THE USE OF APL IN SIMCORP DIMENSION

NIELS HALLENBERG
RELEASE TRAIN ENGINEER, DIRECTOR
DYALOG ’16, GLASGOW
THE WORLD’S LEADING PROVIDER
BRINGING INTEGRATED SOLUTIONS TO INVESTMENT MANAGERS

• Servicing the global buy-side market
• Established in 1971
• 1250+ Employees worldwide
• 250+ Developers
• 2 Releases per Year
• Headquartered in Copenhagen, Denmark
• Listed on Copenhagen Nasdaq
• USD 19 trillion managed on SimCorp Dimension®
THE WORLD’S LEADING PROVIDER
BRINGING INTEGRATED SOLUTIONS TO INVESTMENT MANAGERS

MORE THAN 20 OFFICES GLOBALLY
MORE THAN 20% OF REVENUE INVESTED IN R&D
MORE THAN 16,000 USERS
MORE THAN 40 YEARS OF DEVELOPMENT
MORE THAN 170 CLIENTS WORLDWIDE
PEOPLE IN DEVELOPMENT
MANY COMPETENCES WORK TOGETHER

Investment Management Software requires many skills to develop, e.g.:

- Math, Physics, Statistics
- Finance
- Computer Science
- UX Designers
- QA Analysts

SimCorp implements the Agile Development model called Scaled Agile Framework (SAFe).
# CURRENT TECHNOLOGY STACK

LOTS OF CODE, MANY THIRD PARTY LIGRARIES, BIG NUMBERS, LOTS OF DATA

<table>
<thead>
<tr>
<th>Language</th>
<th>#Lines</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>APL</td>
<td>2,500,000</td>
<td>1971</td>
</tr>
<tr>
<td>C#</td>
<td>5,000,000</td>
<td>2004</td>
</tr>
<tr>
<td>C++</td>
<td>1,000,000</td>
<td>&lt;1990</td>
</tr>
<tr>
<td>Ocaml</td>
<td>1,000,000</td>
<td>2000</td>
</tr>
</tbody>
</table>

Additional languages such as F#, Java, Python, Powershell, HTML5 web front end etc. are also used.

Dyalog APL integrates well with Windows

A large number of 3rd party libraries.

Source: http://informationisbeautiful.net
APL is fairly easy to start using when you do not have a computer science background. Especially if you have a mathematical background.

C# was introduced to make modern multi threaded user interfaces with online calculations.

OCaml was introduced to accommodate for a combinator library (DSL) for financial contracts.

C++ is used for several low level high performing components.

HTML5 front end stack for web and mobile.

Cloud as future platform
WHY IS APL COST EFFICIENT FOR SIMCORP
MANY YEARS OF FRAMEWORK AND ABSTRACTION

An extensive and easy to use APL framework.

UI rendering, data synchronization, auditing, logging, authorization, form layout, etc. are abstracted in highly efficient framework code.

Developers, with math background, can focus on business logic. Operations on arrays are well understood abstractions.

Tracing and Debugging is well supported; even at customers.

Code running today will also run in 20 years.
DIFFERENT TECHNOLOGIES ARE VISIBLE
APL, C#, OCAML, HTML5 - WHICH IS OK FROM A CUSTOMER PERSPECTIVE
Enhancements are 30% new development and 70% extension of existing functionality. Abstraction is key

Generation Y and Z.

Increased demands and Moore’s law dictates more parallelism. Is C# legacy? Technology driven

Declarative languages are gaining momentum. Mutable data and state is problematic

Cloud as your platform. Static type safety is valued with millions of lines of code. Dyalog’14 presentation SimCorp continues the effort.

When 64 bit floats are not enough Lifespan of modern technology

See nbim.no

Web technologies comes and go

Java got Scala C# got F#
APL GOING FORWARD, PART I

TOUGH DECISIONS

APL has been kept modern and feature rich by Dyalog. Absolutely great and important work

Language centric extensions:
  Objects and classes are already available

  What about optional type information to create a static safety net? It doesn’t have to be perfect.

  What about type safe declarative extensions? Imagine algebraic data types and higher order functions

  What about high precision numbers and large integers? Dyalog APL can be first in class.

Compilation is important

Parallization Make Futures & Isolates native part of interpreter. Include GPU’s & optimize, optimize, optimize, …
APL GOING FORWARD, PART II

TOUGH DECISIONS

Language centric extensions:
- Memory efficiency is key to enterprise systems. (Continue optimizing the External Workspace concept)

Keep going on smaller optimizations. (Constants, TimeStamps, …)

Multiple APL instances (Shared R/W caches and External Workspace)

IDE (Great work by opening up such that Visual Studio plugin’s etc. can be made – because future will bring more and more technologies together)

Cloud (Service orientation and Elasticity is key)

**Dyalog APL is a great tool of thought.**

Now we just need support for multi-processors, big data volumes, well known IDE and a nice fit with cloud platforms like Azure and AWS.
Congratulations with the 50 years

QUESTIONS?
SimCorp
The contents of this presentation are for general information and illustrative purposes only and are used at the reader’s own risk. SimCorp uses all reasonable endeavours to ensure the accuracy of the information.

However, SimCorp does not guarantee or warrant the accuracy, completeness, factual correctness, or reliability of any information in this publication and does not accept liability for errors, omissions, inaccuracies, or typographical errors.

The views and opinions expressed in this publication are not necessarily those of SimCorp.

© 2016 SimCorp A/S. All rights reserved. Without limiting rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form, by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose without the express written permission of SimCorp A/S.