

WP4: Domain-specific languages

for System Dynamics Models

Patrik Jansson, Prof. of Computer Science

Software Technology division, Chalmers University of Technology

2016-03-10



GRACeFUL

Personnel at the Chalmers site

- Until 2015-09 (8 months) Cezar Ionescu was a PostDoc on the project. Now in Oxford (but remains co-supervisor of Irene).
- PhD student: Irene Lobo Valbuena (started 2015-08-01, 80%)
- Site leader: Patrik Jansson, working 20% for GRACeFUL



Main tasks of WP4 (the DSL work package)

Build a Domain Specific Language (DSL) for translating the concept maps developed and manipulated during Group Model Building (GMB) sessions to system dynamics models adequate for the constraint functional programming (CFP) layer.

The overall purpose of WP4 is to use a DSL for policy concept maps, logic and relations

- to bridge between
 - the complexity of the CRUD case study from WP2 (visualised in WP3)
- and
 - the underlying science and technology of WP5.

In the longer term this will lead to a DSL aimed at building scalable Rapid Assessment Tools (RATs) for collective policy making in Global Systems.

During the project we will work with embedded DSLs to improve scalability, verifiability and correctness of the models.

What have we worked on?

- Description of work

- Done: **T4.1 identify key underlying concepts needed for the CRUD case study**
- Started: **T4.2 develop a DSL to describe the concept maps developed during GMB sessions**
- Started: **T4.3 provide a formal semantics for the elements of the DSL**
- T4.4 implement a middleware for connecting the DSL to the CFP layer
- T4.5 build a testing and verification framework for RATs

- Deliverables

- D4.1 **Done**: Formal description of concept map elements needed for CRUD case study (m6)
- D4.2 **DSL for description of concept maps, with formal semantics (m24)**
- D4.3 Translation of concept map descriptions to system dynamics models for the CFP layer (m30)
- D4.4 Testing and verification framework for RATs with applications to the CRUD case study (m36)

- Main focus: applying the state-of-the-art in Domain Specific Languages to the domain of Climate Resilient Urban Design (CRUD).
- Submitted a paper on “Contributions to a computational theory of policy advice and avoidability” ¹ which is clearly extending the state-of-the-art.
- (Re-submitted paper: “Sequential decision problems, dependent types and generic solutions” ²)

¹Paper submitted 2016-01-06 to the Journal of Functional Programming, Special Issue on Dependently typed Programming. Full text pre-print available. Uses dependent types to model constraints, while GRACeFUL uses relations

²Pre-print and associated source available at [github](#)

- 2015-07: D4.1: Formal description of concept map elements needed for CRUD case study (m6)
 - The work towards the first WP4 deliverable helped clarifying the underlying theoretical basis and the terminology.
 - It contains
 - a short introduction to Functional Programming and DSLs,
 - an explanation of GRACeFUL as computer aided policy analysis,
 - and a formalisation of the core concepts in Haskell.

(Brief history of) Functional Programming, Types and DSLs

- An imperative language usually has “expressions” and “statements”
- It turns out that “expressions” is enough!
- Types are used to check that expressions “make sense”
 - `(1 :: Int)`, `(1.0 :: Double)`, `([1,7] :: [Int])`
- Users defined types and function types

```
data Maybe a = Nothing | Just a
fun :: Input -> Output
```

- Tiny “Domain Specific Language” for simple arithmetics

```
data Expr = Lit Double | Add Expr Expr | Mul Expr Expr
parse :: String -> Maybe Expr
eval  :: Expr -> Double
```

Theory example: Sequential Decision Problems

- The starting point for the “theory of policy advice” paper is
 - A time-dependent state space
 - A time- and state-dependent control space
 - A monadic step function
 - A reward function

State : (t : \mathbb{N}) \rightarrow Type

Control : (t : \mathbb{N}) \rightarrow (x : X t) \rightarrow Type

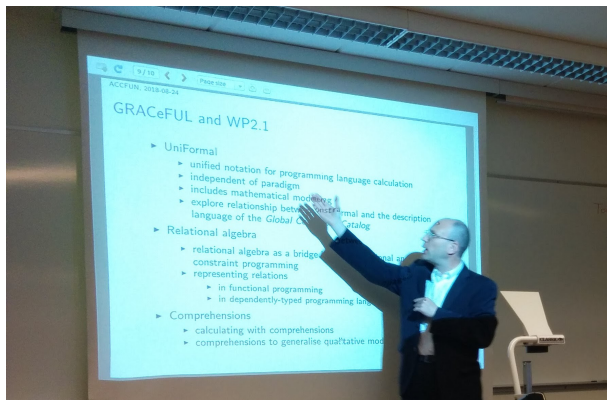
step : (t : \mathbb{N}) \rightarrow (x : X t) \rightarrow (y : Y t x) \rightarrow M (X (S t))

reward : (t : \mathbb{N}) \rightarrow (x : X t) \rightarrow (y : Y t x) \rightarrow X (S t) \rightarrow \mathbb{R}

- The aim is to find a policy that maximises the accumulated reward for a finite time horizon.

ACCFun: the main event of the DSL work package in Y1

- 2015-08: ACCFun: Algorithmic Calculi for Constraint Functional Programming
 - A 5-day “by invitation only”-workshop in the GRACeFUL project.
 - 4 participants from GRACeFUL: Patrik, Cezar, Irene, Tom Schrijvers
 - 28 in total: researchers in software technology and programming languages
 - Topics explored:
 - unified expression of programs, constraints and qualitative models.
 - notations for calculating programs from specifications



Next actions (in year 2)

- T4.2: Create a prototype DSL and iteratively refine it (in collab. with WP2)
- T4.3: Provide a simple semantics for the elements of the DSL
- T4.4: Connect the DSL layer to the CFP layer (WP 5)
- T4.5: Develop simple examples and properties (towards the testing and verification framework)

Related work:

- Connect to the Centre of excellence in Global Systems Science (CoeGSS)
 - Patrik is site-leader in CoeGSS: working on DSLs for Synthetic Populations
 - Potential access to HPC centres and big data expertise
 - Pilot studies on “Global urbanisation” and “Green Growth”
- New BSc course “Domain Specific Languages of Mathematics” (DSLsofMath)
 - Write up lecture notes as a book
 - Could lead to tutorial material for “DSLs for GSS”

Summary of Y1 (Chalmers site and WP4 on DSLs)

- Capture the CRUD domain in a *Domain Specific Language*
- Formalise and implement using *Typed Functional Programming*
- Develop a theory of *policy advice and avoidability*

Questions?



Figure 3:ACCFun venue (near Marstrand, Sweden)

Extra slides: Irene's Individual PhD Study Plan:

- Apply Functional Programming and Software Technology to hybrid modelling
- Develop Domain Specific Languages for system dynamics modelling
- Example: Climate Resilient Urban Design (CRUD)
- Hybrid modelling = must handle both discrete and continuous behaviour
- Policy = a protocol or scheme of actions to take

Aim:

- combine and extend functional and logic programming
- formalise GSS-relevant application domains using embedded DSLs
- provide support for policy making in the frame of global systems by
 - refining and translating the system and goal descriptions,
 - along with actions which could embody policies,
 - to adequate models for a constraint solver framework
 - to search for and characterise feasible policies to take.

“The consortium will organize a number of workshops (3-5) throughout the duration of the project specifically aimed at disseminating the results of the project to related communities. These workshops will be prepared to tackle and address the specific needs of the community particularly selected per invitation, and addressed within the context of a workshop. [...] So each workshop boosts the impact of the project by forwarding and explaining project results to the community, and the project itself benefits from the input and the discussion of the community during the workshop in order to tune the project development road map.”



Figure 4: Tom



Figure 5:Patrik