helpen	leren	zwemmen
…that	Jan	Piet	Marie	the children	saw	help	teach	swim

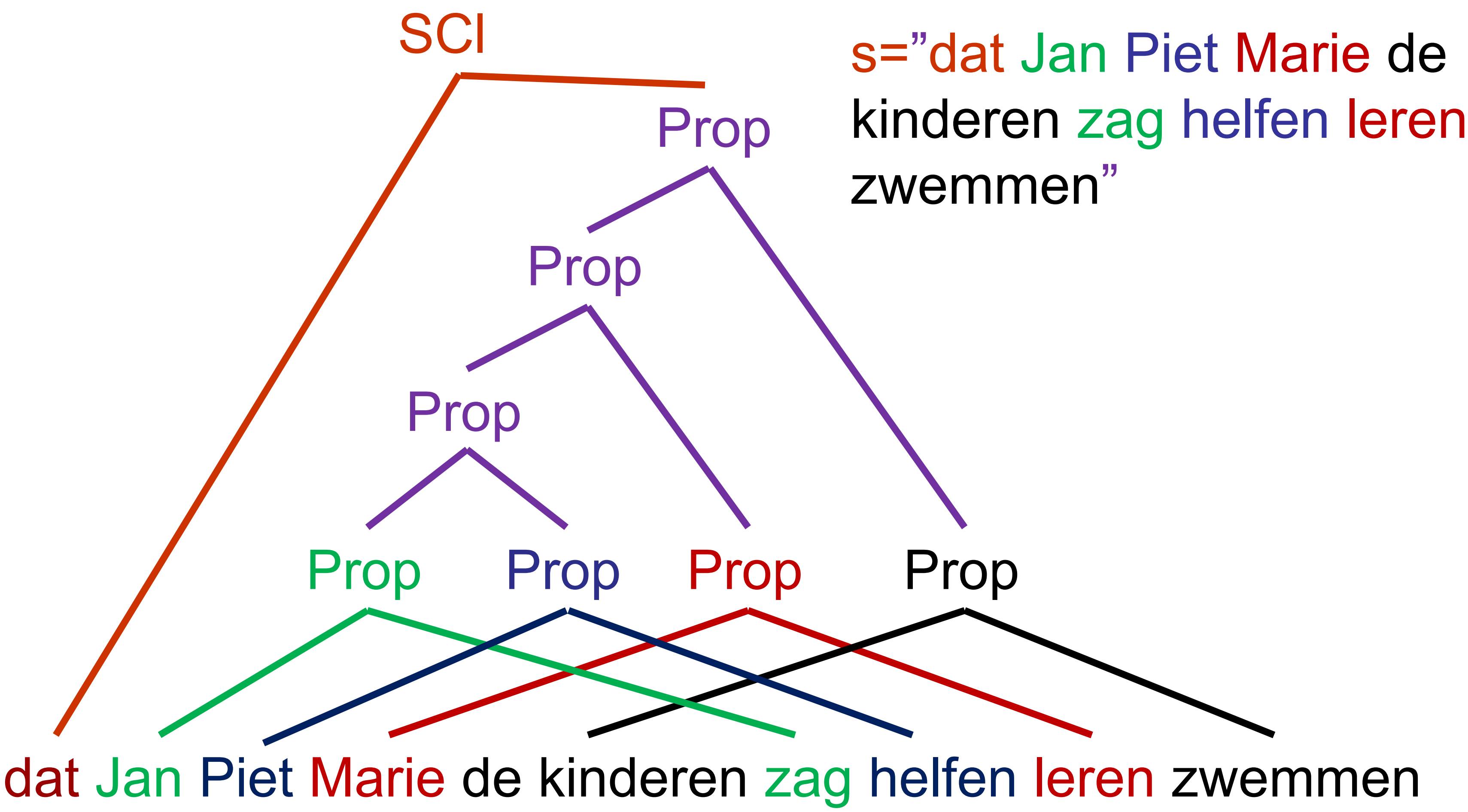
A Grammar

- Prop → jz[ ] := { subj=”Jan” ; pred=”zag” }
- Prop → ph[ ] := { subj=”Piet” ; pred=”helfen” }
- Prop → ml[ ] := { subj=”Marie” ; pred=”leren” }
- Prop → kz[ ] := { subj=”de kinderen” ; pred=”zwemmen” }

- Prop → rec[Prop$_1$, Prop$_2$] := { subj=Prop$_1$.subj Prop$_2$.subj;
pred=Prop$_1$.pred Prop$_2$.pred }

- SCl → subcl[Prop] := { cl=”dat” Prop.subj Prop.pred }

Bottom-up Parsing



s=”dat Jan Piet Marie de kinderen zag helfen leren zwemmen”

Natural Language Generation from Class Diagrams

Håkan Burden and Rogardt Heldal

MoDeVVa'11

8th International Workshop on Model-Driven Engineering, Verification and Validation

2011

Background

Natural Language Generation:

- Text planning
- Sentence planning
- Linguistic realisation

Class Diagram:

- Software Model

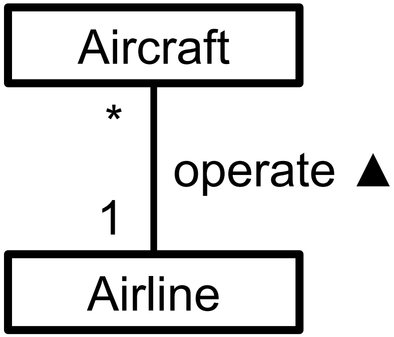
Two Models

Grammatical Framework:

```
cat Concept, Prop
fun Aircraft, Airline : Concept
      Statement : Concept × Concept→Prop

lincat Concept = N, Prop = S
lin      Airline = mkN ”airline”
         Aircraft = mkN ”aircraft” ”aircraft”
         Statement = mkS (mkCl (mkNP  a_Det  A    e)
                                (mkV2  ”oper
                                (mkNP  ma     et  Aircraft))
```

Class Diagram:



”an airline operates many aircraft”

Executable and Translatable UML – How Difficult Can it Be?

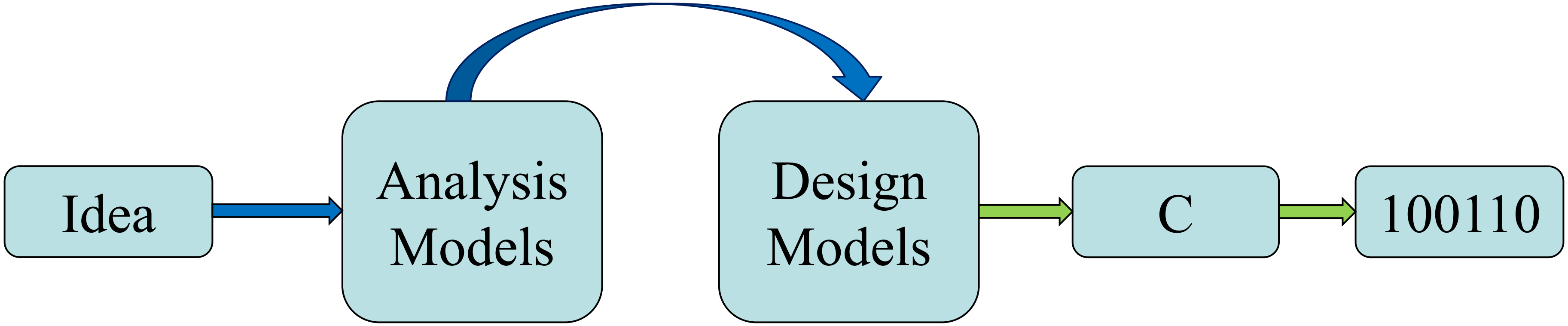
Håkan Burden, Rogardt Heldal and Toni Siljamäki

APSEC'11

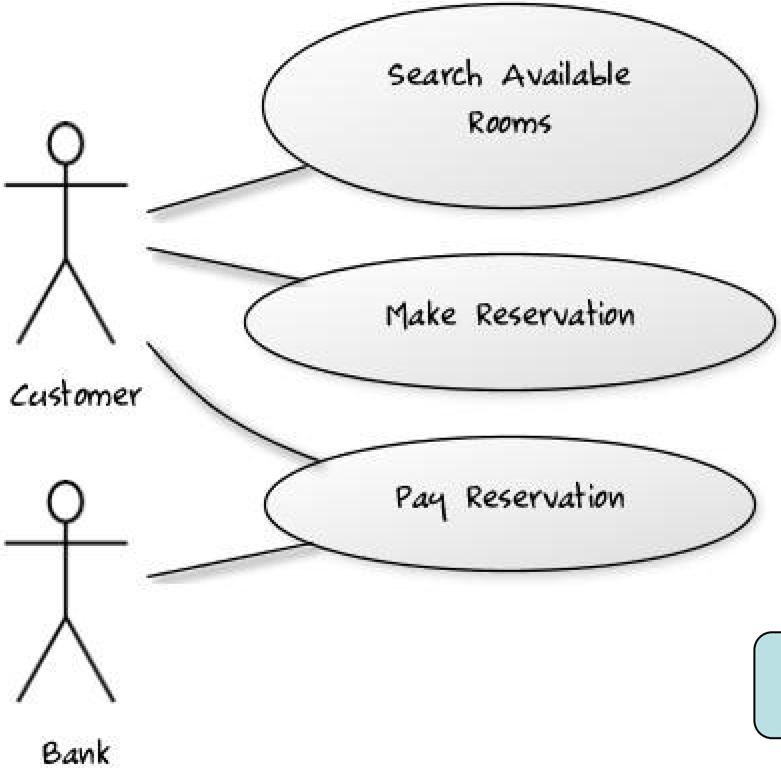
18th Asia-Pacific Software Engineering Conference

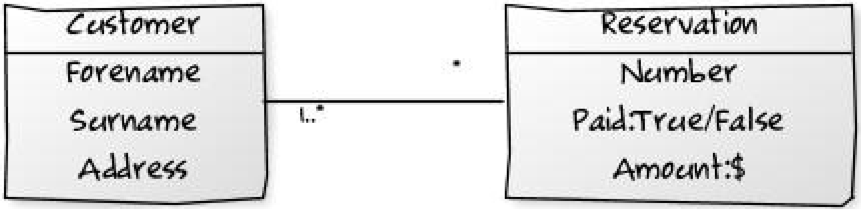
2011

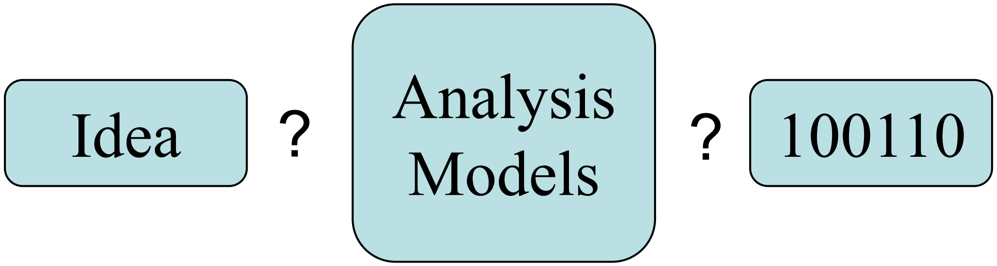
From Idea to Software



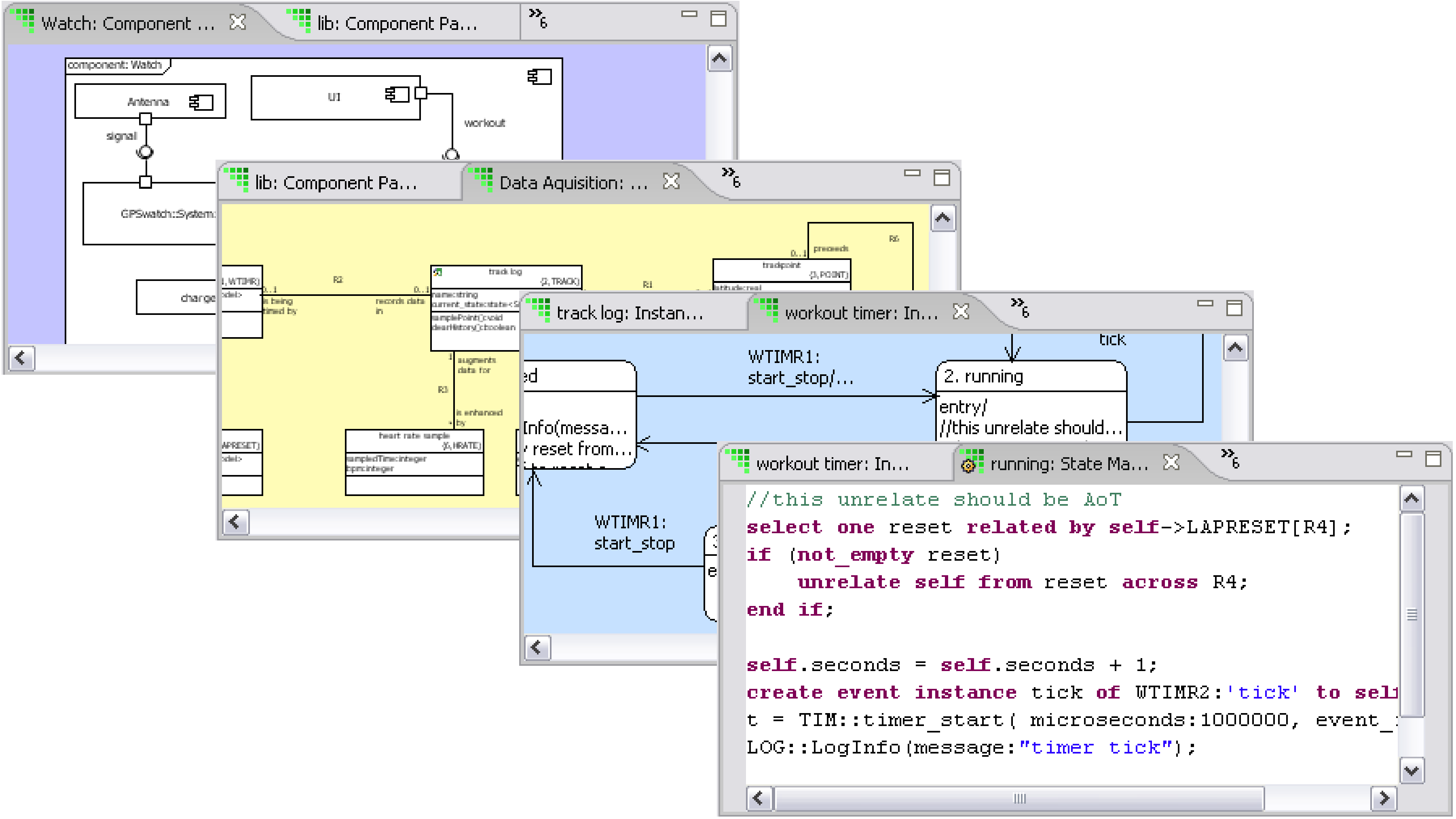
Analysis Models






Idea ? Analysis Models ? 100110

Design Models



```
//this unrelate should be AoT
select one reset related by self->LAPRESET[R4];
if (not_empty reset)
    unrelate self from reset across R4;
end if;

self.seconds = self.seconds + 1;
create event instance tick of WTIMR2:'tick' to sel
t = TIM::timer_start( microseconds:1000000, event_
LOG::LogInfo(message:"timer tick");
```

Outcome

Can teams of four students solve a problem that is complex enough to require using the full potential of the design models within a total of 300 hours?

	2009	2010	2011	Total
#Teams	22	28	24	74
#Success	18	25	23	66
#Failure	4	3	1	8

Conclusion

Parsing:

- Endogenuous
- Refinement

Ease of Use:

- Exogenuous
- Synthesis

Generation:

- Exogenuous
- Reverse engineering

Future Work

- Further explore generation from SW models
  - Action language to natural language

- Models and transformations in industry
  - Transformations as re-usable assets
  - Modelling cross paradigms

- Educational implications