Computational models for contracts

John J. Camilleri

UiO talk, 2014-05-13

CHALMERS | UNIVERSITY OF GOTHENBURG

About me

- No background in law
- PhD student in Computer Science
  at Chalmers / University of Gothenburg
- Formal methods & language technology

REMU

Reliable **Mu**ltilingual Digital Communication: Methods and Applications

http://remu.grammaticalframework.org/

REMU

REMU

1. Hybrid machine translation
    - Grammars + statistics
2. Formal methods for grammars
    - Ambiguity detection
3. Reasoning

What is a contract?

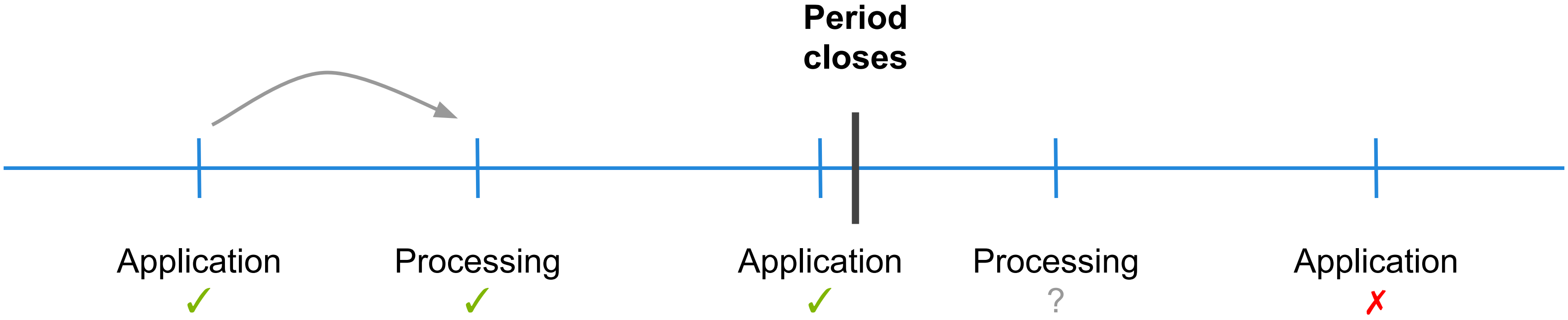
- **Document containing norms** prescribing procedures, behaviours, rights
- Examples
  - requirement specification
  - privacy policy
  - terms of service
  - service-level agreement

Motivating example

Application procedure

1. Applications may be submitted between 1st–31st May.
2. The secretary must process each application within 5 days.
3. The secretary should not process any applications after the period has closed.

Motivating example



Potential conflicts

- Is there something wrong with this contract?
- Should it be changed?

- That’s for a human to decide
- Computer: highlight potential conflicts for us

Desired tasks: static

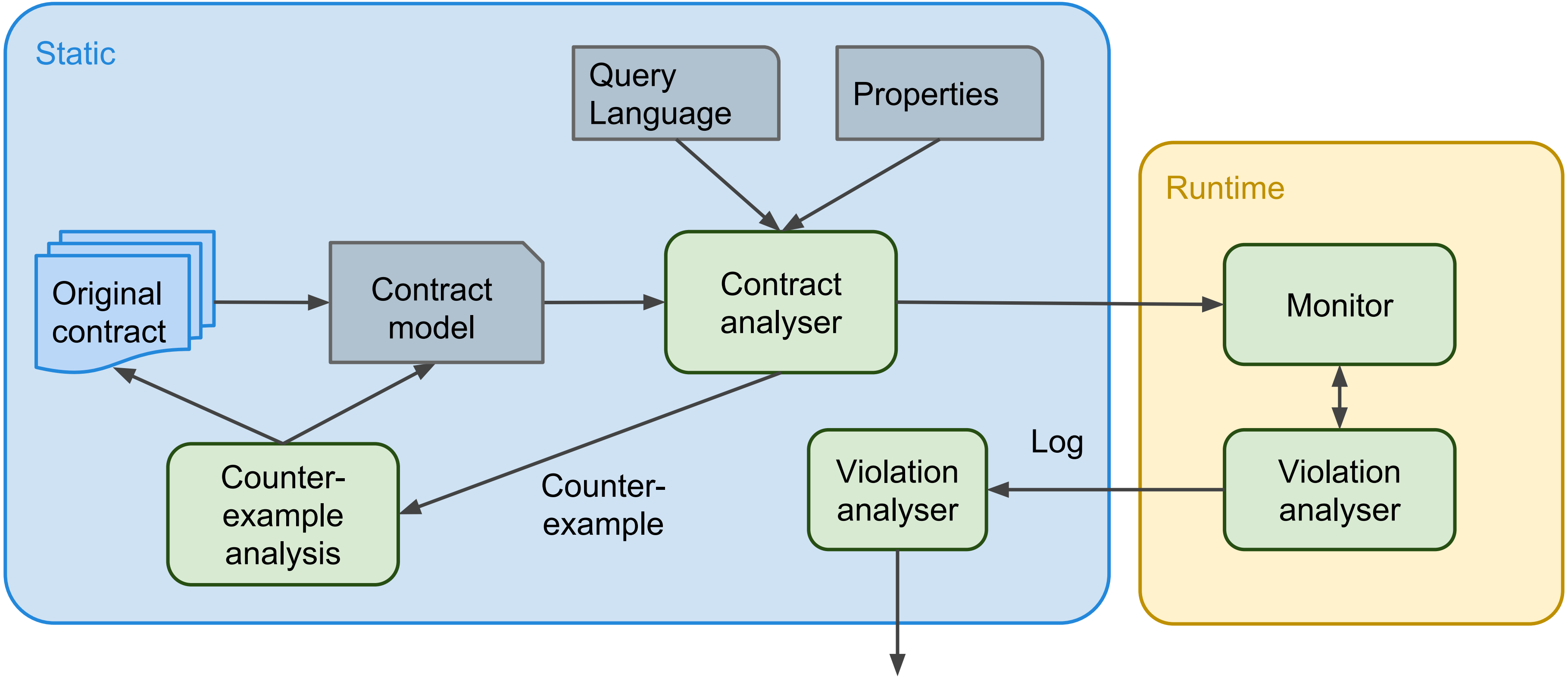
- Detect conflicts
    - While writing (author)
    - Before accepting (party)
- Query
    - Am I allowed to …?
    - What happens if …?
- Simulate
    - Discover undesirable possibilities, loopholes

Desired tasks: runtime

- Detect violations
- Enact reparations
- Logs, without interference

- Only **computer-mediated transactions**

The dream


Static
Query Language
Properties
Original contract
Contract model
Contract analyser
Counter-example analysis
Counter-example
Violation analyser
Log
Runtime
Monitor
Violation analyser

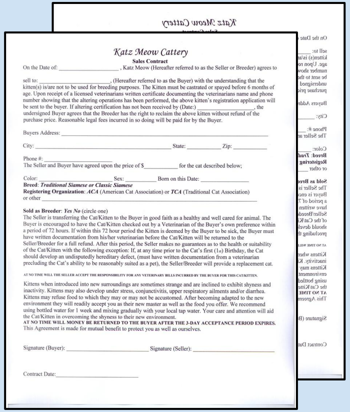
E-contracts and models

- Computers cannot process raw text
- Use formal **model** to represent our real-world contract

Originals & models

Natural Language

Original contract



Katz Meow Cattery

Sales Contract

On the Date of: ____________________ , Katz Meow (Hereafter referred to as the Seller or Breeder) agrees to

sell to: ________________________, (Hereafter referred to as the Buyer) with the understanding that the kitten(s) is/are not to be used for breeding purposes. The Kitten must be castrated or spayed before 6 months of age. Upon receipt of a licensed veterinarians written certificate documenting the veterinarians name and phone number showing that the altering operations has been performed, the above kitten's registration application will be sent to the buyer. If altering certification has not been received by (Date:) ___________________, the undersigned Buyer agrees that the Breeder has the right to reclaim the above kitten without refund of the purchase price. Reasonable legal fees incurred in so doing will be paid for by the Buyer.

Buyers Address: ____

City: ____ State: ___________ Zip: ___________

Phone #: _______________________
The Seller and Buyer have agreed upon the price of $___________ for the cat described below;

Color: _________________________ Sex: __________ Born on this Date: ________________
Breed: ***Traditional Siamese or Classic Siamese***
Registering Organization: ***ACA*** (American Cat Association) or ***TCA*** (Traditional Cat Association)
or other ____

Sold as Breeder: ***Yes No*** (circle one)
The Seller is transferring the Cat/Kitten to the Buyer in good faith as a healthy and well cared for animal. The Buyer is encouraged to have the Cat/Kitten checked out by a Veterinarian of the Buyer's own preference within a period of 72 hours. If within this 72 hour period the Kitten is deemed by the Buyer to be sick, the Buyer must have written documentation from his/her veterinarian before the Cat/Kitten will be returned to the Seller/Breeder for a full refund. After this period, the Seller makes no guarantees as to the health or suitability of the Cat/Kitten with the following exception: If, at any time prior to the Cat's first (1st) Birthday, the Cat should develop an undisputedly hereditary defect, (must have written documentation from a veterinarian precluding the Cat's ability to be reasonably suited as a pet), the Seller/Breeder will provide a replacement cat.

AT NO TIME WILL THE SELLER ACCEPT THE RESPONSIBILITY FOR ANY VETERINARY BILLS INCURRED BY THE BUYER FOR THIS CAT/KITTEN.

Kittens when introduced into new surroundings are sometimes strange and are inclined to exhibit shyness and inactivity. Kittens may also develop under stress, conjunctivitis, upper respiratory ailments and/or diarrhea. Kittens may refuse food to which they may or may not be accustomed. After becoming adapted to the new environment they will readily accept you as their new master as well as the food you offer. We recommend using bottled water for 1 week and mixing gradually with your local tap water. Your care and attention will aid the Cat/Kitten in overcoming the shyness to their new environment.
AT NO TIME WILL MONEY BE RETURNED TO THE BUYER AFTER THE 3-DAY ACCEPTANCE PERIOD EXPIRES.
This Agreement is made for mutual benefit to protect you as well as ourselves.

Signature (Buyer): ________________________ Signature (Seller): ________________________

Contract Date:________________________

Formalism

Model

$$P(su) \wedge [cl]F_{O(x)}(pr.co) \oplus O_{\perp}(xy \wedge (w \vee z))$$

```
<contract>
 <clauses>
 <clause>[a3](P(a1))</clause>
 <clause>(P(a2.a3))</clause>
 </clauses>
 <concurrentActions>
  <action>a3#a4</action>
 </concurrentActions>
</contract>
```

The formalism: Contract Logic (CL)

- A formal language for specifying “contracts”
- Action-based
  - Simple and complex actions
  - and, exclusive or, sequence
- Main modalities
  - Obligation
  - Permission
  - Prohibition
- Reparations

Examples of CL (1)

Student is **allowed** to submit an application

P(submit)

Secretary **must** process application and then send the reply

O(process . reply)

Examples of CL (2)

1. Student is allowed to submit an application
2. If the period is closed, the secretary should not process any application

$$P(\text{submit}) \wedge [\text{close}]F(\text{process})$$

Conflicts

- What is a conflict
  - O(a) and F(a)
  - P(a) and F(a), etc.
- Only easy in tiny examples

$$P(\text{submit}) \wedge F(\text{submit \& process})$$

Conflicts: big example

- Quickly becomes difficult

$$[\mathrm{open}]P(\mathrm{submit}) \wedge [\mathrm{submit}]O(\mathrm{process}) \\ \wedge [\mathrm{close}]F(\mathrm{process})$$

- CL Analyser “CLAN”
- Finds conflicts, gives counter-example

Counter-example

- Trace of actions which lead to conflict

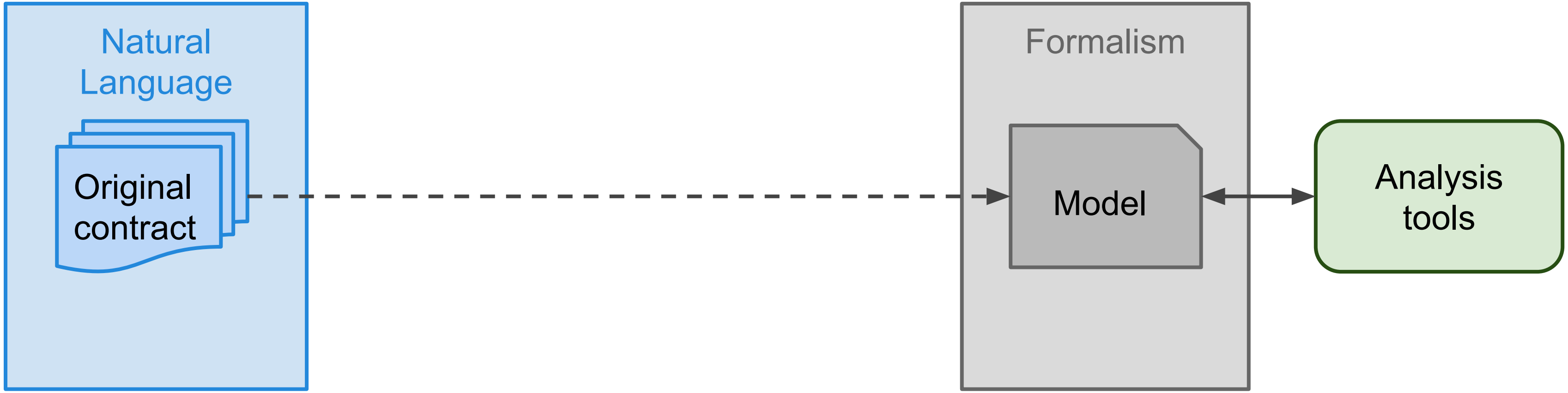
1. open
2. submit
3. close & submit

- May not be minimal!

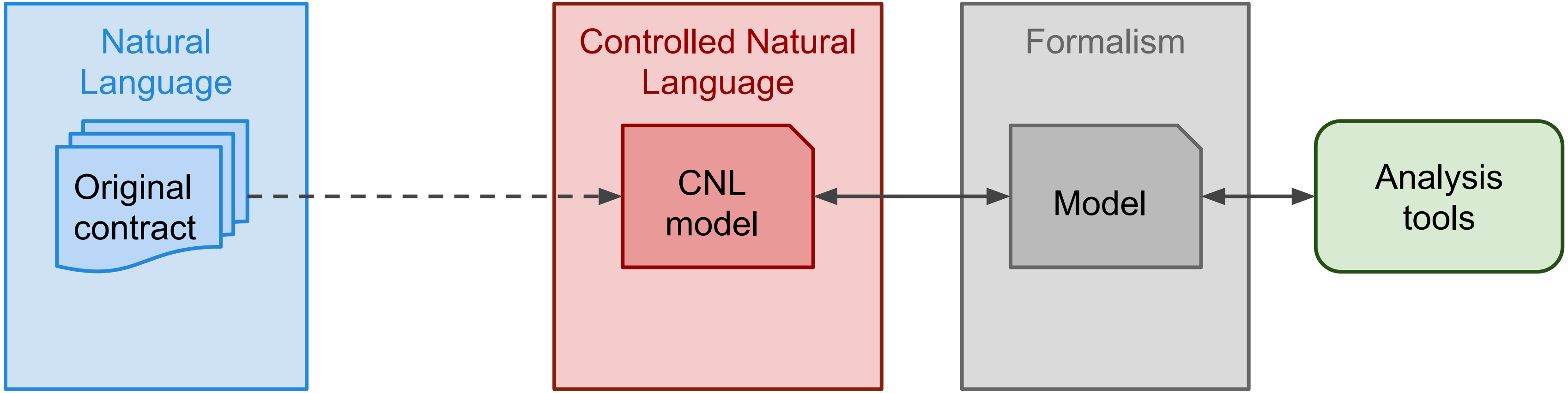
What could be simpler?

- Nearly impossible to do manually (without making mistakes)
  - Writing long formulas (model)
  - Analysing counter-examples
- Goal
  - Bridge the gap between **formal model** and **natural language**

The divide


Natural Language
Original contract
Formalism
Model
Analysis tools

The divide


Natural Language
Original contract
Controlled Natural Language
CNL model
Formalism
Model
Analysis tools

Controlled Natural Language

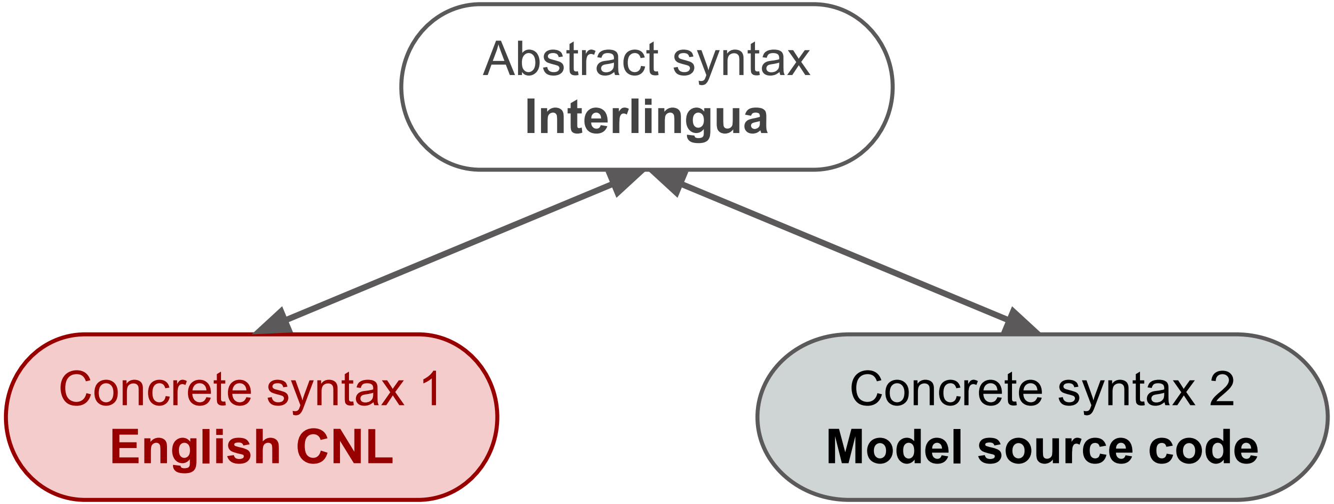
- Simplified version of NL
- Restricted syntax, vocabulary
- Formal language (described precisely by grammar)
- Parse-able
- Unambiguous (ideally)
- Still human-friendly

Grammatical Framework (GF)

- Functional programming language for multilingual grammars
- Language-independent semantic interlingua
- Generation and parsing from a single grammar

http://www.grammaticalframework.org

Grammatical Framework (GF)



Abstract syntax
Interlingua

Concrete syntax 1
English CNL

Concrete syntax 2
Model source code

- Same grammar for both directions
- Only one grammar per language (no pairs)

Live demo: AnaCon

- Workflow
    - Input
    - Result
    - Intermediate files

Limitations (1)

- Interface
  - Low-level (terminal)
  - No help in composing
- Formalism
  - No separation of agent
  - No subject / object
  - No concept of time (only sequence of actions)

Limitations (2)

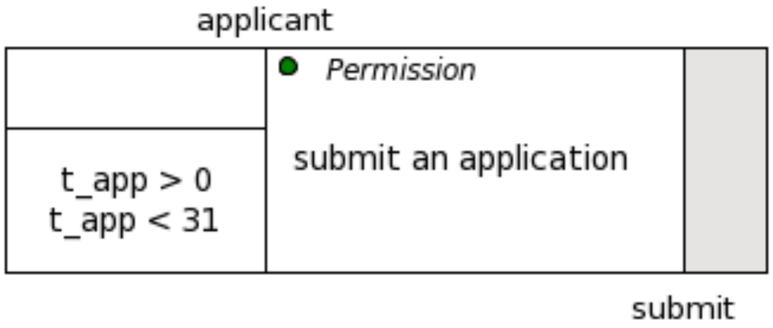
- Conflict detection
  - Counter-examples are not minimal
  - Analysis is a hard problem
- Querying?

↑ AnaCon

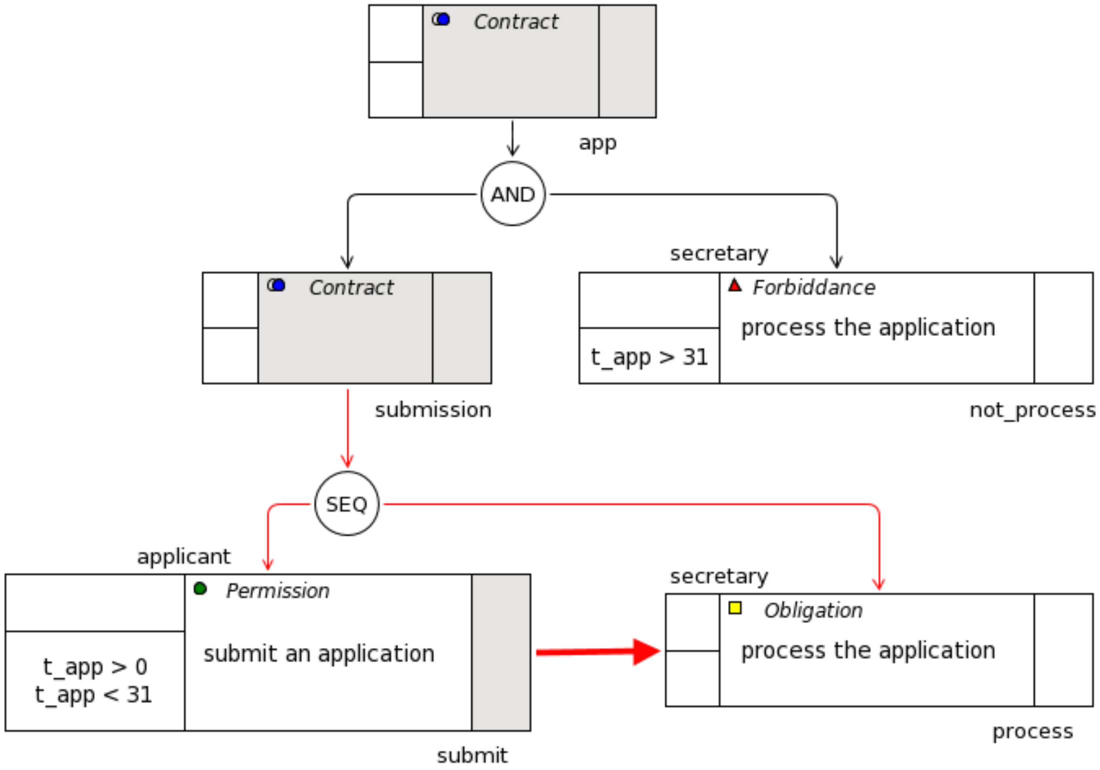
↓ Current & future work

Contract-Oriented diagrams

- Visual, modular
- Naming of clauses
- Separation of agent
- Real timing restrictions
- Guard conditions
- Cross-references



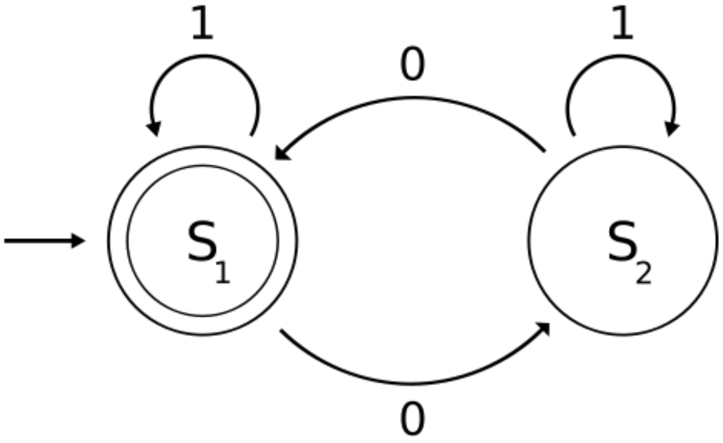
Running example



Intro to automata

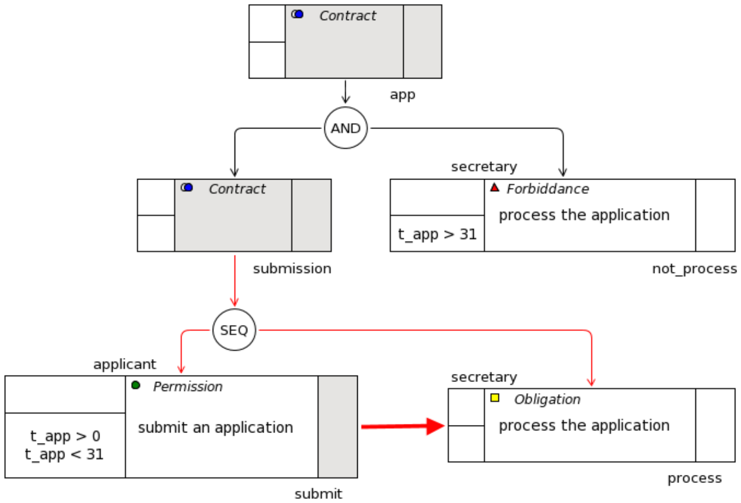
- States/nodes
- Edges/transitions

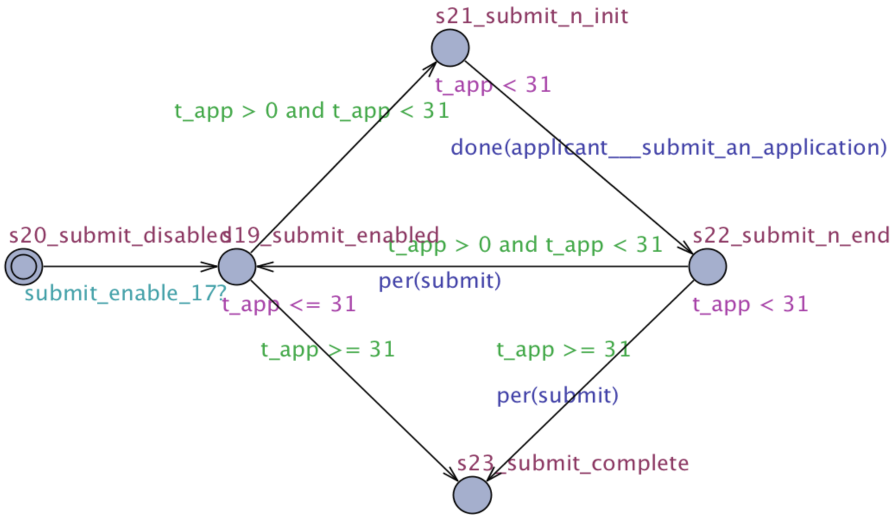
- Examples
  - 0110 ✓
  - 01100 ✗
  - Even number of 0's



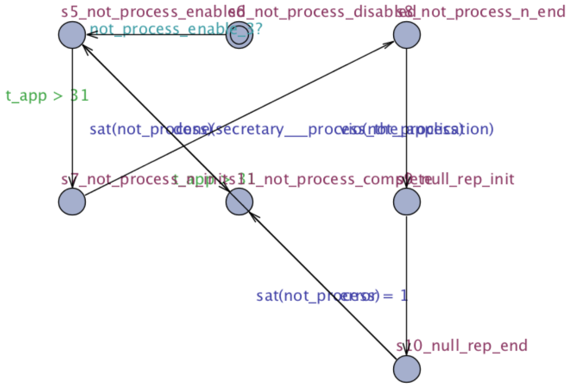
Conversion to NTA

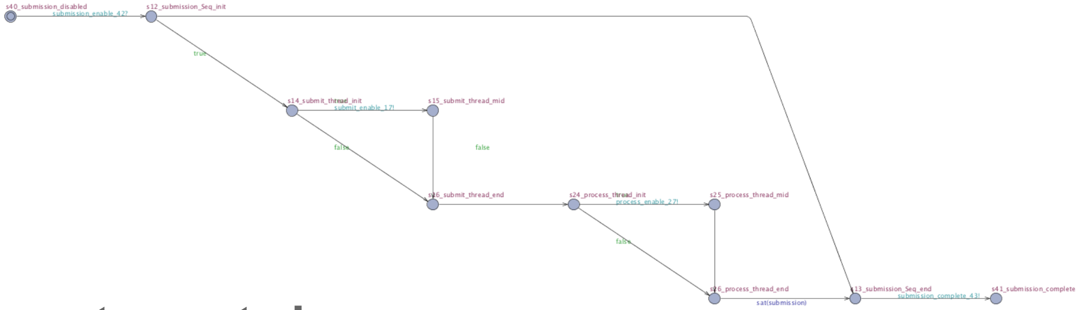
Translation function from diagrams into networks of **timed automata**

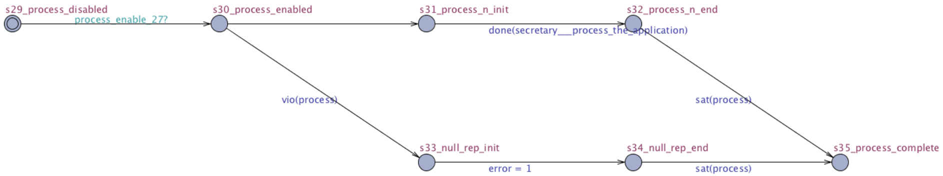


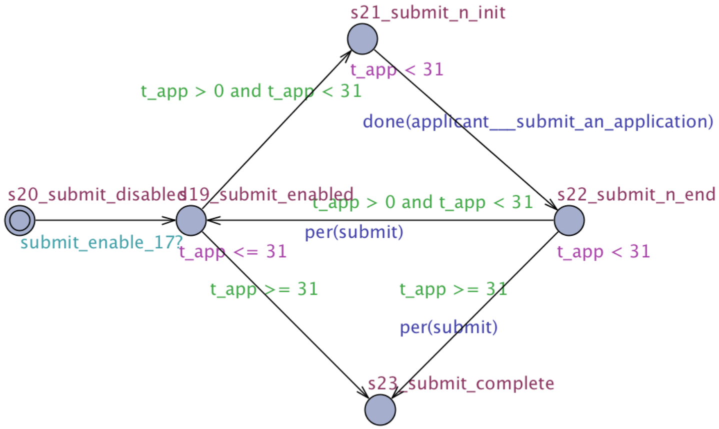


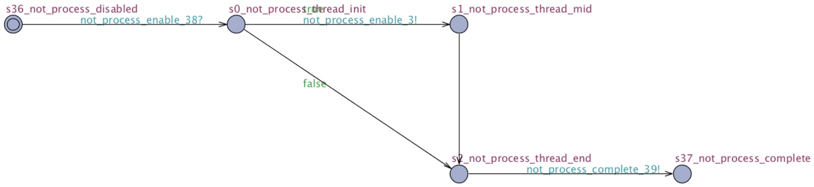
6 automata!

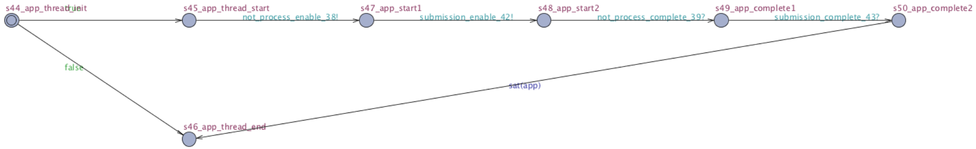

s5_not_process_enabled
s6_not_process_disabled
s8_not_process_n_end
not_process_enable_3?
t_app > 31
sat(not_process)
done(secretary___process_the_application)
vio(not_process)
s7_not_process_n_init
t_app <= 31
s11_not_process_complete
s9_null_rep_init
sat(not_process)
error = 1
s10_null_rep_end


s40_submission_disabled
submission_enable_42?
s12_submission_Seq_init
true
s14_submit_thread_init
true
submit_enable_17!
s15_submit_thread_mid
false
false
s16_submit_thread_end
s24_process_thread_init
true
process_enable_27!
s25_process_thread_mid
false
s26_process_thread_end
sat(submission)
s13_submission_Seq_end
submission_complete_43!
s41_submission_complete


s29_process_disabled
process_enable_27?
s30_process_enabled
s31_process_n_init
s32_process_n_end
done(secretary___process_the_application)
vio(process)
sat(process)
s33_null_rep_init
error = 1
s34_null_rep_end
sat(process)
s35_process_complete


s21_submit_n_init
t_app < 31
t_app > 0 and t_app < 31
done(applicant___submit_an_application)
s20_submit_disabled
s19_submit_enabled
t_app > 0 and t_app < 31
s22_submit_n_end
submit_enable_17?
per(submit)
t_app <= 31
t_app < 31
t_app >= 31
t_app >= 31
per(submit)
s23_submit_complete


s36_not_process_disabled
not_process_enable_38?
s0_not_process_thread_init
true
not_process_enable_3!
s1_not_process_thread_mid
false
s2_not_process_thread_end
not_process_complete_39!
s37_not_process_complete


s44_app_thread_init
true
s45_app_thread_start
not_process_enable_38!
s47_app_start1
submission_enable_42!
s48_app_start2
not_process_complete_39?
s49_app_complete1
submission_complete_43?
s50_app_complete2
false
sat(app)
s46_app_thread_end

Model checking

What can we do with automata?

- Simulation
- Model checking
  - Test properties in temporal logic
    “it is never the case that x and y”

    $$A \ \Box \ !\,(x \wedge y)$$
- Querying by model checking

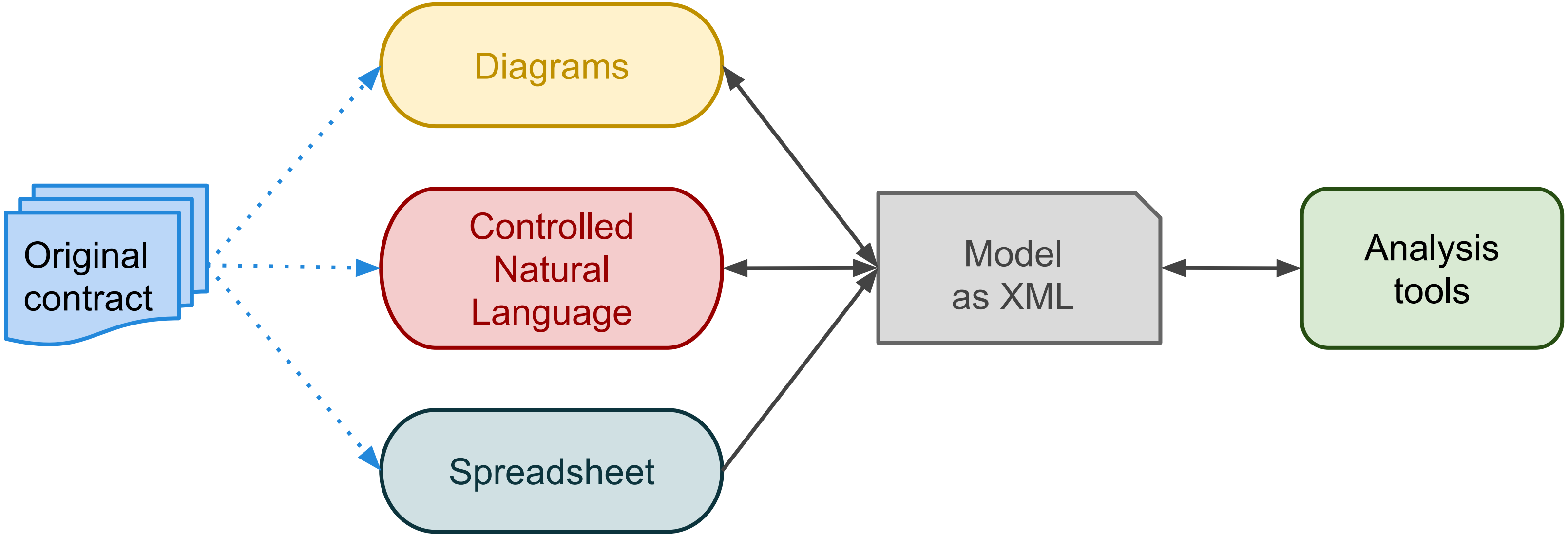
UPPAAL Demo

- Simulation
- Property checking

Lost in the details

- This gets pretty low level!
- We need to work at higher abstraction levels
- Separate tasks
  - Building models
  - Generating properties
  - Interpreting their result

One model, different views



Visual diagram editor

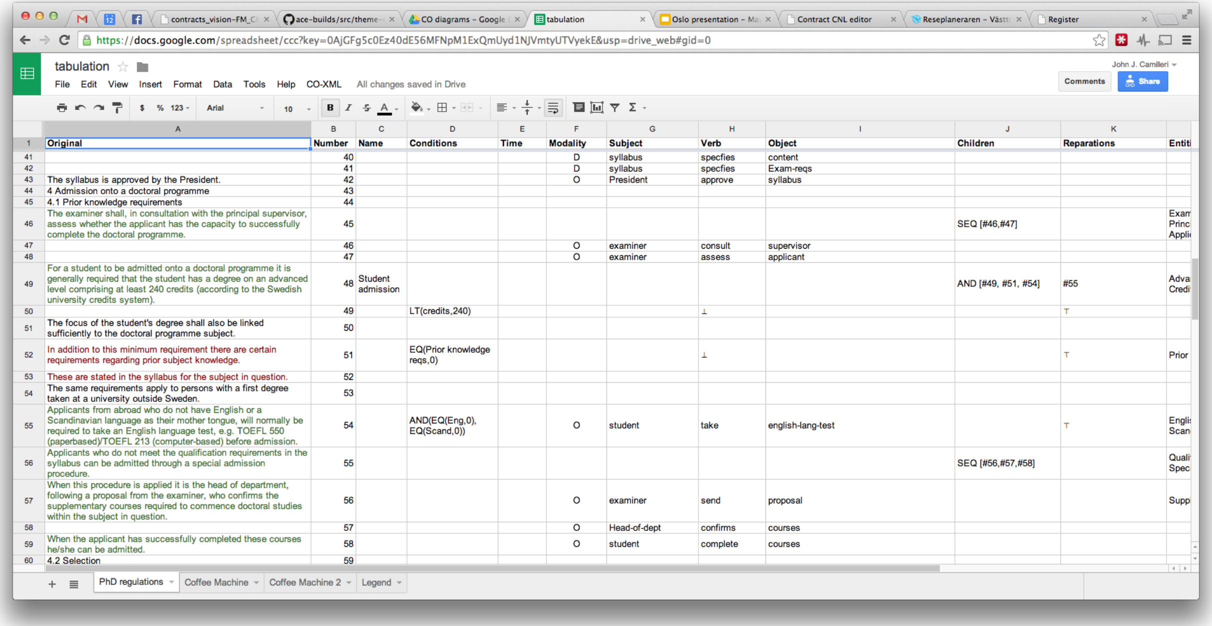
- Point and click diagram editor
- Web-based
- Automatic validation
- Export

(Another) CNL

- Similar to CNL in AnaCon
- More kinds of expressions
- Includes large-scale dictionary
- Focus on **tools**

Spreadsheet view

Starting place for auto-generated partial output

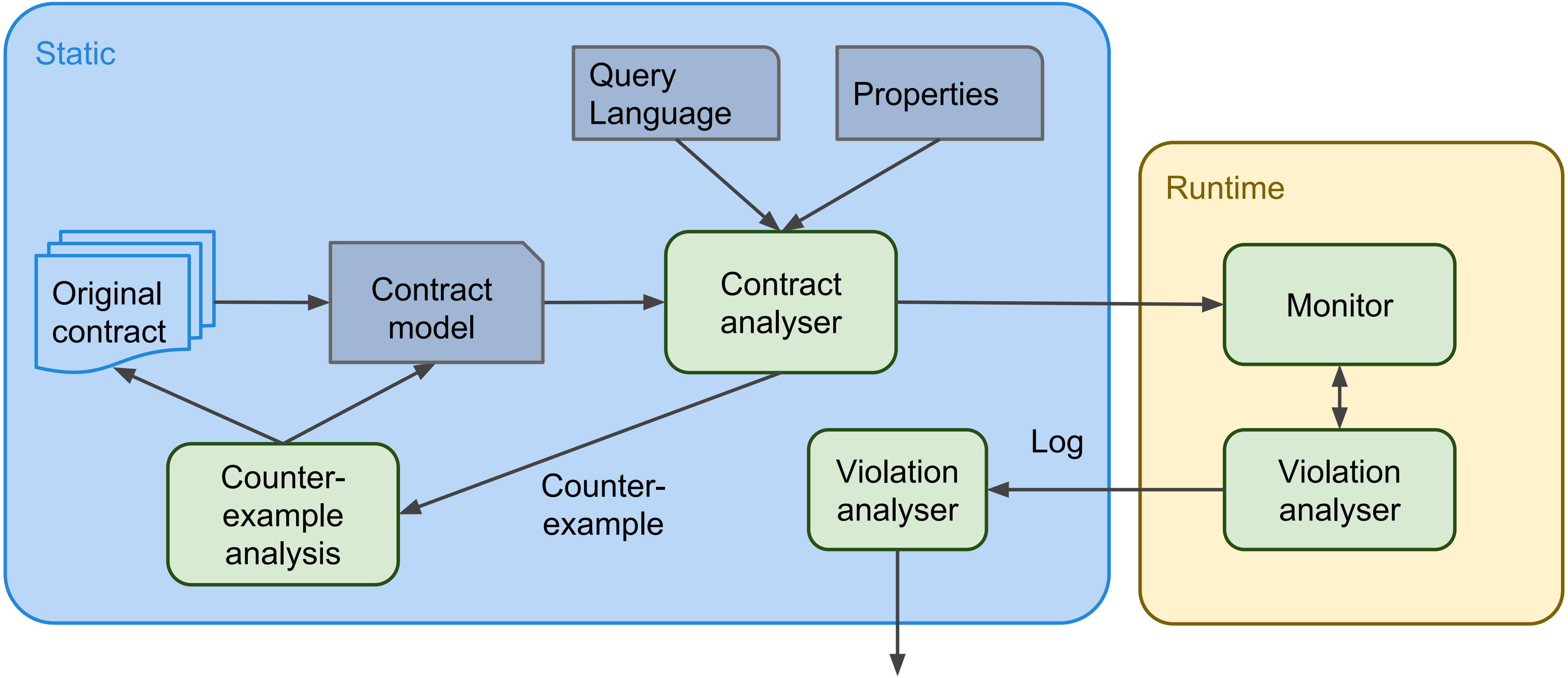


	A	B	C	D	E	F	G	H	I	J	K	
1	Original	Number	Name	Conditions	Time	Modality	Subject	Verb	Object	Children	Reparations	Entiti
41		40				D	syllabus	specfies	content			
42		41				D	syllabus	specfies	Exam-reqs			
43	The syllabus is approved by the President.	42				O	President	approve	syllabus			
44	4 Admission onto a doctoral programme	43										
45	4.1 Prior knowledge requirements	44										
46	The examiner shall, in consultation with the principal supervisor, assess whether the applicant has the capacity to successfully complete the doctoral programme.	45								SEQ [#46,#47]		Exam Princ Appli
47		46				O	examiner	consult	supervisor			
48		47				O	examiner	assess	applicant			
49	For a student to be admitted onto a doctoral programme it is generally required that the student has a degree on an advanced level comprising at least 240 credits (according to the Swedish university credits system).	48	Student admission							AND [#49, #51, #54]	#55	Adva Credi
50		49		LT(credits,240)				⊥			⊤	
51	The focus of the student's degree shall also be linked sufficiently to the doctoral programme subject.	50										
52	In addition to this minimum requirement there are certain requirements regarding prior subject knowledge.	51		EQ(Prior knowledge reqs,0)				⊥			⊤	Prior
53	These are stated in the syllabus for the subject in question.	52										
54	The same requirements apply to persons with a first degree taken at a university outside Sweden.	53										
55	Applicants from abroad who do not have English or a Scandinavian language as their mother tongue, will normally be required to take an English language test, e.g. TOEFL 550 (paperbased)/TOEFL 213 (computer-based) before admission.	54		AND(EQ(Eng,0), EQ(Scand,0))		O	student	take	english-lang-test		⊤	Englis Scan
56	Applicants who do not meet the qualification requirements in the syllabus can be admitted through a special admission procedure.	55								SEQ [#56,#57,#58]		Quali Speci
57	When this procedure is applied it is the head of department, following a proposal from the examiner, who confirms the supplementary courses required to commence doctoral studies within the subject in question.	56				O	examiner	send	proposal			Supp
58		57				O	Head-of-dept	confirms	courses			
59	When the applicant has successfully completed these courses he/she can be admitted.	58				O	student	complete	courses			
60	4.2 Selection	59										

Demo

- CNL editor
  - Example of CNL in text editor
  - Load in CNL editor
    - highlighting of labels, folding
    - snippets, autocompletion
    - validation
- Export to visual editor

The dream (revisited)



Problems: NLP

Natural language processing is hard

- CNL aims to bridge the gap
- Requires more time to sound less mechanical
- More work on editing tools

Problems: NLP (another solution)

- Machine Learning techniques for generating partial model
  - identification of modality
  - entity recognition for agents
- Use of spreadsheet format

Problems: formalism

Finding suitable formalism

- Expressivity, features, generality
- Validating translation to target formalism
- Formal semantics

Problems: query language

Properties and counter-examples

- We want queries and answers in NL
  a. NL query → property in temp. logic
  b. Manipulation of properties to find answer
  c. “answer” → NL response

Real-world link

- All of this is done in a closed environment
- Realising “the dream” requires collaboration with stakeholders outside our field
- That’s really why we’re here!

Hope you enjoyed listening, very happy to hear your comments and suggestions.

Thanks

References

AnaCon

Angelov, K., Camilleri, J. J., & Schneider, G. (2013). A framework for conflict analysis of normative texts written in controlled natural language. The Journal of Logic and Algebraic Programming.

CL

Prisacariu, C., & Schneider, G. (2009). CL : An Action-based Logic for Reasoning about Contracts. In 16th Workshop on Logic, Language, Information and Computation (WOLLIC'09) (pp. 335–349). Tokyo, Japan: Springer.

CLAN

Fenech, S., Pace, G. J., & Schneider, G. (2009). CLAN: A Tool for Contract Analysis and Conflict Discovery. In Automated Technology for Verification and Analysis (Vol. 5799, pp. 90–96). Berlin, Heidelberg: Springer.

C-O Diagrams

Díaz, G., Cambronero, M. E., Martínez, E., & Schneider, G. (2013). Specification and Verification of Normative Texts using C-O Diagrams. IEEE Transactions on Software Engineering, 1–28.