Hummingbird
An Open Source Ground Segment for Small Satellites

A true story of doing things different

GSAW 2012
Gert Villemos (Villemos Solutions)
Mark Doyle (Logica) and Johannes Klug (Logica)
What is 'Hummingbird'?

- An open source software framework (Apache License)
- For building ground segment systems for small satellites
- A 'back to basics' approach
- Using simplicity as a design principle
- Pushing as much functionality as possible to existing technologies

www.hbird.org and facebook

© 2012 by Villemos Solutions and Logica. Published by The Aerospace Corporation with permission.
The world

5 years...

8 years...

© 2012 by Villemos Solutions and Logica. Published by The Aerospace Corporation with permission.
The world we live in
Business Plan

2.0

© 2012 by Villemos Solutions and Logica. Published by The Aerospace Corporation with permission.
Back to Basics
Thesis

- Ground segment systems for satellites is no longer special; we move data from A to B

- Modern network technologies can be used and are better. Let's stop reinventing the wheel

- Complexity is inherited (... and often cultural) and propagates through our systems. It is the root of all evil

- Find the root course and remove it. Ground segment system can be really simple

© 2012 by Villemos Solutions and Logica. Published by The Aerospace Corporation with permission.
'New' Implementation Concept

This is hardly brain surgery!
Technology Stack (fantastic four)

- Spring
- Camel
- ActiveMQ
- Cometd

... but components in other languages can be integrated

... and yes, its all Java
Evolution

- **Hummingbird 0.1**
  - Classical separation into tiers (transport, business, presentation)
  - CCSDS stack Frames → Packets → Parameters
  - Distribution of predefined types
  - Centered on centralized ‘System Model’ (think runtime XTCE)

- **Hummingbird 0.2**
  - CCSDS stack ‘banned’ to transport tier, fully encapsulated
  - True asynchronous processing
  - Semantic information model, non-relational databases
  - Distribution of ‘what-ever’ with plugable services
System Integration
Demand complete rethinking of processing model. An architectural driver.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Value</td>
<td></td>
</tr>
<tr>
<td>Eng Value</td>
<td></td>
</tr>
<tr>
<td>State A</td>
<td></td>
</tr>
<tr>
<td>State B</td>
<td></td>
</tr>
<tr>
<td>State C</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State X</th>
<th>Description</th>
</tr>
</thead>
</table>

Extendible

Scalable

Non-normalized

withRaw

stateOf
Limit Check Service
Consumes: Parameter message
Emits: State Parameter
Description: The limit check service takes as input a parameter, checks its limit and emits a state parameter corresponding to the limit with value ‘true’ (in limit) or ‘false’ (out of limit).

Heartbeat Service
Consumes: None
Emits: Heartbeat
Description: Issues a ‘alive’ message at intervals.
First Flight

➢ Comming up
  ➢ EstCube (University of Estland)
  ➢ Strand (Surrey Satellites)
  ➢ TechDemoSat-1 (UK Space Agency)
Initial Results

Having implemented guestimated 10% of the functionality of similar existing systems

- Commanding (scheduling, pre release validation, release, verification and history)
- Monitoring (parameter creation, limit check, consistency check, calibration and history)
- Orbit and contact prediction

- New code base 1k LoC
- Projected code base 10k LoC
- Existing code base ~1m Lines of Code (LoC)
Value

- Highly motivated staff, learning by doing
- Concepts feedback into winning ‘normal’ work
- Door opener to new, and frequently unexpected, markets
- Great PR
- Neither predictable nor quantifiable but very real
Business Model

- Open Source Core
- Enterprise Edition
  - Liability
  - Tested
  - Proven
  - Documented
  - Evolution plan
  - Ahead of the curve
- Services and added value products
Conclusions

- Nothing is more convincing than to stop talking and start doing
- Standard technologies (of course) work for space data systems
- System integration can be very simple, we make it complex
- 'Managing' an open source community is not simple
Read more at: www.hbird.org and facebook
Business Tier

Spring Component Assembly

Camel Integration

ActiveMQ:topic:Messages

CometD Streaming

Browser

Services

© 2012 by Villemos Solutions and Logica. Published by The Aerospace Corporation with permission.
WARNING: Source code ahead!

... the message is not in the code itself, but in the changes to the code.
public class Manager implements IManager {
    protected IWorker worker = new Worker();
    public void manage() {
        worker.work("1:2:3");
    }
}

public class Worker implements IWorker {
    protected IPublisher publisher = new Publisher();
    public void work(String values) {
        String[] elements = values.split(":");
        for (String element : elements) {
            long value = Long.parseLong(element);
            publisher.display(value);
        }
    }
}

public class Publisher implements IPublisher {
    public void display(long value) {
        System.out.println("Test value: "+ value);
    }
}
public class Manager implements IManager {
    protected IWorker worker = new Worker();
    public void manage() {
        worker.work("1:2:3");
    }
}

public class Worker implements IWorker {
    protected IPublisher publisher = new Publisher();
    public void work(String values) {
        String[] elements = values.split(":");
        for (String element : elements) {
            long value = Long.parseLong(element);
            publisher.display(value);
        }
    }
}

public class Publisher implements IPublisher {
    public void display(long value) {
        System.out.println("Test value: "+value);
    }
}
The Modern Code Base

```java
public class Manager implements IManager {
    protected IWorker worker = new Worker();
    public void manage() {
        worker.work("1:2:3");
    }
}

public class Worker implements IWorker {
    protected IPublisher publisher = new Publisher();
    public void work(String values) {
        String[] elements = values.split(":");
        for (String element : elements) {
            long value = Long.parseLong(element);
            publisher.display(value);
        }
    }
}

public class Publisher implements IPublisher {
    public void display(long value) {
        System.out.println("Test value: "+value);
    }
}
```

```xml
<bean id="manager" class="foo.Manager"/>
<bean id="worker" class="foo.Worker"/>
<bean id="publisher" class="foo.Publisher"/>
<route>
    <from uri="timer://foo?period=60000"/>
    <to uri="bean:worker"/>
    <to uri="bean:manager"/>
    <to uri="bean:publisher"/>
</route>
```
The Modern Code Base

```java
public class Manager implements IManager {
    protected IWorker worker = new Worker();
    public void manage() {
        worker.work("1:2:3");
    }
}
```

```java
public class Worker implements IWorker {
    protected IPublisher publisher = new Publisher();
    public void work(String values) {
        String[] elements = values.split(":");
        for (String element : elements) {
            long value = Long.parseLong(element);
            publisher.display(value);
        }
    }
}
```

```java
public class Publisher implements IPublisher {
    public void display(long value) {
        System.out.println("Test value: "+ value);
    }
}
```

```xml
<bean id="manager" class="foo.Manager"/>
<bean id="worker" class="foo.Worker"/>
<bean id="publisher" class="foo.Publisher"/>
<route>
    <from uri="timer://foo?period=60000"/>
    <to uri="bean:manager"/>
    <split>
        <method bean="worker"/>
        <to uri="bean:publisher"/>
    </split>
</route>
```

© 2012 by Villemos Solutions and Logica. Published by The Aerospace Corporation with permission.
The Modern Code Base

public class Manager {
    public String manage() {
        return "1:2:3";
    }
}

public class Worker {
    public String[] work(String values) {
        return values.split(".");
    }
}

public class Publisher {
    public void display(long value) {
        System.out.println("Test value: " + value);
    }
}

<bean id="manager" class="foo.Manager"/>
<bean id="worker" class="foo.Worker"/>
<bean id="publisher" class="foo.Publisher"/>

<route>
    <from uri="timer://foo?period=60000"/>
    <to uri="bean:manager"/>
    <split>
        <method bean="worker"/>
        <to uri="bean:publisher"/>
    </split>
</route>

Massive code base reduction

Only business logic in the code, routing in the configuration

Fantastic code metrics (no coupling)

100% test coverage easy to reach

© 2012 by Villemos Solutions and Logica. Published by The Aerospace Corporation with permission.