# Ensō, redux

## William’s vision, and a new implementation

Tijs van der Storm, 11 March, 2022

CWI

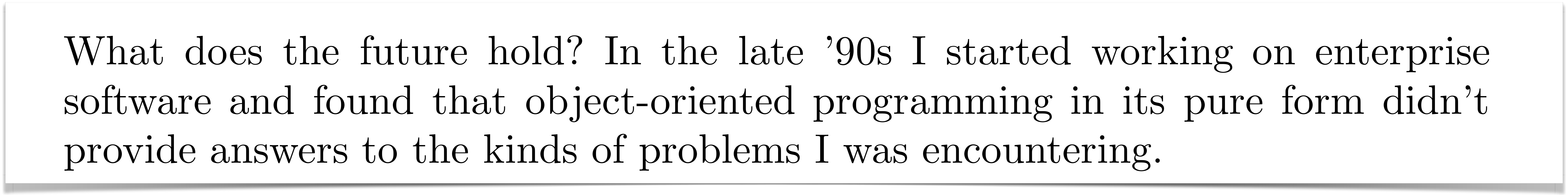
Centrum Wiskunde & Informatica

university of groningen

# What is Ensō?

- Is it a language workbench?
- An architectural pattern?
- A model-driven framework?
- Object relational mapping?
- A programming paradigm?
- **No, it’s Ensō**
- (all of the above, and then some)

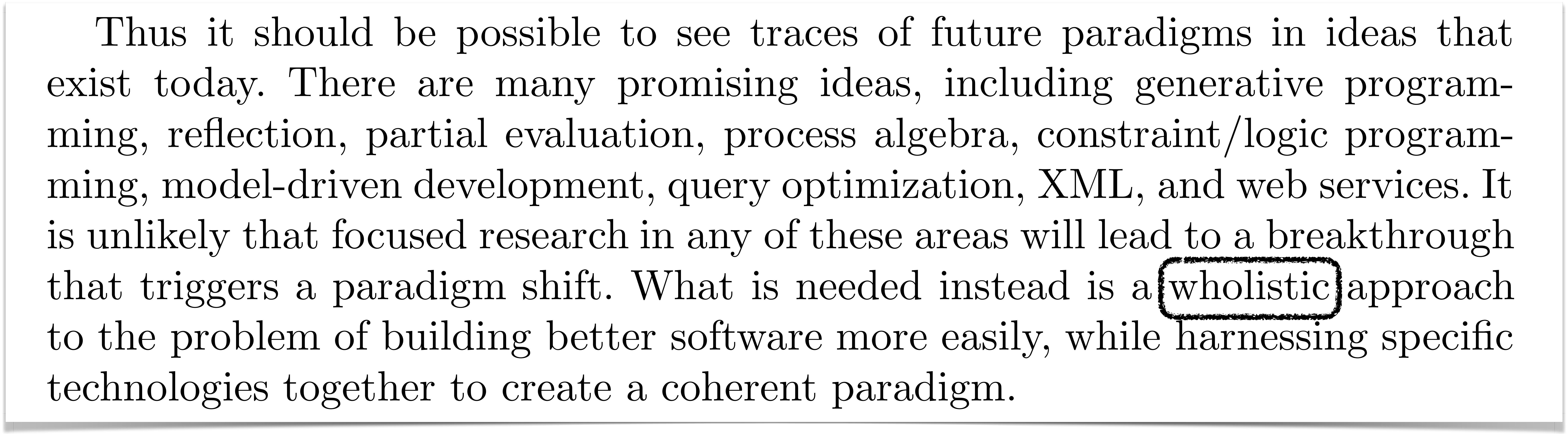
# Peak objects



What does the future hold? In the late ’90s I started working on enterprise software and found that object-oriented programming in its pure form didn’t provide answers to the kinds of problems I was encountering.

William R. Cook, *Peak Objects,* ECOOP 2006 – Object-Oriented Programming pp 180-185, LNCS 4067

# Peak objects



Thus it should be possible to see traces of future paradigms in ideas that exist today. There are many promising ideas, including generative programming, reflection, partial evaluation, process algebra, constraint/logic programming, model-driven development, query optimization, XML, and web services. It is unlikely that focused research in any of these areas will lead to a breakthrough that triggers a paradigm shift. What is needed instead is a wholistic approach to the problem of building better software more easily, while harnessing specific technologies together to create a coherent paradigm.

William R. Cook, *Peak Objects,* ECOOP 2006 – Object-Oriented Programming pp 180-185, LNCS 4067

# Peak objects

I want a more declarative description of systems. I find myself using domain-specific languages: for semantic data models, security rules, user interfaces, grammars, patterns, queries, consistency constraints, upgrade transformations, workflow processes. Little bits of procedural code may be embedded in the declarative framework, acting as procedural plugins.

William R. Cook, *Peak Objects,* ECOOP 2006 – Object-Oriented Programming pp 180-185, LNCS 4067

# Ensō: a prototype to realize that vision

## a constellation of little languages and tools

### Languages/DSLs

- Schema: data modeling, ASG
- Grammar: parsing/formatting
- Diagram: graphical editors
- Security: access policies
- Web: web applications
- **All** have schema, grammar, and one or more interpreters

### Generic Tools

- Print
- Traverse and fold
- Diff: versioning/data migration
- Merge: the basis for modularity
- Persist
- …
- *All guided by model's schema*

# Demo

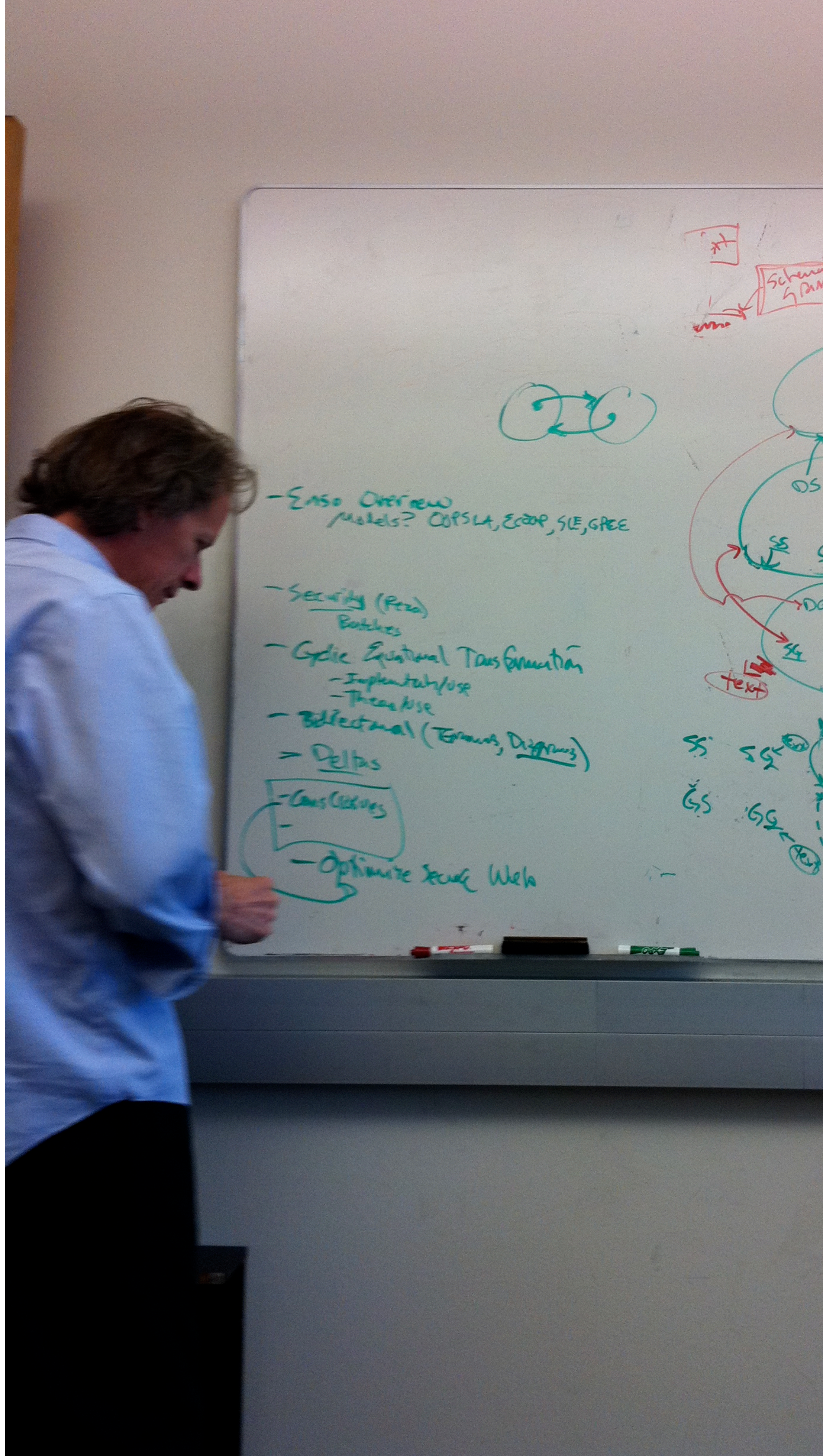
# Related areas of William’s work

- Object Algebras: to modularize interpreters
- Functional graphs: inspired by Ensō models
- Batches: fast remote execution/query processing
- Orc: workflow language (“and I haven’t even integrated Orc!”)
- Generic Extensible Language (GEL)
- Hybrid Partial Evaluation: to make Ensō fast…

# Ensō

## Some of William's slogans…

- Don't design your programs, program your designs
- DSLs: synthesis lite + verification lite
- Enable good, instead of preventing bad
- AOP is a very bad solution for a great problem
- The Smalltalk of modeling
- Objects inside
- Overriding the "."


-Enso Overview
Models? OOPSLA, ECOOP, SLE, GPCE
-Security (Pets)
Batches
-Cyclic Equational Transformation
-Bidirectional (Grammars, Diagrams)
-Deltas
-Optimize Secure Web
text