



# CAPTURING MATHEMATICAL KNOWLEDGE IN DRASIL: THE CASE OF THEORIES

Jason Balaci (balacij), Jacques Carette (carette)  
Department of Computing and Software, McMaster University



Computing  
& Software

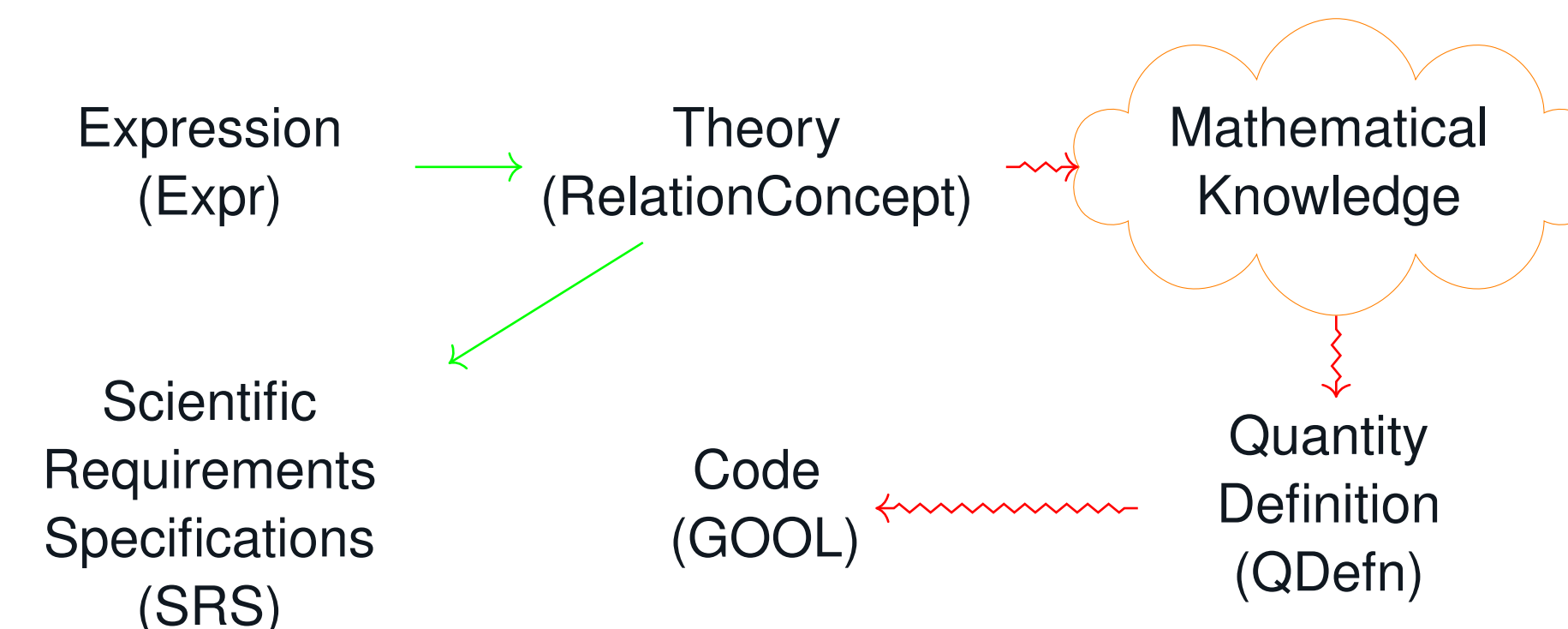
## What is “Drasil”?

Drasil is a framework for generating families of software artifacts from a coherent knowledge base, following its mantra; “Generate All The Things!”. Drasil uses a series of variably sized Domain-Specific Languages (DSLs) to describe various fragments of knowledge that domain experts and users alike may use to piece together fragments of knowledge into a coherent “story”. Through forming some coherent “story” in a domain captured by Drasil, a representational software artifact may be generated. Drasil currently focuses on Scientific Computing Software (SCS), following Smith and Lai’s Software Requirements Specifications (SRS) template as described in [3]. Behind the scenes of the SRS, a mathematical language is used to describe various theories, and have representational software constructed via compiling to Generic Object-Oriented Language (GOOL) [1]. Through encoding knowledge in Drasil, an increase in productivity (and maintainability) in building reliable and traceable software artifacts is observed [4], specifically in SCS [2].

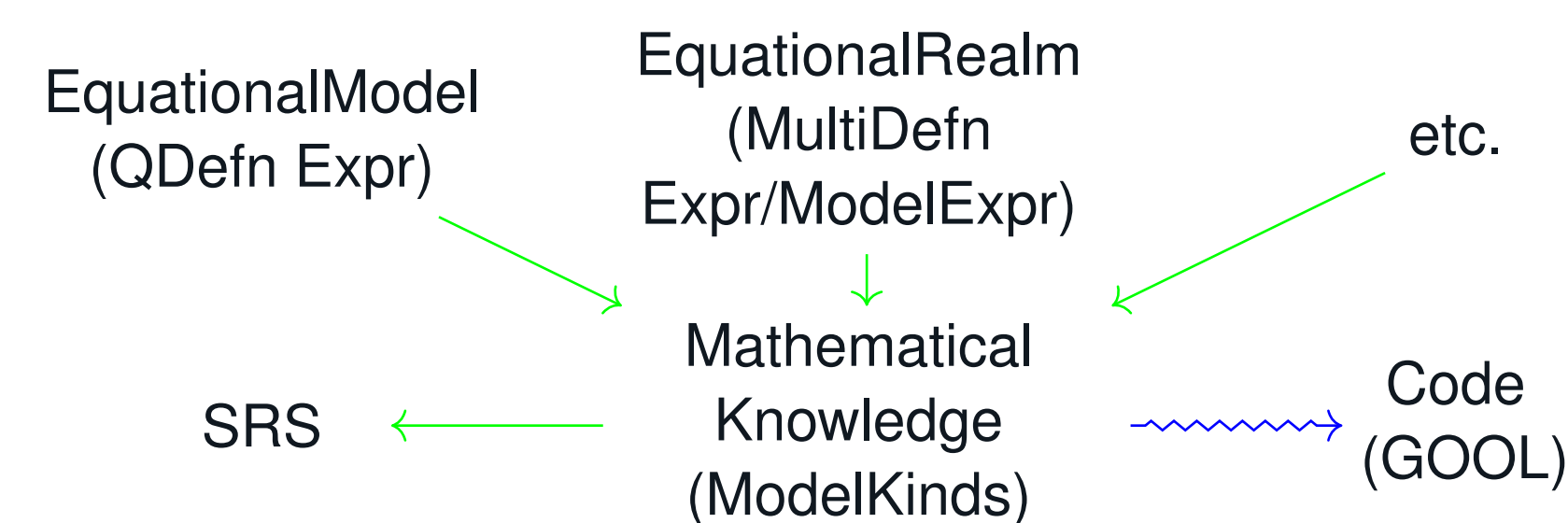
## What’s the problem?

## Why do we care?

## Mathematical Knowledge Flow



## Capturing Mathematical Knowledge



## Comparison

## Future Work

- Something

## Acknowledgements

## References

- [1] Jacques Carette, Brooks MacLachlan, and Spencer Smith. “GOOL: a generic object-oriented language”. In: *Proceedings of the 2020 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation*. 2020, pp. 45–51.
- [2] W. Spencer Smith. “Beyond Software Carpentry”. In: *2018 International Workshop on Software Engineering for Science (held in conjunction with ICSE’18)*. 2018, pp. 32–39.
- [3] W. Spencer Smith and Lei Lai. “A New Requirements Template for Scientific Computing”. In: *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP’05*. In conjunction with 13th IEEE International Requirements Engineering Conference. Paris, France, 2005, pp. 107–121.
- [4] Daniel Szymczak, W. Spencer Smith, and Jacques Carette. “Position Paper: A Knowledge-Based Approach to Scientific Software Development”. In: *Proceedings of SE4Science’16 in conjunction with the International Conference on Software Engineering (ICSE)*. Austin, Texas, United States, May 2016.