

GigaSystemBuilder: Model-driven GigaSpaces

Matthew Fowler
NT/e

Agenda

1. NT/e and JeeWiz
2. Introduction - Last Two Challenges
3. Why - use model-driven approach
4. Who - should use it
5. How - does it work
6. What - does it produce; and what not
7. When - is it available

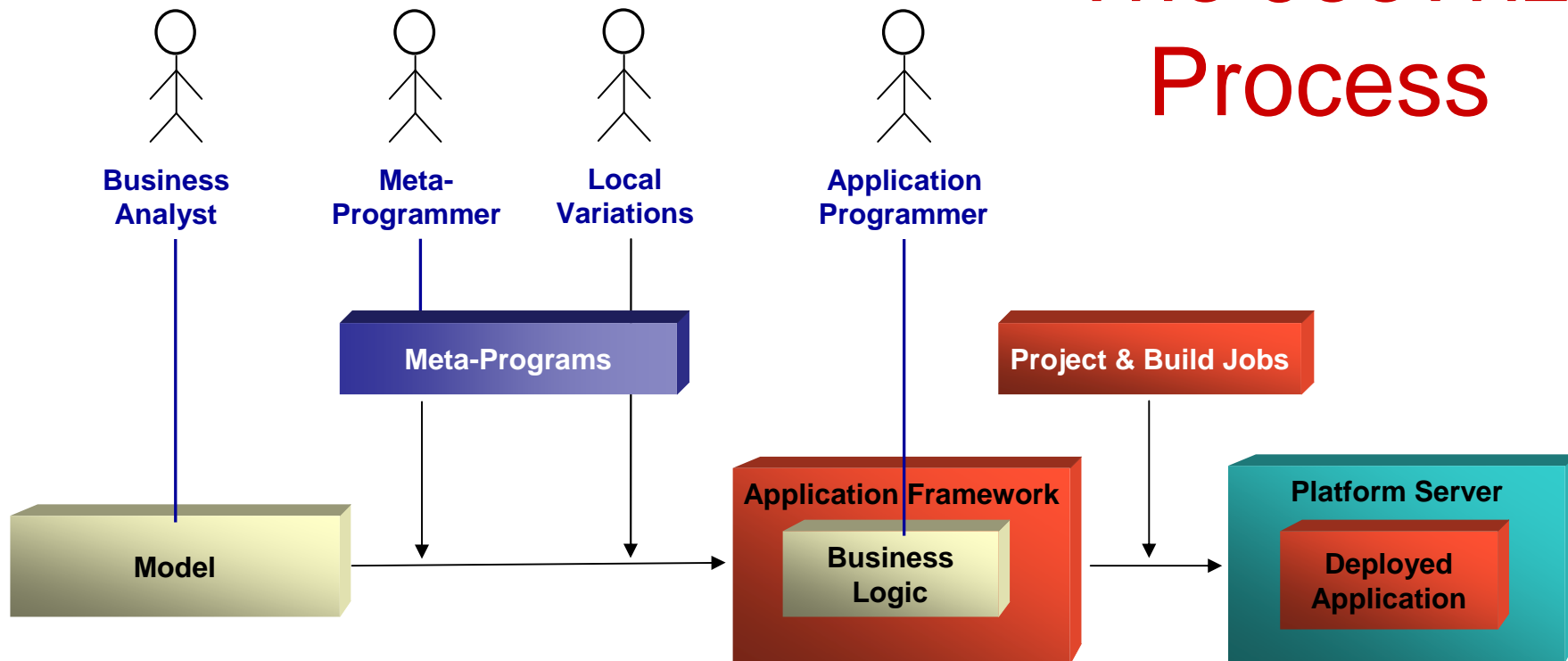
NT/e

- NT/e - New Technology / *enterprise*
- Infrastructure / Architecture Consultancy
- JeeWiz - www.jeewiz.org, www.jeewiz.com
 - building software factories, system generation
- £2.5m turnover, £100k profit
- Key relationships: DeutschePost, UBS, BEA/WebLogic, GigaSpaces, Jive Clearspace

JeeWiz Background

- BEA: "Stop customers messing up architecture"
- Most powerful system generation engine
 - Java, XML, Ant, Velocity
 - Special features for building "meta-components"
- Engine is open-source - jeewiz.org
- Large-scale generation systems, end-to-end:
 - Struts, mid-tier views, J2EE
 - Trinidad, JSF, Spring, Hibernate
 - "Payment on sale"; otherwise free

The JeeWiz Process



- Up to 97% generation in Struts/J2EE
- Cost-time savings

Introduction

- Essential difference in operation between notify and polling containers
- What's wrong with this configuration snippet:



```
<bean id="offerNormalizer" class="com.gigaspace.common.offerNormalizer" />
<bean id="matcher" class="com.gigaspace.Matcher" />
<os-remoting:service-exporter id="matcherServiceExporter">
  <os-remoting:service ref="matcher" />
</os-remoting:service-exporter>
<os-core:space id="trainingSpace" url="/./TrainingSpace">
  <os-core:method-adapter-filter before-write="DisplayOffer">
    <os-core:filter ref="offerNormalizer" />
  </os-core:method-adapter-filter>
  <os-core:filter-provider ref="matcherServiceExporter" />
</os-core:space>
```

* It's the feeling that counts!

Why ... do "GigaSystemBuilder"?

- GigaSpaces
 - reduces "platform feeding" for developers
 - solves some tricky problems elegantly
 - e.g. scalability and SLA container
 - combined with Azul, Java addresses large-scale, high-performance applications
 - ▶▶ potential for breakthrough product
- No modelling/generation capability...
 - capture the essence of event-driven computing

Why ... Model-Driven System Gen?

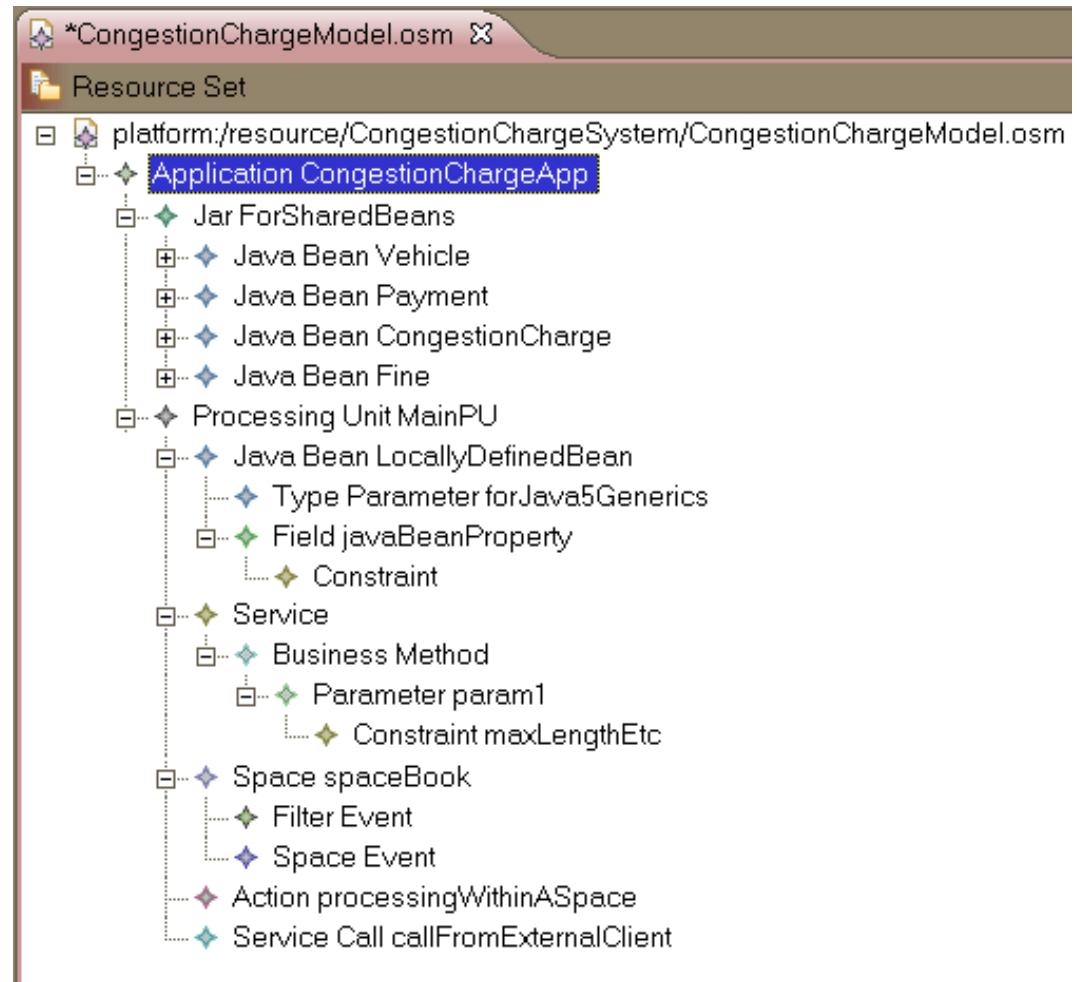
- Platform-specific knowledge (e.g. notify-containers) encoded
 - Use generic concepts - communicate easier
 - Get going quicker, painless Eclipse+GS setup
- One change in intent, multiple changes in system
 - Faster, surer development
- Tweak the architecture late in the project
 - Reduce risk/rigidity

Who ... is it for?

- Architects / Developers today
 - Tactical: no changes in development process
 - Maybe later
 - modellers/analysts
 - deployment/operations
- Development and maintenance
 - Iterate the model: NOT A WIZARD

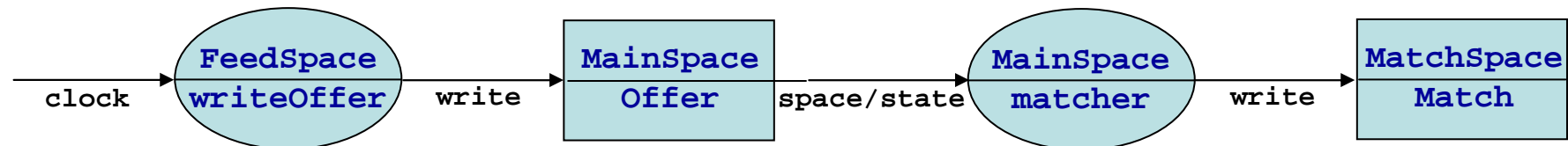
How ... 1. The Model

- Shows available concepts



2. Modelling the Dynamics of Grids

- Basics: Data + Methods ... + Spaces + Events
- Events:
 - space/lifecycle; space/state; service call; clock
 - Implies the main input to a business method
- Business methods / 'Actions':
 - Main inputs and outputs: space+class
 - Subsidiary inputs and outputs: same
- Use cases == events → action → space entry.

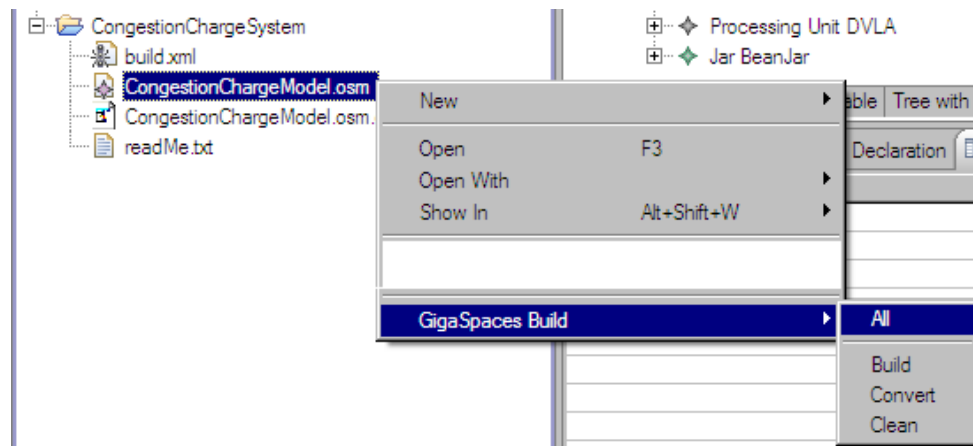


3. What's missing

- Business Logic code
- Containers - notify/polling
- Bean references in Spring
- ISpaceFilterEntry
- etc. etc.

4. Build

- Operations from Eclipse:



What ... is generated?

- Projects and config
 - Within Eclipse
 - External Ant builds (use your own editor)
- Spring config, all in pu.xml
- Java classes to hold business methods
- Constraints

Adding Business Logic

- Implementation methods in special class
 - Tagged with "uid" to support name changes

```
package com.cc.zone;

import java.util.*;
/* BEGIN: EXTRA IMPORTS */
/* END: EXTRA IMPORTS */

class CongestionZoneProcessorHelper {
    java.util.logging.Logger log = java.util.logging.Logger.getLogger( "com.cc.zone.CongestionZoneProces

    private static final String __className = "com.cc.zone.CongestionZoneProcessorHelper";

    /* uid: ProcessVehicle_bm */
    Fine processVehicle( final Vehicle vehicle, final GigaSpace gigaSpace, final TransactionStatus trans
        /* Add your code here */
        return null;
    }

    /* uid: ProcessCharge_bm */
    void processCharge( final CongestionCharge congestionCharge, final GigaSpace gigaSpace, final Transa
        /* Add your code here */
    }
```

When ... is it available?

- Version 1.0Beta available now - www.jeewiz.org
 - XML-style editor
 - Royalty for commercial sales of generated systems
 - Other uses (in-house, contract) no charge
- Futures:
 - Graphical editing of use-cases
 - Smarter checks - e.g. race states on events
 - More high-level styles and best practice
 - Deployment options
 - ? Business rules

Benefits

- Faster POCs
- Add detailed knowledge
 - technical details of configuration
 - no need to know details of containers
- Faster and surer to change
- Decouple
 - application structure and intent in model
 - business logic in Java
 - know-how/best practice in meta-programs

End