First Committee Meeting

Progress Report

Jason Balaci

McMaster University

Oct. 21st, 2021

Table of Contents

1. Introduction

2. Project
   - Drasil
   - Goal #1: Typed Expression Language
   - Goal #2: Model Discrimination – "ModelKinds"

3. References

Table of Contents

1. Introduction

2. Project
   - Drasil
   - Goal #1: Typed Expression Language
   - Goal #2: Model Discrimination – "ModelKinds"

3. References

Who am I?

Who am I?

- I am **Jason Balaci**

Me, Fall 2019

Who am I?

- I am **Jason Balaci**
- Graduate of *McMaster University*, holding...

Me, Fall 2019

Who am I?

- I am **Jason Balaci**
- Graduate of *McMaster University*, holding...
  - Hons. Actuarial and Financial Mathematics (B.Sc.)

Me, Fall 2019

Who am I?

- I am **Jason Balaci**
- Graduate of *McMaster University*, holding...
  - Hons. Actuarial and Financial Mathematics (B.Sc.)
  - Minor in Computer Science

Me, Fall 2019

Who am I?

- I am **Jason Balaci**
- Graduate of *McMaster University*, holding...
  - Hons. Actuarial and Financial Mathematics (B.Sc.)
  - Minor in Computer Science
- Currently pursuing a thesis-based Master's of Computer Science (M.Sc) at *McMaster University*, under the supervision of **Dr. Jacques Carette**

Me, Fall 2019

Overview of Progression Towards C.S. M.Sc.

Course-related progression

- I'm required to complete[1]:

[1] https://academiccalendars.romcmaster.ca/preview_program.php?catoid=45&poid=23470&returnto=9166

[2] http://www.cas.mcmaster.ca/cas/0files/reg_master_cs_2019a.pdf

Overview of Progression Towards C.S. M.Sc.

Course-related progression

- I'm required to complete[1]:
  - One (1) "Software" course

[1] https://academiccalendars.romcmaster.ca/preview_program.php?catoid=45&poid=23470&returnto=9166

[2] http://www.cas.mcmaster.ca/cas/0files/reg_master_cs_2019a.pdf

Overview of Progression Towards C.S. M.Sc.

Course-related progression

- I'm required to complete[1]:
  - One (1) "Software" course
  - Either of:

[1] https://academiccalendars.romcmaster.ca/preview_program.php?catoid=45&poid=23470&returnto=9166

[2] http://www.cas.mcmaster.ca/cas/0files/reg_master_cs_2019a.pdf

Overview of Progression Towards C.S. M.Sc.

Course-related progression

- I'm required to complete[1]:
  - One (1) "Software" course
  - Either of:
    - Two "Theory" courses, and one "Systems" course
    - One "Theory" course, and two "Systems" courses

[1] https://academiccalendars.romcmaster.ca/preview_program.php?catoid=45&poid=23470&returnto=9166

[2] http://www.cas.mcmaster.ca/cas/0files/reg_master_cs_2019a.pdf

Overview of Progression Towards C.S. M.Sc.

Course-related progression

- I'm required to complete[1]:
  - One (1) "Software" course
  - Either of:
    - Two "Theory" courses, and one "Systems" course
    - One "Theory" course, and two "Systems" courses
- I've completed:

[1] https://academiccalendars.romcmaster.ca/preview_program.php?catoid=45&poid=23470&returnto=9166

[2] http://www.cas.mcmaster.ca/cas/0files/reg_master_cs_2019a.pdf

Overview of Progression Towards C.S. M.Sc.

Course-related progression

- I'm required to complete[1]:
  - One (1) "Software" course
  - Either of:
    - Two "Theory" courses, and one "Systems" course
    - One "Theory" course, and two "Systems" courses
- I've completed:
  - CAS 701 "Logic & Discrete Mathematics" - Theory course, Fall 2020
  - CAS 761 "Generative Programming" - Software course, Fall 2020
  - CAS 763 "Certified Programming with Dependent Types" - Theory & Software course, Winter 2021
  - COMPSCI 6TB3 "Syntax-Based Tools and Compilers" - Systems course, Winter 2021

[1] https://academiccalendars.romcmaster.ca/preview_program.php?catoid=45&poid=23470&returnto=9166

[2] http://www.cas.mcmaster.ca/cas/0files/reg_master_cs_2019a.pdf

Overview of Progression Towards C.S. M.Sc.

Course-related progression

- I'm required to complete[1]:
  - One (1) "Software" course
  - Either of:
    - Two "Theory" courses, and one "Systems" course
    - One "Theory" course, and two "Systems" courses
- I've completed:
  - CAS 701 "Logic & Discrete Mathematics" - Theory course, Fall 2020
  - CAS 761 "Generative Programming" - Software course, Fall 2020
  - CAS 763 "Certified Programming with Dependent Types" - Theory & Software course, Winter 2021
  - COMPSCI 6TB3 "Syntax-Based Tools and Compilers" - Systems course, Winter 2021
- Together, the courses completed satisfies the "Courses Requirement" as mentioned in the academic calendar[1] and the "Regulations for the Computer Science M.Sc. Program" document[2].

[1] https://academiccalendars.romcmaster.ca/preview_program.php?catoid=45&poid=23470&returnto=9166

[2] http://www.cas.mcmaster.ca/cas/0files/reg_master_cs_2019a.pdf

Overview of Progression Towards C.S. M.Sc.

Thesis/research-related Progression

- Conducted "full-time" research for at least 1 full semester (Spring/Summer 2021), and "part-time" research during courses.

Overview of Progression Towards C.S. M.Sc.

Thesis/research-related Progression

- Conducted "full-time" research for at least 1 full semester (Spring/Summer 2021), and "part-time" research during courses.
- Continuing to research "full-time".

Overview of Progression Towards C.S. M.Sc.

Thesis/research-related Progression

- Conducted "full-time" research for at least 1 full semester (Spring/Summer 2021), and "part-time" research during courses.
- Continuing to research "full-time".
- Attended a thesis defence to learn about what to expect from a thesis defence meeting (and learn about their research).

Overview of Progression Towards C.S. M.Sc.

Thesis/research-related Progression

- Conducted "full-time" research for at least 1 full semester (Spring/Summer 2021), and "part-time" research during courses.
- Continuing to research "full-time".
- Attended a thesis defence to learn about what to expect from a thesis defence meeting (and learn about their research).
- Supervisory committee is formed, and we're currently having our first supervisory committee.
  - *Supervisor*: Dr. Jacques Carette
  - Dr. Spencer Smith
  - Dr. Wolfram Kahl

Table of Contents

1 Introduction

2 Project
- Drasil
- Goal #1: Typed Expression Language
- Goal #2: Model Discrimination – "ModelKinds"

3 References

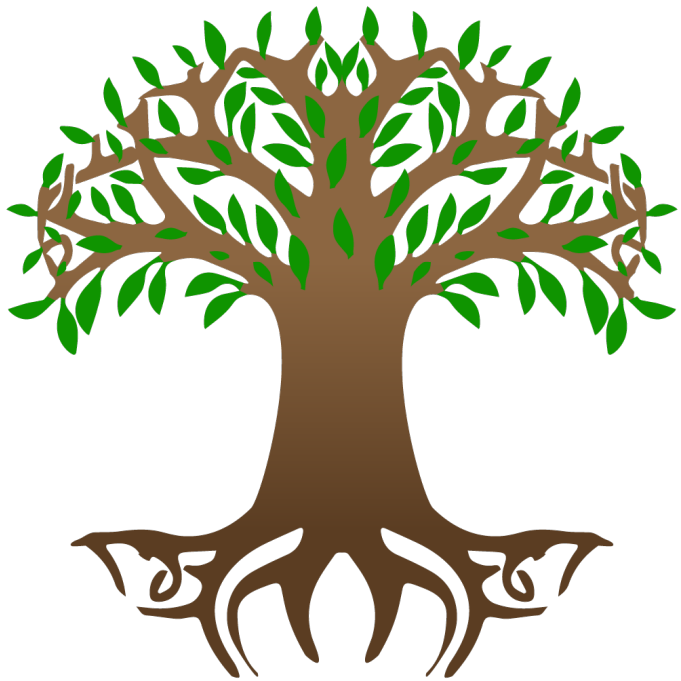
Drasil...

- is managed by Dr. Carette & Dr. Smith.

Drasil's Logo
[Carette et al., 2021][Yggdrasil - Wikipedia, 2021]

Drasil...

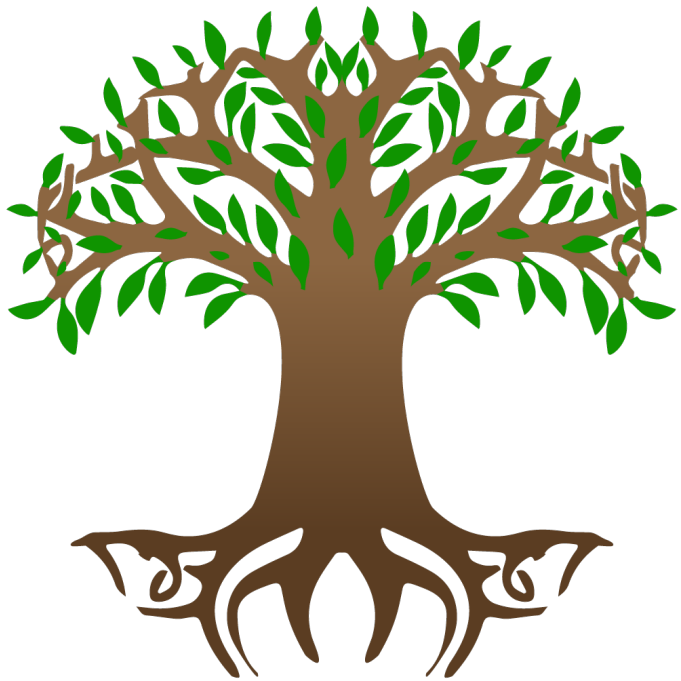
- is managed by Dr. Carette & Dr. Smith.
- originates from the work of Dan Szymczak.



Drasil's Logo
[Carette et al., 2021][Yggdrasil - Wikipedia, 2021]

Drasil...

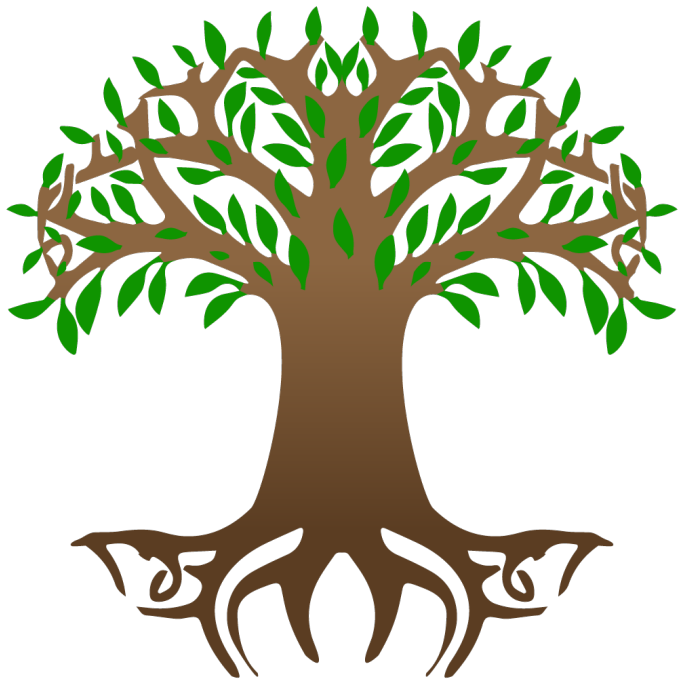
- is managed by Dr. Carette & Dr. Smith.
- originates from the work of Dan Szymczak.
  - Originally focused on scientific software (*Literate Scientific Software*).



Drasil's Logo
[Carette et al., 2021][Yggdrasil - Wikipedia, 2021]

Drasil...

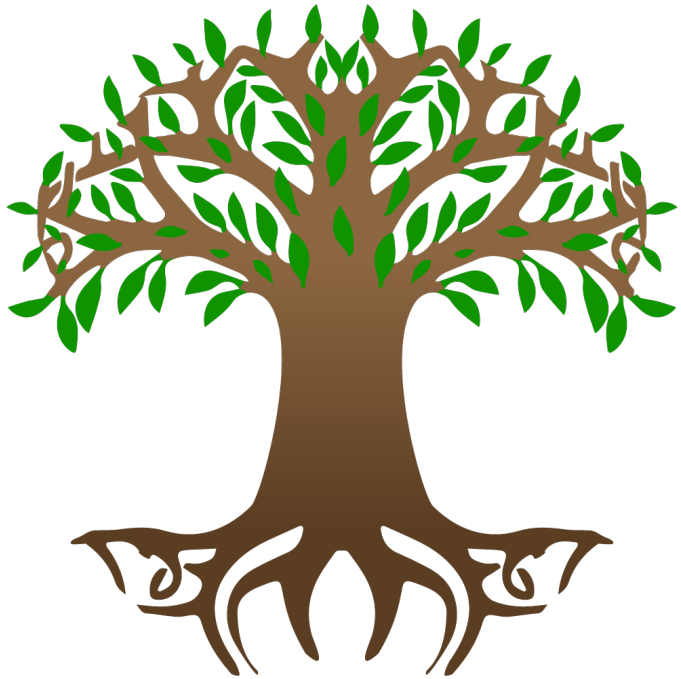
- is managed by Dr. Carette & Dr. Smith.
- originates from the work of Dan Szymczak.
  - Originally focused on scientific software (*Literate Scientific Software*).
  - Focus expanded...



Drasil's Logo
[Carette et al., 2021][Yggdrasil - Wikipedia, 2021]

Drasil...

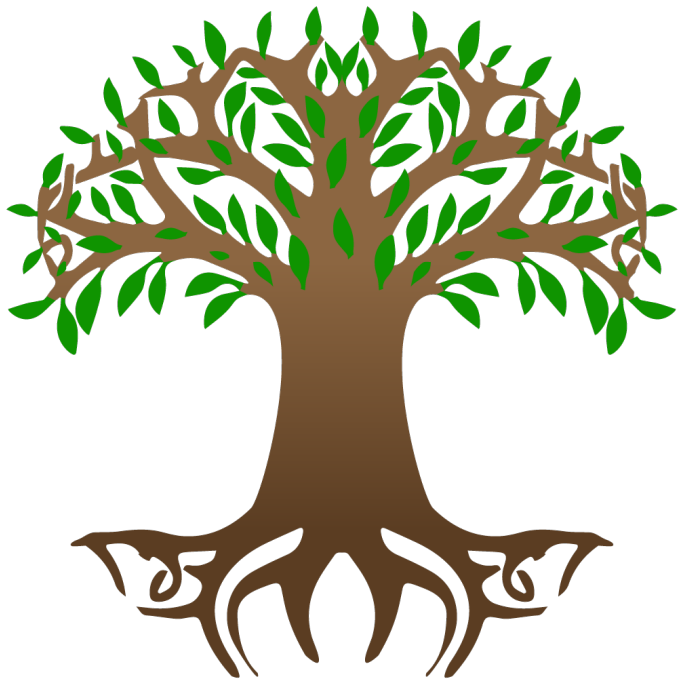
- is managed by Dr. Carette & Dr. Smith.
- originates from the work of Dan Szymczak.
  - Originally focused on scientific software (*Literate Scientific Software*).
  - Focus expanded...
- tries to "Generate All The Things"...



Drasil's Logo
[Carette et al., 2021][Yggdrasil - Wikipedia, 2021]

Drasil...

- is managed by Dr. Carette & Dr. Smith.
- originates from the work of Dan Szymczak.
  - Originally focused on scientific software (*Literate Scientific Software*).
  - Focus expanded...
- tries to "Generate All The Things"...
  - with a focus on research software.



Drasil's Logo
[Carette et al., 2021][Yggdrasil - Wikipedia, 2021]

Drasil

"Generate All The Things!"

- TODO: here!

Drasil Case Studies

Drasil Case Studies

- Drasil currently contains a significant amount of Physics-related knowledge.

Drasil Case Studies

- Drasil currently contains a significant amount of Physics-related knowledge.
- As of writing, current case studies include:

Drasil Case Studies

- Drasil currently contains a significant amount of Physics-related knowledge.
- As of writing, current case studies include:
  - **GlassBR** - Predicting whether or not a glass slab is likely to resist a specified blast.

Drasil Case Studies

- Drasil currently contains a significant amount of Physics-related knowledge.
- As of writing, current case studies include:
  - **GlassBR** - Predicting whether or not a glass slab is likely to resist a specified blast.
  - **Single Pendulum** - Observing the motion of a single pendulum.

Drasil Case Studies

- Drasil currently contains a significant amount of Physics-related knowledge.
- As of writing, current case studies include:
  - **GlassBR** - Predicting whether or not a glass slab is likely to resist a specified blast.
  - **Single Pendulum** - Observing the motion of a single pendulum.
  - **Double Pendulum** - Observing the motion of a double pendulum.

Drasil Case Studies

- Drasil currently contains a significant amount of Physics-related knowledge.
- As of writing, current case studies include:
  - **GlassBR** - Predicting whether or not a glass slab is likely to resist a specified blast.
  - **Single Pendulum** - Observing the motion of a single pendulum.
  - **Double Pendulum** - Observing the motion of a double pendulum.
  - **Game Physics** - Modelling of an open source 2D rigid body physics library used for games.

Drasil Case Studies

- Drasil currently contains a significant amount of Physics-related knowledge.
- As of writing, current case studies include:
  - **GlassBR** - Predicting whether or not a glass slab is likely to resist a specified blast.
  - **Single Pendulum** - Observing the motion of a single pendulum.
  - **Double Pendulum** - Observing the motion of a double pendulum.
  - **Game Physics** - Modelling of an open source 2D rigid body physics library used for games.
  - **Proportional Derivative Controller (PDController)** - Examining the output of a "Power Plant" (Process Variable) over time.

Drasil Case Studies

- Drasil currently contains a significant amount of Physics-related knowledge.
- As of writing, current case studies include:
  - **GlassBR** - Predicting whether or not a glass slab is likely to resist a specified blast.
  - **Single Pendulum** - Observing the motion of a single pendulum.
  - **Double Pendulum** - Observing the motion of a double pendulum.
  - **Game Physics** - Modelling of an open source 2D rigid body physics library used for games.
  - **Proportional Derivative Controller (PDController)** - Examining the output of a "Power Plant" (Process Variable) over time.
  - **Solar Water Heating System (SWHS)** - Modelling of a solar water heating system with phase change material, predicting temperatures and change in heat energy of water and the PCM over time.

What does Drasil currently support?

- *cont.d*:
  - **SWHS without Phase Change Material (NoPCM)** - Modelling of a solar water heating system without phase change material, predicting temperatures and change in heat energy of water and the PCM over time.

What does Drasil currently support?

- *cont.d*:
  - **SWHS without Phase Change Material (NoPCM)** - Modelling of a solar water heating system without phase change material, predicting temperatures and change in heat energy of water and the PCM over time.
  - **Projectile** - Determining if a launched projectile hits a target, assuming no flight collisions.

What does Drasil currently support?

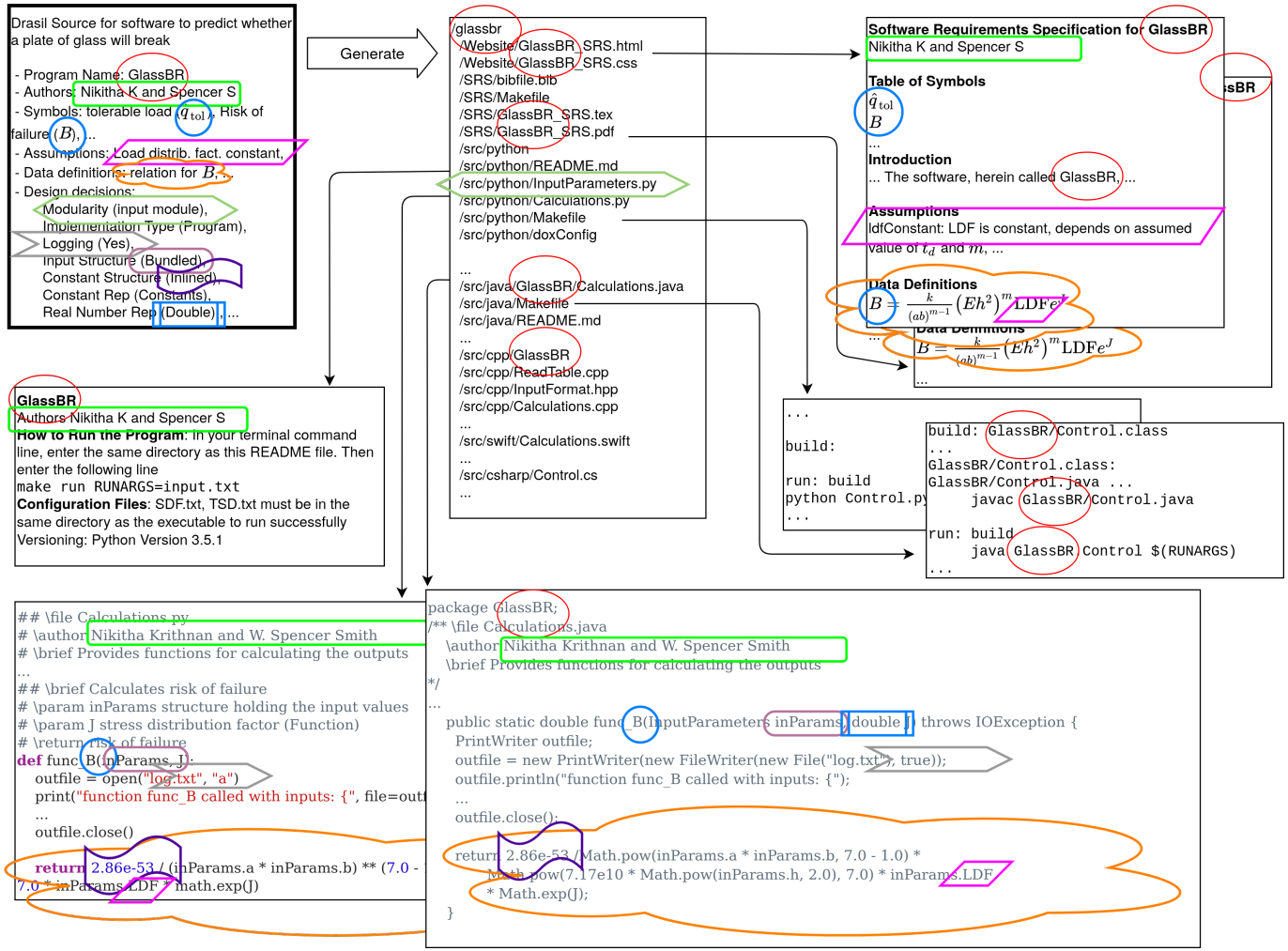
- *cont.d*:
  - **SWHS without Phase Change Material (NoPCM)** - Modelling of a solar water heating system without phase change material, predicting temperatures and change in heat energy of water and the PCM over time.
  - **Projectile** - Determining if a launched projectile hits a target, assuming no flight collisions.
  - **Slope Stability Analysis (SSP)** - Assessment of the safety of a slope (composed of rock and soil) subject to gravity, identifying the surface most likely to experience slip and an index of its relative stability (factor of safety).

What does Drasil currently support?

- *cont.d*:
  - **SWHS without Phase Change Material (NoPCM)** - Modelling of a solar water heating system without phase change material, predicting temperatures and change in heat energy of water and the PCM over time.
  - **Projectile** - Determining if a launched projectile hits a target, assuming no flight collisions.
  - **Slope Stability Analysis (SSP)** - Assessment of the safety of a slope (composed of rock and soil) subject to gravity, identifying the surface most likely to experience slip and an index of its relative stability (factor of safety).
  - **Heat Transfer Coefficients between Fuel and Cladding in Fuel Rods (HGHC)** - Examining the heat transfer coefficients related to clad.

Taking a closer look at one of the examples: GlassBR

GlassBR Generates Code!



Knowledge flow from "knowledge-base"/source code to artifacts, by Dr. Spencer Smith

Which case studies currently generate code?

- **GlassBR** - Predicting whether or not a glass slab is likely to resist a specified blast.

Which case studies currently generate code?

- **GlassBR** - Predicting whether or not a glass slab is likely to resist a specified blast.
- **Proportional Derivative Controller (PDController)** - Examining the output of a "Power Plant" (Process Variable) over time.

Which case studies currently generate code?

- **GlassBR** - Predicting whether or not a glass slab is likely to resist a specified blast.
- **Proportional Derivative Controller (PDController)** - Examining the output of a "Power Plant" (Process Variable) over time.
- **SWHS without Phase Change Material (NoPCM)** - Modelling of a solar water heating system without phase change material, predicting temperatures and change in heat energy of water and the PCM over time.

Which case studies currently generate code?

- **GlassBR** - Predicting whether or not a glass slab is likely to resist a specified blast.
- **Proportional Derivative Controller (PDController)** - Examining the output of a "Power Plant" (Process Variable) over time.
- **SWHS without Phase Change Material (NoPCM)** - Modelling of a solar water heating system without phase change material, predicting temperatures and change in heat energy of water and the PCM over time.
- **Projectile** - Determining if a launched projectile hits a target, assuming no flight collisions.

Why don't all case studies generate software artifacts?

Where will I be contributing?

[1] Terminology is currently being changed, but is not reflected in many documents yet.

Why don't all case studies generate software artifacts?

Where will I be contributing?

After all,

[1] Terminology is currently being changed, but is not reflected in many documents yet.

Why don't all case studies generate software artifacts?

Where will I be contributing?

After all,

- They're all "well-understood"!

[1] Terminology is currently being changed, but is not reflected in many documents yet.

Why don't all case studies generate software artifacts?

Where will I be contributing?

After all,

- They're all "well-understood"!
- The SRS documents are generated!

[1] Terminology is currently being changed, but is not reflected in many documents yet.

Why don't all case studies generate software artifacts?

Where will I be contributing?

After all,

- They're all "well-understood"!
- The SRS documents are generated!

A few, notable, blocking problems:

[1] Terminology is currently being changed, but is not reflected in many documents yet.

Why don't all case studies generate software artifacts?

Where will I be contributing?

After all,

- They're all "well-understood"!
- The SRS documents are generated!

A few, notable, blocking problems:

- Confidently generating usable software artifacts without strong type information places significant stress on developers, resulting in a higher likelihood of bugs in artifacts.

[1] Terminology is currently being changed, but is not reflected in many documents yet.

Why don't all case studies generate software artifacts?

Where will I be contributing?

After all,

- They're all "well-understood"!
- The SRS documents are generated!

A few, notable, blocking problems:

- Confidently generating usable software artifacts without strong type information places significant stress on developers, resulting in a higher likelihood of bugs in artifacts.
- Existing "theories"/"*Models"[1] don't expose enough information. They must be enriched.

[1] Terminology is currently being changed, but is not reflected in many documents yet.

Acknowledgements

Fin.

Thank you!

Table of Contents

1. Introduction

2. Project
   - Drasil
   - Goal #1: Typed Expression Language
   - Goal #2: Model Discrimination – "ModelKinds"

3. References

References I


Carette, J., Smith, S., Balaci, J., Hunt, A., Wu, T.-Y., Crawford, S., Chen, D., Szymczak, D., MacLachlan, B., Scime, D., and Niazi, M. (2021).
Drasil.


Yggdrasil - Wikipedia (2021).
Yggdrasil.