J2EE, XML and one step further: cool4us, good for Henkel

Dr. Falk Langhammer and
Dr. Joachim Buth

Living Pages Research GmbH
Munich, Germany
www.living-pages.de

Net.ObjectDays 2003, Erfurt, Germany

Table of Contents

(1) The Henkel Challenge
(2) Demo (ercato)
(3) ercatons
(4) The ercato programming model

The Henkel challenge:
Perfume development at Henkel KGaA, Düsseldorf

- Make perfume designers happy (a kind of creative diva)

AND

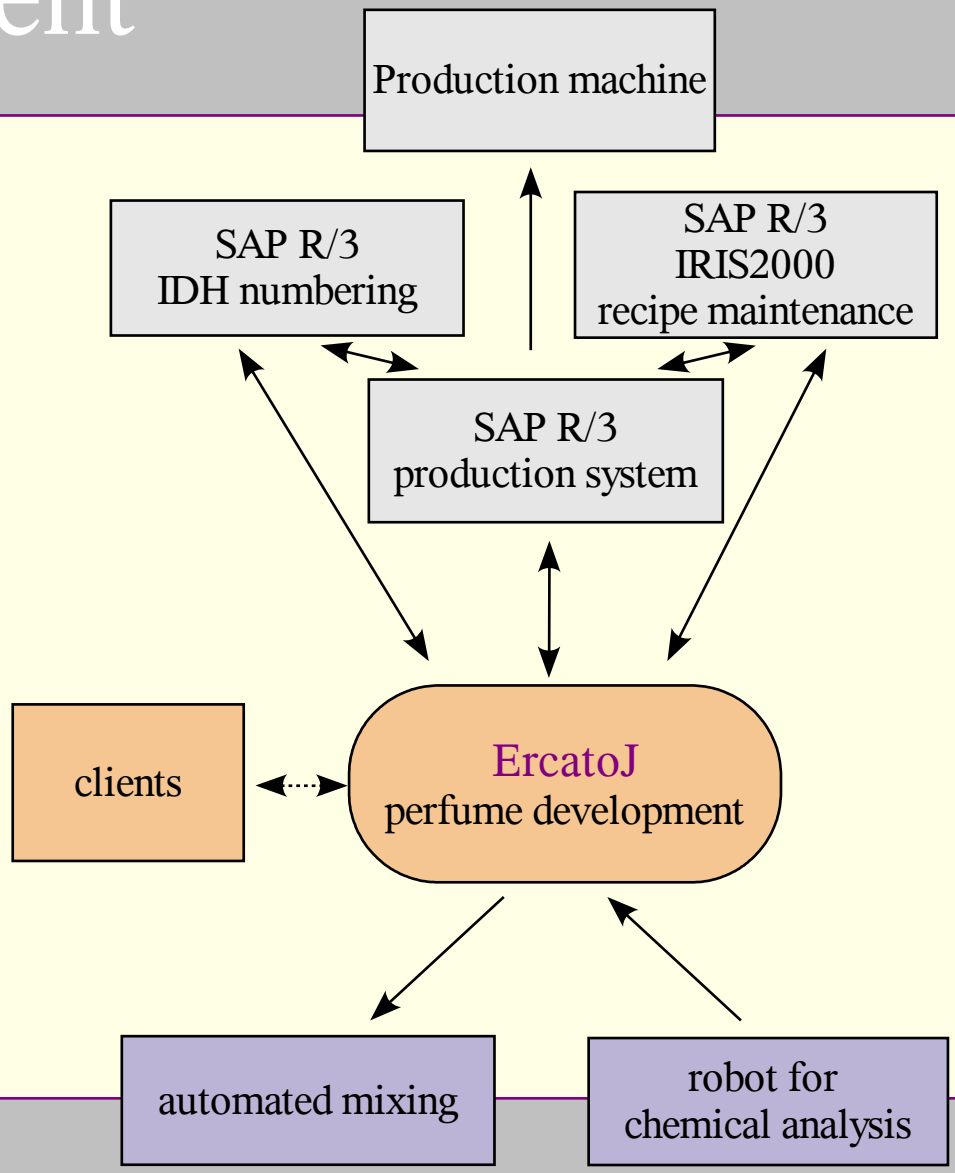
- make corporate IT happy (kind of have paranoia that processes work)
- Replace 1 million lines of code, short time, tight budget.

  I.e.:

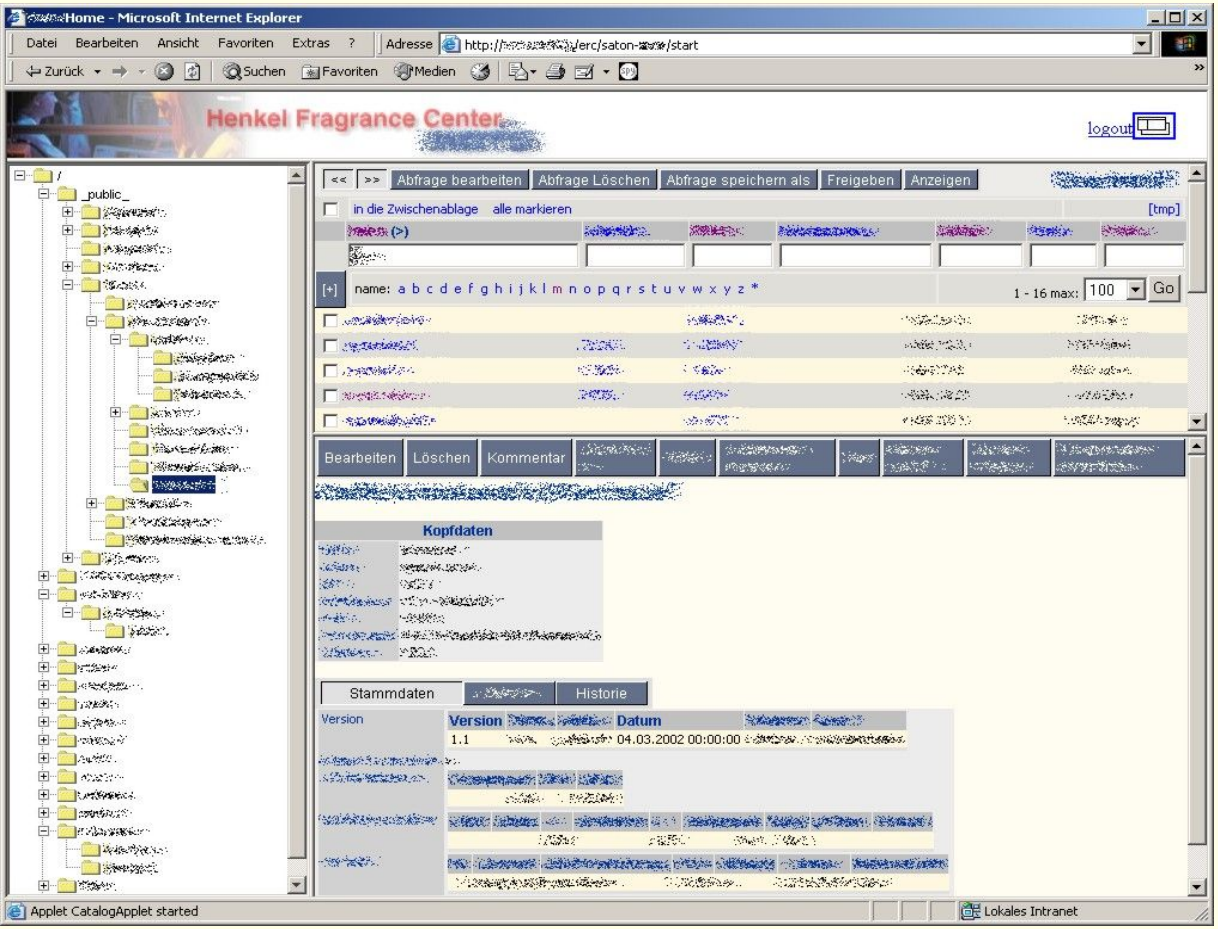
  - Map all business processes of all 5 departments at Henkel Fragrance Center
    - Departments of Analytic, Test and Mixing focus on technical requirements from chemistry and physics
    - Recipe administration department loves SAP
  - Meet security requirements for perfume recipes
  - Meet requirements for configurable searches and their performance

System environment

- Interface
  - to SAP production systems
  - to specialized machines for mixing and chemical analysis
- Substitute mission-critical host system
- Migrate existing base of corporate data



deployed solution



- Browser based intranet solution
- J2EE server application

Features

- Document-centric (versioned)
- Efficient configurable data mining
- Generated user interface
- Comprehensible implementation of workflow

Living Pages Research

so far so good. now the cool stuff

When in J2EE project, did you hear, too that...

development takes 42 percent longer than the worst estimate?

progress has slowed down?

hours of delay between coding and testing?

builds are a nightmare?

multiday transition to live systems?

your architecture becomes obfuscated?

business logic moves into JSPs?

What did we do different?

Don't know what You use, but we use...

ercatons



...because ercatons **do**
cut J2EE project complexity

Example wanted?

Task: Design & implement an “Address Manager” application
Detail: Should be network-ready and extensible

Time: You have ...5 minutes

here we go.

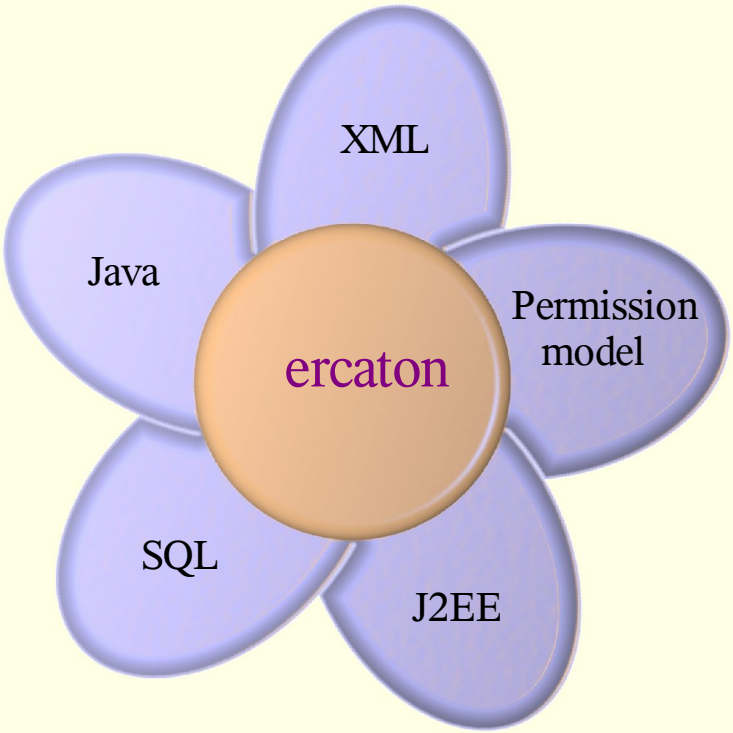
Verification left as an exercise to the reader

```
<?xml version="1.0" encoding="iso-8859-1"?>
<address xmlns:erc="http://ercato.com/xmlns/ErcatoCore"
         xmlns:erx="http://ercato.com/xmlns/ErcatoExtensions">
      <erc:id>~livis/adr/template</erc:id>
      <erc:catalog  category="/Adressen" id-ref="~livis/catalog"/>

      <name    erx:field-ref="string" erc index="~livis/catalog">Oster Hase</name>
      <street  erx:field-ref="string" erc:index="~livis/catalog">Wiese 7</street>
      <zipcode erx:field-ref="string" erc:index="~livis/catalog">Waldbröhl</city>
      <city    erx:field-ref="string" erc:index="~livis/catalog">12345</city>
      <phone   erx:field-ref="string">0190 666 666</phone>
      <erc:action name="edit">    /bin/edit       </erc:action>
      <erc:action name="delete">  /bin/rm$wizard  </erc:action>
      <erc:action name="newCopy"> /bin/cp$forEdit </erc:action>
      <erc:action name="check">
          <erc:arg name="default">Buxtehude</erc:arg>
          <erc:action-ref>~livis/check.xsl</erc:action-ref>
      </erc:action>
      <erc:trigger name="on-change">$check</erc:trigger>
</address>
```

What **is** an ercaton?

- Every ercaton encapsulates a business object or process. It is implemented in XML and an optional set of languages,incl. Java.
- Ercatons are fully programable and have easy access to transactions, permissions and indexing in a database. They are persistent by definition.
- Ercatons are instances, not classes. Nevertheless dynamic inheritance is defined and allows to extract common parts of sets of ercatons.



Where do ercatons live?

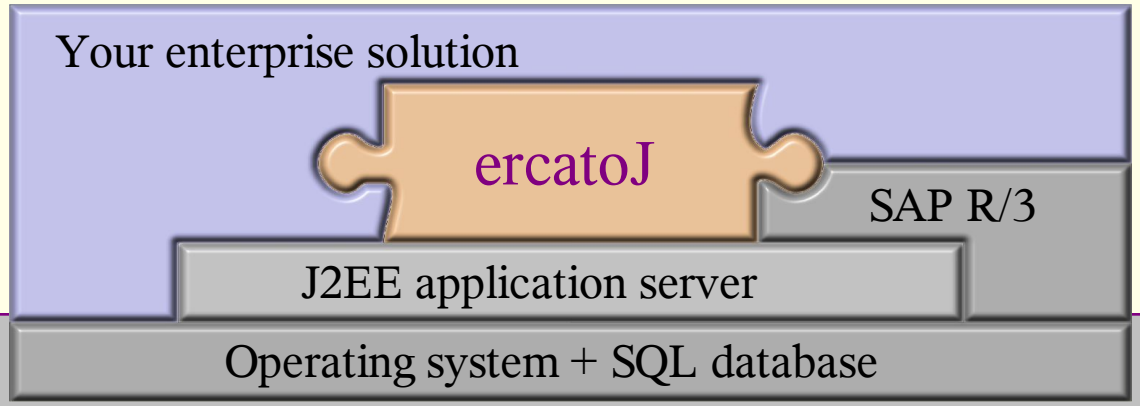
In the first place, ercatons need a certain amount of '<'-characters to exit.

For the more subtle features, a virtual environment is needed.

One such virtual environment is ercatoJ, a J2EE-based implementation.

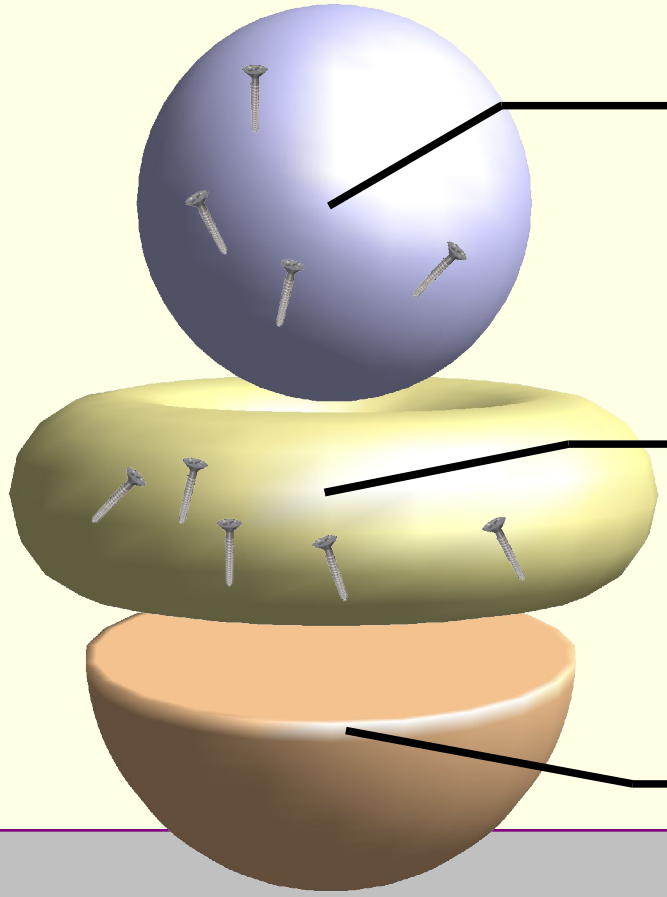
ercatoJ is a set of EJB's which need no redeployment, when the ercatons change. These EJB's integrate well into an existing J2EE application.

Every ercaton introduced absorbs complexity.



Living Pages Research

How does an ercato application work?



business processes:
customers
perfume oils
status propagation
trigger external systems
check legal validity
etc.

ercatoX standard extensions: (erx)
user interface management
standard services (cp, check, edit, etc.)
administration tools (backup etc.)
etc.

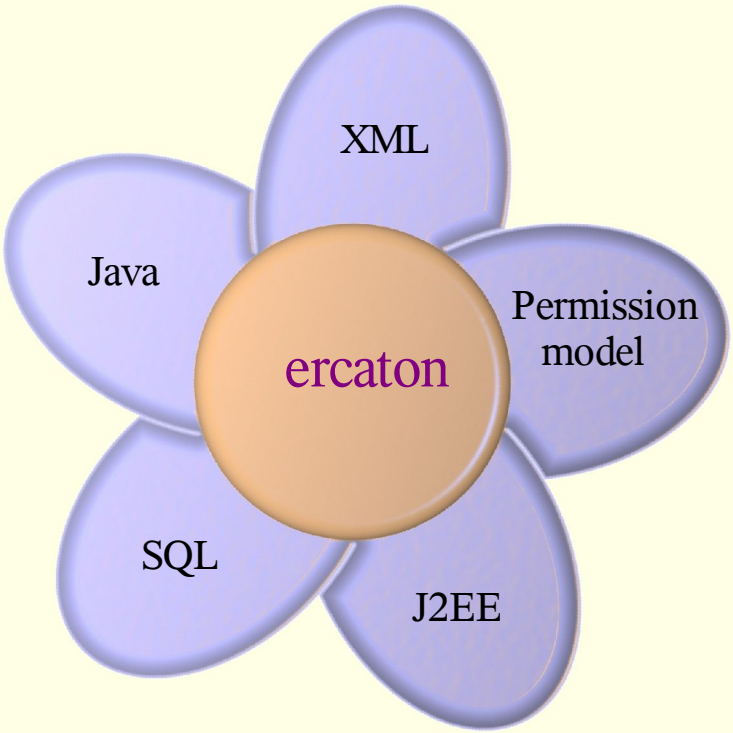
ercatoJ engine: (erc)
ercatons
permissions, transactions
actions, clones, versions
xml interchange
http (browser) access
etc.

Living Pages Research

More features of an ercaton

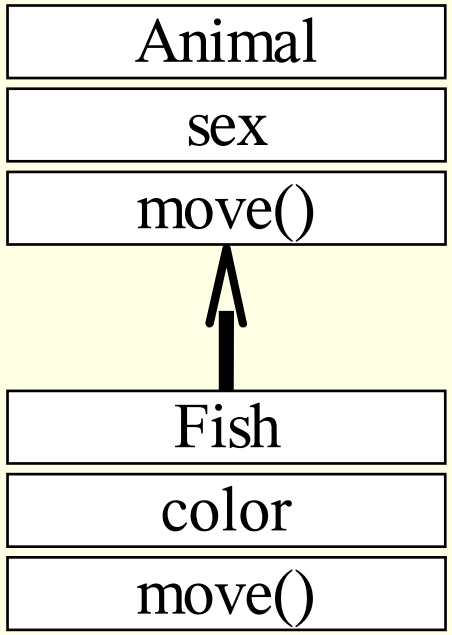
- User interfaces may be generated by style sheets depending on an output target
- Non XML resources are represented as “resource” ercatons. This includes JAR files, i.e., code may change at runtime.
- ercatons are versioned

ercatons are the foundation of a new programming model



Thing-oriented programming





Are the both *really* the same?

- software objects are a nice try to capture “the essential part” of real-world objects as far as a given model is concerned.
- software objects are model-dependent aka model-driven.
- models vary
- software CAN come much closer to real-world objects
- rich instances w/o classes
- call those rich objects “things”
- aka “prototype-based”

Living Pages Research

The ercaton programming model **is** thing-oriented

- an ercaton is a self-contained entity
  - with behavior
  - with persistent state and structure
  - with a (mutable) user interface
  - with ownership
- an ercaton stands up for itself, e.g., it does not depend on a class etc.
- each ercaton is an individual entity where no two are equal
- named after elementary particle convention (electron)

Why even screws are superior to objects

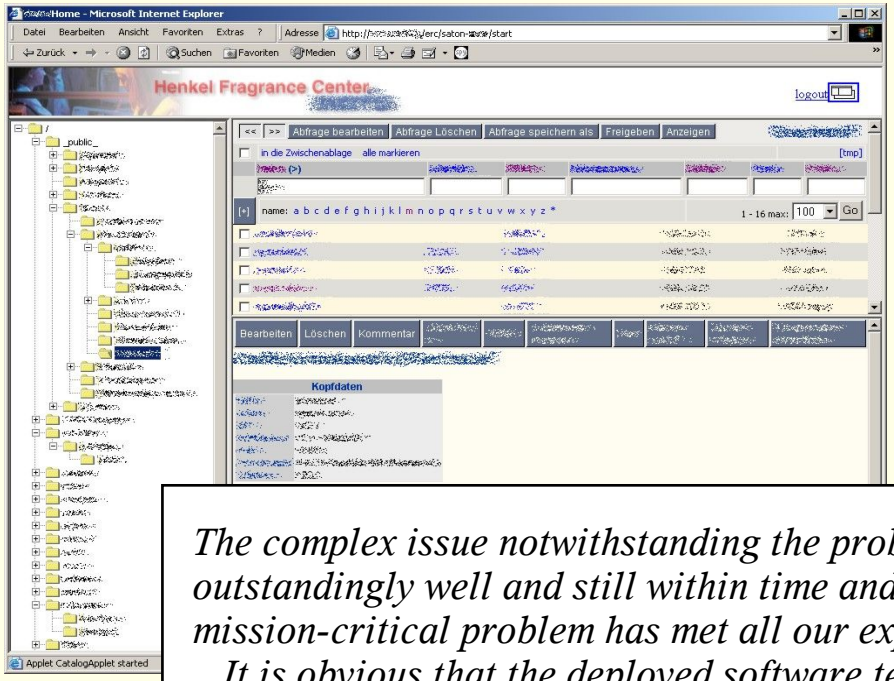


what screws have what objects don't have...

- screws have some vague properties (M6)
- screws are bought in quantities
- screws can be modified
- screws can be used without factory plan
- screws are self-explanatory
- screws scale to Eiffel-towers and beyond
- screws perfectly combine with other things

ercatons are more like screws than objects

Problem solved



- Was 1 million lines in host-based system
- One calendar year for development
- Went productive on schedule
- 10k+ lines of XML plus 10k+ lines of Java now make up the business logic

The complex issue notwithstanding the problem has been solved outstandingly well and still within time and budget. The solution of our mission-critical problem has met all our expectations.

It is obvious that the deployed software technology is well suited to address problems of a more complex nature, too.

Dr. Alexander Boeck
Geschäftsführer [Managing Director]
Henkel Fragrance Center GmbH

Henkel

Living Pages Research