I CHEATED ON EMF WITH RDF.
AND I MAY DO IT AGAIN!
DISCLAIMERS

• The opinions and ideas expressed here are my own, and do not reflect the views of my current and previous employers.

• All the images used on this presentation are available from the Internet and, to the best of my knowledge, are allowed to be used in this context.
EVERYONE IS ENTITLED TO MY OPINION
Ecore Model for Purchase Orders

is represented in Ecore as

```
EClass
(name="PurchaseOrder")
```

```
EClass
(name="Item")
```

```
EAttribute
(name="shipTo")
```

```
EAttribute
(name="billTo")
```

```
EAttribute
(name="items")
```

```
EAttribute
(name="productName")
```

HELLO
MY NAME IS
RDF
H24

Water

Flavored Water

Enhanced Water

Premium Water


Figure 1: An RDF Graph Describing Eric Miller

Figure 1 illustrates that RDF uses URIs to identify:

- individuals, e.g., Eric Miller, identified by http://www.w3.org/People/EM/contact#me
- kinds of things, e.g., Person, identified by http://www.w3.org/2000/10/swa/vb/pim/contact#Person
- properties of these things, e.g., mailbox, identified by http://www.w3.org/2000/10/swa/vb/pim/contact#mailbox
- values of those properties, e.g.,mailto:em@w3.org as the value of the mailbox property (RDF also uses character strings such as "Eric Miller", and values from other datatypes such as integers and dates, as the values of properties)

RDF also provides an XML-based syntax (called RDF/XML) for recording and exchanging these graphs. Example 1 is a small chunk of RDF in RDF/XML corresponding to the graph in Figure 1:

Example 1: RDF/XML Describing Eric Miller

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:contact="http://www.w3.org/2000/10/swa/pim/contact#">
  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```
\[ R_{ik} = 0 \]
```java
package org.example.library

class Library {
    String name
    contains Book [] books
    contains Writer [] authors
    op Book getBook(String title) {
        for (book : books) {
            if (title.equals(book.title)) {
                return book
            }
        }
        return null
    }
}

class Book {
    String title
    int pages
    Category category
    refers Writer author opposite books
}

class Writer {
    String name
    refers Book [] books opposite author
}

enum Category {
    Mystery
    ScienceFiction
    Biography
}
```
package org.example.library

class Library {
    String name
    contains Book [] books
    contains Writer [] authors
    op Book getBook(String title) {
        for (book : books) {
            if (title.equals(book.title)) {
                return book
            }
        }
        return null
    }
}

class Book {
    String title
    int pages
    Category category
    refers Writer author opposite books
}

class Writer {
    String name
    refers Book [] books opposite author
}

class Category {
    Mystery
    ScienceFiction
    Biography
}
package org.example.library

class Library {
   String name
   contains Book [] books
   contains Writer [] authors

   op Book getBook(String title) {
      for (book : books) {
         if (title.equals(book.title)) {
            return book
         }
      }
      return null
   }
}

class Book {
   String title
   int pages
   Category category
   refers Writer author opposite books
}

class Writer {
   String name
   refers Book [] books opposite author
}

class Category {
   Mystery
   ScienceFiction
   Biography
}

Library Manager 201006101454
package org.example.library

class Library {
    String name
    contains Book [] books
    contains Writer [] authors
    op Book getBook(String title) {
        for (book : books) {
            if (title.equals(book.title)) {
                return book
            }
        }
        return null
    }
}

class Book {
    String title
    int pages
    Category category
    refers Writer author opposite books
}
class Writer {
    String name
    refers Book [] books opposite author
}
enum Category {
    Mystery
    ScienceFiction
    Biography
}
package org.example.library

class Library {
    String name
    contains Book [] books
    contains Writer [] authors
    op Book getBook(String title) {
        for (book : books) {
            if (title.equals(book.title)) {
                return book
            }
        }
        return null
    }
}

class Book {
    String title
    int pages
    Category category
    refers Writer author opposite books
}

class Writer {
    String name
    refers Book [] books opposite author
}

class Category {
    Mystery
    Science
    Fiction
    Biography
}
package org.example.library

class Library {
    String name
    contains Book [] books
    contains Writer [] authors
    Book getBook(String title) {
        for (book : books) {
            if (title.equals(book.title)) {
                return book
            }
        }
        return null
    }
}

class Book {
    String title
    int pages
    Category category
    refers Writer author opposite books
}

class Writer {
    String name
    refers Book [] books opposite author
}

enum Category {
    Mystery
    Science
    Fiction
    Biography
}
Dave
Wednesday, 11 April, 12
<isbn:123>

<title>
"EMF"

</title>

"EMF"

^^xsd:string
“The model of a model is a model”
Let's Recap
CHANGE AHEAD
THANK YOU!
Submit Feedback

Session: *
I cheated on EMF with RDF. And I may do it again!

Overall evaluation of this session:
-1 0 +1

Additional comments and suggestions:

Save