Domain-Specific Languages for Finance
DSL Research in the HIPERFIT Research Center

Jost Berthold

Department of Computer Science
University of Copenhagen

Outline of the Talk

1. The HIPERFIT Research Center
   - HIPERFIT Research Organisation
   - Motivation and Main Hypothesis
   - Research Themes and Areas

2. On Domain-Specific Languages in HIPERFIT
   - Domain-Specific Languages
   - ... for Economics and Finance
   - Vision: HIPERFIT DSL Framework

3. Summary
1. The HIPERFIT Research Center
   - HIPERFIT Research Organisation
   - Motivation and Main Hypothesis
   - Research Themes and Areas

2. On Domain-Specific Languages in HIPERFIT
   - Domain-Specific Languages
   - ... for Economics and Finance
   - Vision: HIPERFIT DSL Framework

3. Summary
HIPERFIT Overview

HIPERFIT is a strategic research center funded by the Danish Council for Strategic Research (DSF) under grant no. 10-092299, founded in cooperation with the following partners from the financial industry: Danske Bank, Jyske Bank, LexiFi, Nordea, Nykredit Bank, and SimCorp.

Functional High-Performance Computing for Financial Information Technology
HIPERFIT Overview

HIPERFIT is a strategic research center funded by the Danish Council for Strategic Research (DSF) under grant no. 10-092299, founded in cooperation with the following partners from the financial industry: Danske Bank, Jyske Bank, LexiFi, Nordea, Nykredit Bank, and SimCorp.

- Funding volume: 5.8M €(43 Mio. DKK), 78% funding from DSF, 22% from partners and university (in kind)
- 6 PhD + 3 post-doctoral positions (CS and Mathematics)

- Mathematical Finance
- Domain-Specific Languages
- Parallel Functional Programming
- High-Performance Systems
Motivation: The Credit Crunch...

- Worldwide recession of 2008 (starting 2007)
- caused by dramatic price drop in the US house market.
Motivation: The Credit Crunch...

- **Subprime loans** - expensive credits to unreliable borrowers
- Accumulated to **CDOs** (collateral debt obligations) for **selling-on**
- The latter considered “secure” (AAA rating)
- Banks selling “garbage” CDOs to pension insurances later denied responsibility

Inadequate risk assessment (by rating agencies)
Ignoring interdependencies and risk of accumulated failure

...Market complexity beyond comprehension.
Motivation: The Credit Crunch...

- **Subprime loans** - expensive credits to unreliable borrowers
- accumulated to **CDOs** (collateral debt obligations) for **selling-on**
- The latter considered “secure” (AAA rating)
- Banks selling “garbage” CDOs to pension insurances later denied responsibility

**Inadequate risk assessment** (by rating agencies) **Ignoring** interdependencies and risk of **accumulated failure**

...Market complexity beyond comprehension.
Motivation: The Flash Crash…

- Dow Jones Index on May 6 2010

- Almost 10% drop within a few minutes
- Almost completely recovered again, only minutes later
- Systemic effect of algorithmic trading at very high volume(?)
Motivation: The Flash Crash...

- Dow Jones Index on May 6 2010

- Almost 10% drop within a few minutes
- Almost completely recovered again, only minutes later
- Systemic effect of algorithmic trading at very high volume (?)
Motivation: The Flash Crash...

- Dow Jones Index on May 6 2010

- Almost 10% drop within a few minutes
- Almost completely recovered again, only minutes later
- **Systemic effect of algorithmic trading** at very high volume(?)
Credit crunch, flash crash...

**Lack of systemic understanding** in a globalised economy.
Credit crunch, flash crash…

Lack of systemic understanding in a globalised economy. Complex models in other contexts: Large simulations, requiring large computing power.

Collective task of governments.
Transparency Requirements, Auditing, Regulations

- **Financial rating** needs more accuracy.
  - **Risk analysis** needs a larger scale,
  - **dependencies** have to be modelled.

- **Increasing regulation** for the financial industry.
  - **European banking supervision:** “Basel III” standard
    - increased capital requirements
    - Insolvency risk and rating **from large simulations**

- **New US Securities and Exchange Commission** Regulations
  - **Auditing requirement** for risk analysis and trading
Performance, Transparency, Expressiveness

This is where computer science enters the game...

- **Accuracy** of models (reliable results, auditing)
- **Performance** of computations (quick reactions, handling large data)
- **Ease** of development and maintenance (rapid and reliable development)

Choose any two?

Our Claim: Integrated solutions to achieve all three.
Performance, Transparency, Expressiveness

This is where computer science enters the game...

- **Accuracy** of models
  (reliable results, auditing)
- **Performance** of computations
  (quick reactions, handling large data)
- **Ease** of development and maintenance
  (rapid and reliable development)

Choose any two?
Our Claim: Integrated solutions to achieve all three.
Central Claim: Less is More!

- **Performance: Computing** more in less (time)!
  Applying domain-specific methodologies for parallel hardware.
  Capturing domain-specific parallelism in libraries and DSLs.

- **Transparency: Understanding** more from less (code)!
  Understanding the computation as a mathematical formula with clear semantics and controlled non-determinism.

Skip the indirection of imperative software architecture.
Do not sequentialise inherently parallel operations.

- **Productivity: Expressing** more with less (lines)!
  Writing high-level specifications instead of low-level code.
Central Claim: Less is More!

- **Performance**: **Computing** more in less (time)!
  Applying domain-specific methodologies for parallel hardware.
  Capturing domain-specific parallelism in libraries and DSLs.

- **Transparency**: **Understanding** more from less (code)!
  Understanding the computation as a mathematical formula with clear semantics and controlled non-determinism.

Skip the indirection of imperative software architecture.
Do not sequentialise inherently parallel operations.

- **Productivity**: **Expressing** more with less (lines)!
  Writing high-level specifications instead of low-level code.
Central Claim: Less is More!

- **Performance: Computing** more in less (time)!
  Applying domain-specific methodologies for parallel hardware.
  Capturing domain-specific parallelism in libraries and DSLs.

- **Transparency: Understanding** more from less (code)!
  Understanding the computation as a mathematical formula
  with clear semantics and controlled non-determinism.

Skip the indirection of imperative software architecture.
Do not sequentialise inherently parallel operations.

- **Productivity: Expressing** more with less (lines)!
  Writing high-level specifications instead of low-level code.
Research Themes and Areas in HIPERFIT

Research Areas

Mathematical Finance
- Risk Scenarios
- Model specification
- Financial information specification
- Extracting parallelism
- High-performance backends

Domain-Specific Languages

Functional Programming

High-Performance Systems
Research Themes and Areas in HIPERFIT

Research Areas

- Mathematical Finance
- Domain-Specific Languages

Research Themes

- Risk Scenarios
- Model specification
- Financial information specification
- Extracting parallelism
- High-performance backends

ML
DSL
FP
HPS

Department of Computer Science
1 The HIPERFIT Research Center
   • HIPERFIT Research Organisation
   • Motivation and Main Hypothesis
   • Research Themes and Areas

2 On Domain-Specific Languages in HIPERFIT
   • Domain-Specific Languages
   • ... for Economics and Finance
   • Vision: HIPERFIT DSL Framework

3 Summary
What Makes a DSL Special...

In our opinion, FIDO is a compelling example of a domain-specific language. It is focused on a clearly defined and narrow domain: formulas in monadic second-order logic or, equivalently, automata on large alphabets. It offers solutions to a classical software problem: drowning in a swamp of low-level encodings. It advocates a simple design principle: go by analogy to standard programming language concepts. It uses a well-known and trusted technology: all the phases of a standard compiler, including optimizations at all levels. It provides unique benefits that cannot be matched by a library in a standard programming language: notational conveniences, type checking, and global optimizations. And during its development, we discovered new insights about the domain: new notions of tree automata and algorithms.

What Makes a DSL Special…

A tailor-made language for experts,²

- **Domain-specific**, with a limited purpose,
- providing a **concise notation**, 
- using **expert vocabulary and abstractions**.

…providing **type-safety** and **restricted expressivity**, 

- **Semantic checks** disallow nonsensical content, 
- language constructs are added **as and when required**.

…**automating** common tasks and data structures.

What Makes a DSL Special... 

A **tailor-made language** for experts,\(^2\)

- **Domain-specific**, with a limited purpose,
- providing a **concise notation**, 
- using **expert vocabulary and abstractions**.

...providing **type-safety** and **restricted expressivity**, 
- **Semantic checks** disallow nonsensical content, 
- language constructs are added **as and when required**.

...**automating** common tasks and data structures.

What Makes a DSL Special... 

A **tailor-made language** for experts,\(^2\)

- **Domain-specific**, with a limited purpose,
- providing a **concise notation**, 
- using **expert vocabulary and abstractions**.

...providing **type-safety** and **restricted expressivity**, 
- **Semantic checks** disallow nonsensical content,
- language constructs are added **as and when required**.

...**automating** common **tasks** and **data structures**.

---

DSL Design Goals and Principles

**Capture** the *compositional structure* of a domain model

- **Isomorphism Principle**, Correspondence of formal (DSL)specification to informal description

- **Small requirements change** = small specification change

DSL specifications have dual nature:

- A specification can be executed as a program, following a standard semantics.
- Specifications can be analysed as data by processing tools. *Multiple* (open-ended) interpretations usually exist.
DSL Design Goals and Principles

Capture the compositional structure of a domain model

- **Isomorphism Principle**, Correspondence of formal (DSL) specification to informal description

- Small requirements change = small specification change

DSL specifications have dual nature:

- A specification can be executed as a program, following a standard semantics.

- Specifications can be analysed as data by processing tools. Multiple (open-ended) interpretations usually exist.
Past Work on DSLs in Economic Contexts

- **Project 3gERP for Enterprise Resource Planning (ERP)**
  - **POETS** (Process-Oriented Event-driven Transaction System)
  - Includes a *compositional contract language* (DSL).
  - **Contract handling** based on the *standardised format*, translations and other modifications programmatically done using *tools*.
- **Reporting Language**: *FunSETL* (Functional Set-oriented Language)
  - **No recursion or loops!** Only iteration over set elements.
  - Guarantees termination, allows for optimising transformations.

Performance *in the black box* (not a user responsibility).
Finance: A DSL for Financial Contracts

Example: “Zero-coupon bonds”

\[ p_{12} = \text{zcb (date "1 Jun 2012") 100 Pounds :: Contract} \]
\[ e_{12} = \text{zcb (date "1 Jun 2012") 115 Euro :: Contract} \]
\[ p_{13} = \text{zcb (date "1 Jul 2012") 100 Pounds :: Contract} \]

\[ c = (p_{12} \ 'or' \ e_{12}) \ 'and' \ give \ p_{13} \]

Used today in a number of banks: similar in-house languages

How “valuable” is the above opportunity?

Dependencies: Interest rate in June, exchange rate in June and July

Central question: Contract valuation semantics

Finance: A DSL for Financial Contracts

Example: “Zero-coupon bonds”

\[
\begin{align*}
p_{12} &= \text{zcb} \ (\text{date "1 Jun 2012"}) \ 100 \ \text{Pounds} :: \ \text{Contract} \\
e_{12} &= \text{zcb} \ (\text{date "1 Jun 2012"}) \ 115 \ \text{Euro} :: \ \text{Contract} \\
p_{13} &= \text{zcb} \ (\text{date "1 Jul 2012"}) \ 100 \ \text{Pounds} :: \ \text{Contract} \\
c &= (p_{12} \ 'or' \ e_{12}) \ 'and' \ \text{give} \ p_{13}
\end{align*}
\]

Used today in a number of banks: similar in-house languages

How “valuable” is the above opportunity?

Dependencies: Interest rate in June, exchange rate in June and July

Central question: Contract valuation semantics

Contracts Design, Management and Analysis

Example: “American Put Option”

\[
\text{americanPut} \ (t1, t2) \ n \ stk \ str = \text{anytime} \ (between \ t1 \ t2) \ \text{sale} \\
\text{where} \ \text{sale} = \text{give} \ (\text{buyStkUSD} \ n \ stk \ str) \\
\text{buyStkUSD} \ n \ stk \ str = \text{give} \ (\text{scale} \ (\text{konst} \ (n*str)) \ (\text{one USD})) \\
\text{‘and’ scale} \ (\text{konst} \ n) \ (\text{one stk})
\]

- Designing complex financial products
- Managing and scheduling contracts
- Valuation (“pricing”) of a contract
- Risk management for institutions
- Legal reporting requirements and auditing

MLFi language by Lexifi: whole product suite for contract management
## DSL Potential for Valuation and Risk

### Example: “American Put Option”

```plaintext
americanPut (t1, t2) n stk str = anytime (between t1 t2) sale
   where sale = give (buyStkUSD n stk str)
buyStkUSD n stk str = give (scale (konst (n*str)) (one USD))
   'and' scale (konst n) (one stk)
```

- **Execution strategy**: exercise option if price good
- **Scenario Analysis**: pay-off for particular scenarios
- **Stochastic rate models**, Brownian motion and other methods

- **Model-based Analysis**: expected pay-off and variation
DSL Potential for Valuation and Risk

Example: “American Put Option”

\[
\text{americanPut} \ (t1, \ t2) \ n \ stk \ str = \ \text{anytime} \ (\text{between} \ t1 \ t2) \ \text{sale}
\]
where
\[
\text{sale} = \ \text{give} \ (\text{buyStkUSD} \ n \ stk \ str)
\]
\[
\text{buyStkUSD} \ n \ stk \ str = \ \text{give} \ (\text{scale} \ (\text{konst} \ (n*str)) \ (\text{one USD})) \ \text{‘and’ scale} \ (\text{konst} \ n) \ (\text{one stk})
\]

- **Execution strategy**: exercise option if price good **No DSLs**
- **Scenario Analysis**: pay-off for particular scenarios **No DSLs**
- **Stochastic rate models**, Brownian motion and other methods
  **No (programmatic scenario generation)**
- **Model-based Analysis**: expected pay-off and variation
  PDEs and closed forms where possible, otherwise ad-hoc
Vision: A DSL Framework for Valuation

Financial products usually described using DSLs
Pricing models usually not.

- **DSL framework** for valuation
- Describe **stochastic models** for observables using suitable DSLs
- Two **inputs** to a **pricing engine**: Numeric methods for solving stochastic models

Pricing Engine: **tailored functional language** for easy parallelisation.
Massively **increased productivity** for model development.
Meanwhile, others also think about languages...

SECURITIES AND EXCHANGE COMMISSION

17 CFR Parts 200, 229, 230, 232, 239, 240, 243 and 249

Release Nos. 33-9117; 34-61858; File No. S7-08-10

RIN 3235-AK37

ASSET-BACKED SECURITIES

AGENCY: Securities and Exchange Commission

ACTION: Proposed rule.

SUMMARY: We are proposing significant revisions to Regulation AB and other rules regarding the offering process, disclosure and reporting for asset-backed securities. Our proposals would revise the guidelines for ABS offerings to provide investors with more detailed and granular information about the underlying assets for asset-backed securities. This additional information would be provided according to proposed standards and in a tagged data format using eXtensible Markup Language (XML). In addition, we are proposing to require, along with the prospectus filing, the filing of a computer program of the contractual cash flow provisions expressed as downloadable source code in Python, a
Meanwhile, others also think about languages . . .

[US] Securities and Exchange Commission

SECURITIES AND EXCHANGE COMMISSION

17 CFR Parts 200, 229, 230, 232, 239, 240, 243 and 249

Release Nos. 33-9117; 34-61858; File No. S7-08-10

RIN 3235-AK37

ASSET-BACKED SECURITIES

AGENCY: Securities and Exchange Commission

ACTION: Proposed rule.

SUMMARY: We are proposing significant revisions to Regulation AB and other rules regarding the offering process, disclosure and reporting for asset-backed securities. Our proposals would revise the technical rules for ABS filings to provide investors with more information about the characteristics of the underlying collateral. The information would be provided according to proposed standards and in a tagged data format using eXtensible Markup Language (XML). In addition, we are proposing to require, along with the prospectus filing, the filing of a computer program of the contractual cash flow provisions expressed as downloadable source code in Python, a
Meanwhile, others also think about languages...  

[US] Securities and Exchange Commission

SECURITIES AND EXCHANGE COMMISSION

17 CFR Parts 200, 229, 230, 232, 239, 240, 243 and 249

Release Nos. 33-9117; 34-61858; File No. S7-08-10

RIN 3235-AK37

ASSET-BACKED SECURITIES

AGENCY: Securities and Exchange Commission

ACTION: Proposed rule.

SUMMARY: We are proposing significant revisions to Regulation AB and other rules regarding the offering process, disclosure and reporting for asset-backed securities. Our proposals would revise the definitions for ABS offerings to provide investors with more transparency regarding the underlying assets. The information would be provided according to proposed standards and in a tagged data format using eXtensible Markup Language (XML). In addition, we are proposing to require, along with the prospectus filing, the filing of a computer program of the contractual cash flow provisions expressed as downloadable source code in Python, a...
1 The HIPERFIT Research Center
   - HIPERFIT Research Organisation
   - Motivation and Main Hypothesis
   - Research Themes and Areas

2 On Domain-Specific Languages in HIPERFIT
   - Domain-Specific Languages
   - ... for Economics and Finance
   - Vision: HIPERFIT DSL Framework

3 Summary
Summary

**HIPERFIT**: Apply the “Less is more” paradigm, integrated solution

**Research Directions:**
- Advanced *mathematical methods* addressing finance
- Domain-specific *languages* for models and entities
- Modern programming *language and compilation technology*
- Parallel functional programming to use modern massively parallel hardware

Potential for a *domain-specific language* showcase.

http://hiperfit.dk