

PhD Committee Meeting #4

Dan Szymczak

Computing and Software Department
Faculty of Engineering
McMaster University

June 28, 2018

Overview

Recap

Progress

Results

Next Steps

- 1 Research Recap
- 2 Current Progress
- 3 Results to date
- 4 Next Steps.

Research Topic Recap

Motivation

Recap

Progress

Results

Next Steps

- Too much duplication!

Research Topic Recap

Motivation

Recap

Progress

Results

Next Steps

- Too much duplication!
- Inter-/intra-artifact consistency issues

Research Topic Recap

Motivation

Recap

Progress

Results

Next Steps

- Too much duplication!
- Inter-/intra-artifact consistency issues
- Design for change

Research Topic Recap

Motivation

Recap

Progress

Results

Next Steps

- Too much duplication!
- Inter-/intra-artifact consistency issues
- Design for change
- Promote reusability

Research Topic Recap

Motivation

Recap

Progress

Results

Next Steps

- Too much duplication!
- Inter-/intra-artifact consistency issues
- Design for change
- Promote reusability
- (Re-)Certification is expensive

Research Topic Recap

KBSE & The Drasil Framework

A Knowledge-Based Software Engineering Approach

- Too much duplication!
- Inter-/intra-artifact consistency issues
- Design for change
- Promote reusability
- (Re-)Certification is expensive

Research Topic Recap

KBSE & The Drasil Framework

Recap

Progress

Results

Next Steps

A Knowledge-Based Software Engineering Approach

- **Single knowledge-base**
- Inter-/intra-artifact consistency issues
- Design for change
- Promote reusability
- (Re-)Certification is expensive

Research Topic Recap

KBSE & The Drasil Framework

Recap

Progress

Results

Next Steps

A Knowledge-Based Software Engineering Approach

- Single knowledge-base
- **Guaranteed consistency**
- Design for change
- Promote reusability
- (Re-)Certification is expensive

Research Topic Recap

KBSE & The Drasil Framework

Recap

Progress

Results

Next Steps

A Knowledge-Based Software Engineering Approach

- Single knowledge-base
- Guaranteed consistency
- Easy to mix and match
- Promote reusability
- (Re-)Certification is expensive

Research Topic Recap

KBSE & The Drasil Framework

Recap

Progress

Results

Next Steps

A Knowledge-Based Software Engineering Approach

- Single knowledge-base
- Guaranteed consistency
- Easy to mix and match
- Reusable across projects
- (Re-)Certification is expensive

Research Topic Recap

KBSE & The Drasil Framework

Recap

Progress

Results

Next Steps

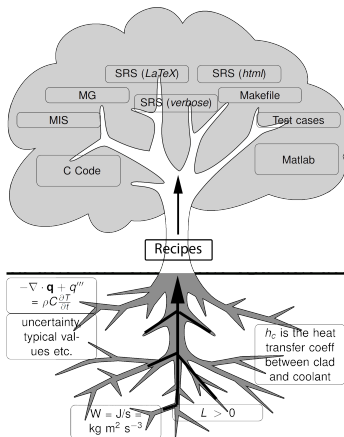
A Knowledge-Based Software Engineering Approach

- Single knowledge-base
- Guaranteed consistency
- Easy to mix and match
- Reusable across projects
- **Generate artifacts**

Research Topic Recap

KBSE & The Drasil Framework

Drasil – Towards generating Software Families



Research Topic Recap

KBSE & The Drasil Framework

Recap

Progress

Results

Next Steps

Drasil – Towards generating Software Families

- One “source”, multiple views
- Full traceability
- Consistent-by-construction artifacts

Research Topic Recap

KBSE & The Drasil Framework

Recap

Progress

Results

Next Steps

Drasil composed of many Domain-Specific Languages (DSLs) including, but not limited to:

- Knowledge Capture
- Recipes (Document generation)
- Code Generation

Current Program Progress

A brief overview

Recap

Progress

Results

Next Steps

- Completed all necessary graduate courses & comprehensive examinations
- Currently Writing:
 - Journal paper for ACM TOSEM
 - Thesis

Current Program Progress

A brief overview continued

Recap

Progress

Results

Next Steps

Research Project: Drasil proof-of-concept “complete”

- Scoped-down due to nature of project
- Generating SRS for six case studies & code for one
- Still improving with the help of summer students

Since Last Time

General

Recap

Progress

Results

Next Steps

Summer 2017: Supervised 5 research students

- Cleaned up case studies
- Helped improve Drasil

Submitted to SE-CoDeSE'17 – Rejected

Began a paper for FASE 2018 – Scrapped

Met with OPG – Positive feedback

Co-supervising 3 research students

Writing

Since Last Time

Drasil-Specific

Recap

Progress

Results

Next Steps

Total of 372 issues closed on the Drasil github

- Haddock
- Chunk and referencing databases
- Improved the Drasil class hierarchy
- Finished creating Document Language (Cont'd)
- Continuous Integration & automated tests
- General source clean-up and refactoring

Currently ~130 open issues guiding development

Results

Document Language

Introduction of new Document Language led to:

- De-embedding a lot of English
- More readable source
- Improved sanity checking

```

1 glassBR_srs :: Document
2 glassBR_srs = Document ((srs ^. defn) +:+ S "for" +:+ (glassBR ^. defn))
   srs_authors
3   [s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11]
4
5 s1 = Section(S "Reference Material") [Con s1_intro , Sub s1_1 , Sub s1_2 ,
6   Sub s1_3]
7
8 s1_intro = Paragraph (S "This section records information for easy reference.")
9
10 s1_1 = table_of_units this_si
11
12 s1_2 = table_of_symbols ((map qs glassBRSymbols) ++
13   (map qs glassBRUnitless)) (^term)
14
15 s1_3 = table_of_abb_and_acronyms acronyms
16
17 s2 = Section(S "Introduction") [Con s2_intro , Sub s2_1 , Sub s2_2 , Sub s2_3]
18
19 s2_intro = Paragraph $
20   S "Software is helpful to efficiently and correctly predict the blast" +:+
21   S "risk involved with the" +:+. (sMap (map toLower) (glaSlab ^. term)) +:+
22   S "The" +:+ (sMap (map toLower) (blast ^. term)) +:+ S "under" +:+
23   S "consideration is" +:+. (sMap (map toLower) (blast ^. defn)) +:+
24   S "The software, herein called" +:+ (glassBR ^. defn) +:+ S "aims to" +:+
25   S "predict the blast risk involved with the" +:+
26   (sMap (map toLower) (glaSlab ^. term)) +:+ S "using an intuitive" +:+
27   S "interface. The following section provides an overview of the" +:+
28   (srs ^. defn) +:+ sParen (srs ^. term) +:+ S "for" +:+. (glassBR ^. defn) +:+
29   S "This section explains the purpose of the" +:+
30   S "document is designed to fulfil , the scope of the requirements and" +:+

```

```

1 glassBR_srs :: Document
2 glassBR_srs = mkDoc mkSRS (for'' titleize phrase) glassSystInfo
3
4 mkSRS :: DocDesc
5 mkSRS = RefSec (RefProg intro [TUnits, tsymb [TSPurpose, SymbOrder], TAandA]) :
6   IntroSec (
7     IntroProg (startIntro software blstRskInvWGlassSlab gLassBR) (short gLassBR)
8       [IPurpose (purpose.intro_p1 document gLassBR glaSlab),
9         IScope incScoR endScoR,
10        IChar (rdrKnldgbleIn glBreakage blastRisk) undIR appStanddIR,
11        IOrgSec intendedReaderIntro dataDefn SRS.dataDefn intendedReaderIntro_end]) :
12   StkhldrSec (StkhldrProg2 [Client gLassBR (S "a" ++ phrase company ++
13     S "named Entuitive. It is developed by Dr." ++ (S $ name mCampidelli)),
14     Cstmr gLassBR]) :
15   GSDSec (GSDProg2
16     [UsrChars [user_chars_bullets endUser gLassBR secondYear undergradDegree
17       civilEng structuralEng glBreakage blastRisk], SystCons [] []]) :
18   ScpOfProjSec (ScpOfProjProg (short gLassBR) (prod_use_case_table)
19     (indiv_prod_use_case (glaSlab) (capacity) (demandq) (probability)))
20   :
21   SSDSec (SSDProg [SSDProblem (
22     PDProg start gLassBR ending [terms_defs, phys_sys_desc, goals]]),
23     [ SSDSolChSpec (
24       SCSProg
25         [ Assumptions
26           , TMs ([Label] ++ stdFields) [t1IsSafe]
27           , GDs [] [] HideDerivation — No Gen Defs for GlassBR
28           , DDs ([Label, Symbol, Units] ++ stdFields) dataDefns ShowDerivation
29           , IMs ([Label, Input, Output, InConstraints, OutConstraints] ++
30             stdFields) [probOfBreak, testIMFromQD] HideDerivation
31         ]

```

```

1  glassBR_srs :: Document
2  glassBR_srs = mkDoc mkSRS (for '' titleize phrase) glassSystInfo
3
4  mkSRS :: DocDesc
5  mkSRS = RefSec (RefProg intro [TUnits, tsymb [TSPurpose, SymbOrder], TAandA]) :
6    IntroSec (
7      IntroProg (startIntro software blstRskInvWGlassSlab gLassBR) (short gLassBR)
8        [IPurpose (purpose_intro_p1 document gLassBR glaSlab),
9          IScope incScoR endScoR,
10         IChar (rdrKnldgbleIn glBreakage blastRisk) undIR appStanddIR,
11         IOrgSec intendedReaderIntro dataDefn SRS.dataDefn intendedReaderIntro_end]) :
12
13  StkhldrSec (StkhldrProg2 [Client gLassBR (S "a" ++ phrase company ++
14    S "named Entuitive. It is developed by Dr." ++ (S $ name mCampidelli)),
15    Cstmr gLassBR]) :
16  GSDSec (GSDProg2
17    [UsrChars [user_chars_bullets endUser gLassBR secondYear undergradDegree
18      civilEng structuralEng glBreakage blastRisk], SystCons [] []]) :
19  ScpOfProjSec (ScpOfProjProg (short gLassBR) (prod_use_case.table)
20    (indiv_prod_use_case (glaSlab) (capacity) (demandq) (probability)))
21  :
22  SSDSec (SSDProg [SSDProblem (
23    PDProg start gLassBR ending [terms_defs, phys_sys_desc, goals]]),
24    [ SSDSolChSpec (
25      SCSPProg
26        [ Assumptions
27          , TMs ([Label] ++ stdFields) [t1IsSafe]
28          , GDs [] [] HideDerivation — No Gen Defs for GlassBR
29          , DDs ([Label, Symbol, Units] ++ stdFields) dataDefns ShowDerivation
30          , IMs ([Label, Input, Output, InConstraints, OutConstraints] ++
31              stdFields) [probOfBreak, testIMFromQD] HideDerivation

```


Results

Sanity Checking

SSP Example (Issue #348)

$$S_i = \frac{P_i}{FS} \quad (1)$$

$$FS = \frac{S_i}{\tau_i} \quad (2)$$

Results

Sanity Checking

SSP Example (Issue #348)

$$S_i = \frac{P_i}{FS} \quad (1)$$

$$FS = \frac{S_i}{\tau_i} \quad (2)$$

Where did τ_i come from?

Were S_i and P_i swapped?

Results

Sanity Checking

SSP Example (Issue #348)

$$S_i = \frac{P_i}{FS} \quad (1) \qquad FS = \frac{S_i}{\tau_i} \quad (2)$$

Where did τ_i come from?

Were S_i and P_i swapped?

- τ_i was not defined anywhere in the documents
- Found with Drasil – undefined symbols throw errors
- Equation based on concepts – symbols automatically retrieved

Results

Conceptual Inconsistencies in Software Artifacts

Manually created artifacts are human-readable.
Problems arise when explaining things to a machine.

- What do our artifacts *mean*?
- What is each section contributing?
- Why do we organize things a given way?
- Are models/definitions different? How?

Need to be more rigorous!

Next Steps

Broad Strokes

What next?

- Finish writing paper for ACM TOSEM
- Complete Thesis writing
- Continue improving Drasil

Problem:

- Duplication
- Inconsistency
- Design for change
- Promote re-usability
- Re-certification

Solution:

Capture fine-grained knowledge & generate **ALL** artifacts