Literate Scientific Software

Spencer Smith, Dan Szymczak and Jacques Carette

Computing and Software Department
Faculty of Engineering
McMaster University

PASC, MS06, June 16, 2016

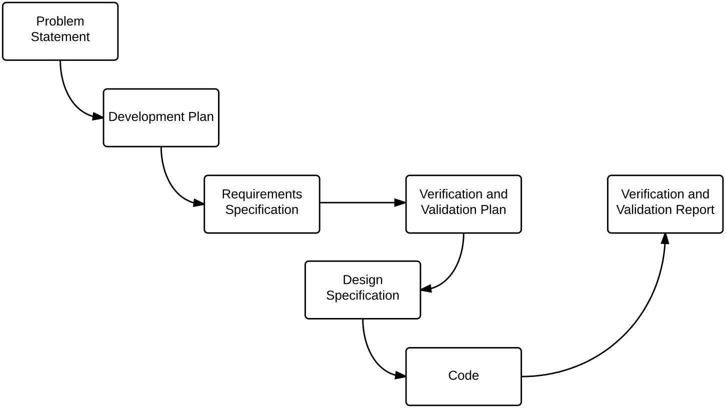
Important SS Qualities

- Reusability
- Maintainability
- Verifiability
- Reproducibility

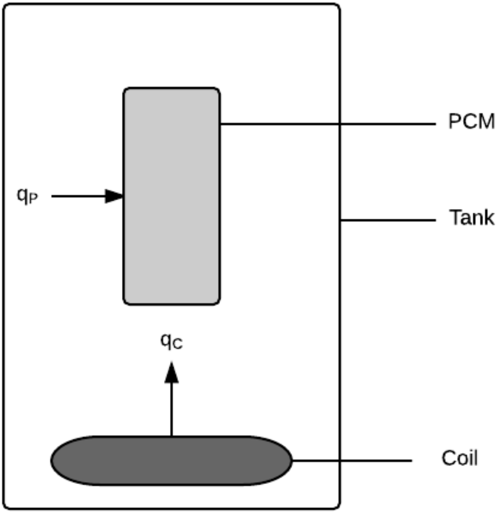
Challenges

- Up front requirements
- Rapid change for numerical algorithms
- Information duplication
- Synchronization headaches between artifacts
- Perceived over-emphasis on non-executable artifacts

"Faked" Rational Design Process



Solar Water Heating Tank



https://github.com/smiths/swhs

Literate Scientific Software

SRS (*verbose*)

SRS (*LaTeX*)

SRS (*html*)

MG

MIS

Test cases

C Code) (*checks*)

Makefile

Matlab (*no checks*)

↑

Recipes

↑

$-\nabla \cdot \mathbf{q} + q''' = \rho C \frac{\partial T}{\partial t}$

uncertainty, typical val- ues etc.

W = J/s = kg m^2 s^{-3}

$L > 0$

h_c is the heat transfer coeff between clad and coolant

How LSS Addresses Challenges

- Supports changing requirements and design
  - Generation
  - Automated traceability
- Supports duplication
  - Knowledge is entered once, generated/transformed
  - Eases maintenance
  - If incorrect, incorrect everywhere
- Non-executable artifacts are generated

Verifiability

Var	Constraints	Typical Value	Uncertainty
L	$L > 0$	1.5 m	10%
D	$D > 0$	0.412 m	10%
V_P	$V_P > 0$	0.05 m^3	10%
A_P	$A_P > 0$	1.2 m^2	10%
ρ_P	$\rho_P > 0$	1007 kg/m^3	10%

$$E_W = \int_0^t h_C A_C (T_C - T_W(t))dt - \int_0^t h_P A_P (T_W(t) - T_P(t))dt$$

- Sanity checks captured and reused
- Generate guards against invalid input
- Generate test cases

Reusability

Num.	**T1**
Label	**Conservation of energy**
Eq	$-\nabla \cdot \mathbf{q} + q''' = \rho C \frac{\partial T}{\partial t}$
Descrip	The above equation gives the conservation of energy for time varying heat transfer in a material of specific heat capacity C and density ρ, where $\mathbf{q}$ is the thermal flux vector, q''' is the volumetric heat generation, T is the temperature, ∇ is the del operator and t is the time.

Maintainability

A1: The only form of energy that is relevant for this problem is thermal energy. All other forms of energy, such as mechanical energy, are assumed to be negligible [T1].

A2: All heat transfer coefficients are constant over time [GD1].

A3: The water in the tank is fully mixed, so the temperature is the same throughout the entire tank [GD2, DD2].

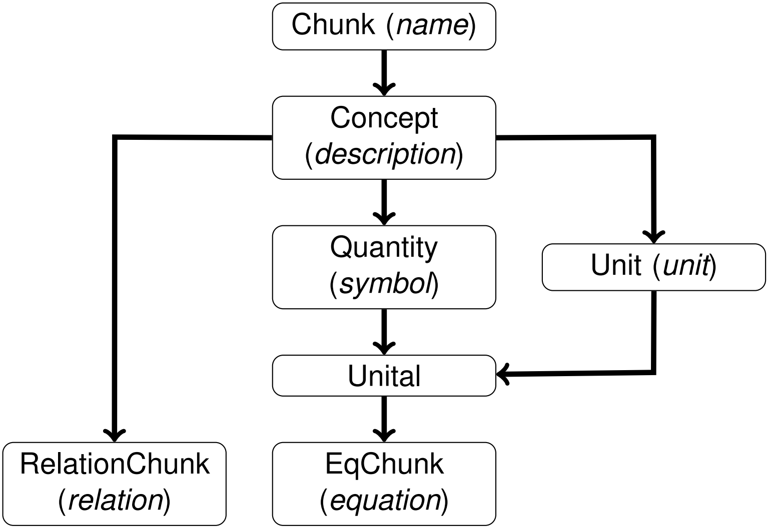
A4: The PCM has the same temperature throughout [GD2, DD2, LC1].

A5: etc.

Reproducibility

- Knowledge is explicitly stored for the future
- Recipes used to regenerate all artifacts, not just code or results
- Recipes include build instructions

Drasil Framework Design



Simple SRS from LaTeX

SRS from LaTeX

```
vars :: [EqChunk]
vars = [h_g, h_c]

s1, s2, s3, s4 :: LayoutObj
s1=table_of_units si_units
s2=table_of_symbols vars
s3=Section 0 (S "Data Definitions") $ map (Definition.Data) vars
s4=Section 0 (S "Code") $ map (CodeBlock.toCode CLang Calc) [h_c]

srs :: Quantity s => [s] -> String -> [LayoutObj] -> Document
srs ls author body =
  Document ((S "SRS for ") :+:
    (foldr1 (:+:) (intersperse (S " and ")
    (map (\x -> U $ x ^. symbol) ls))))
    (S author) body

srsBody :: Document
srsBody = srs vars "Spencer Smith" [s1, s2, s3, s4]
```

```
table_of_symbols :: (Unit s, Quantity s) => [s] -> LayoutObj
table_of_symbols ls=Section 0 (S "Table of Sym") [intro,table ls]

intro :: LayoutObj
intro = Paragraph $
  S "The table that follows ..."

table :: (Unit s, Quantity s) => [s] -> LayoutObj
table ls=Table [S "Symbol",S "Description",S "Units"] (mkTable
  [(\ch -> U (ch ^. symbol)),
   (\ch -> ch ^. descr),
   (\ch -> Sy $ ch ^. unit)] ls)
  (S "Table of Symbols") False
```

```
fundamentals :: [FundUnit]
fundamentals = [metre, kilogram, second, ...]

derived :: [DerUChunk]
derived = [centigrade, joule, watt, calorie, kilowatt]

si_units :: [UnitDefn]
si_units = map UU fundamentals ++ map UU derived
```

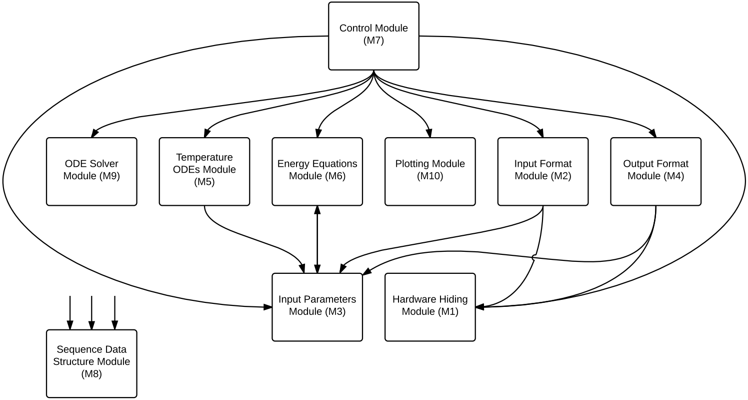
———————— Fundamental SI Units ————————

```
fund :: String -> String -> String -> FundUnit
fund nam desc sym = UD (CC nam (S desc)) (UName $ Atomic sym)

metre, kilogram, second, ... :: FundUnit
metre    = fund "Metre"    "length"              "m"
kilogram = fund "Kilogram" "mass"                "kg"
second   = fund "Second"   "time"                "s"
kelvin   = fund "Kelvin"   "temperature"         "K"
mole     = fund "Mole"     "amount of substance" "mol"
ampere   = fund "Ampere"   "electric current"    "A"
candela  = fund "Candela"  "luminous intensity"  "cd"
```

$$h_c = \frac{2k_c h_b}{2k_c + \tau_c h_b}$$

```
heat_transfer :: DerUChunk
heat_transfer = DUC (UD ht_con ht_symb) heat_transfer_eqn

ht_con :: ConceptChunk
ht_con = makeCC "Heat transfer" "Heat transfer"

ht_symb :: USymb
ht_symb = from_udefn heat_transfer_eqn

heat_transfer_eqn = USynonym (UProd
  [kilogram ^. unit, UPow (second ^. unit) (-3),
   UPow (centigrade ^. unit) (-1)])

h_c_eq :: Expr
h_c_eq = 2*(C k_c)*(C h_b)/(2*(C k_c)+(C tau_c)*(C h_b))

h_c :: EqChunk
h_c = fromEqn "h_c" (S "convective heat transfer ...")
  (lH `sub` lC) heat_transfer h_c_eq
```

Next Steps: Design Documentation



Generate Code for IMs

Input	m_P, C_P^S, C_P^L, h_P, A_P, t_{final}, T_{init}, T_{melt}^P, $T_W(t)$ from IM1
Output	$T_P(t)$, $0 \leq t \leq t_{\text{final}}$, with initial conditions, $T_W(0) = T_P(0) = T_{\text{init}}$ (A12), and $T_W(t)$ from IM1, such that the following governing ODE is satisfied. The specific ODE depends on T_P as follows: $$\frac{dT_P}{dt} = \begin{cases} \frac{dT_P}{dt} = \frac{1}{\tau_P^S}(T_W(t) - T_P(t)) & \text{if } T_P < T_{\text{melt}}^P \\ \frac{dT_P}{dt} = \frac{1}{\tau_P^L}(T_W(t) - T_P(t)) & \text{if } T_P > T_{\text{melt}}^P \\ 0 & \text{if } T_P = T_{\text{melt}}^P \\ & \text{and } 0 < \phi < 1 \end{cases}$$

Approach to Developing Drasil

- Case studies
  - Solar water heating tank
  - Slope stability analysis
  - Glass safety analysis
  - Game physics engine
- Practical
- Not trying to automate everything
- Small chunks of knowledge
- Look for patterns
- Attempt to capture design decisions
  - Version control
  - Issue tracking

Concluding Remarks

- SS has the opportunity to lead other software fields by leveraging its solid existing knowledge base
- Document driven design is feasible with a knowledge-based approach
- Documentation for QA and software certification does not have to be painful, expensive or time consuming
- Drasil will be developed via practical case studies