CSC 405 Lecture Notes Week 3

Literature Survey of Software Deployment, Evolution, and Maintenance

I. Introduction.

A. We’ll discuss 3 papers from the readings

B. They’re wide-ranging surveys.

Three papers chosen from the readings, cont’d

C. The papers are

1. "Software Deployment, Past, Present and Future", Alan Dearle, *Proceedings of the 2007 Conference on the Future of Software Engineering*, IEEE CS.

Three papers chosen from the readings, cont’d

2. "Software evolution - Background, theory, practice", Meir M. Lehman, Juan F. Ramil, *Information Processing Letters 88 (2003)*, Elsevier.

Three papers chosen from the readings, cont’d

3. "25 Years of Software Maintenance", Scott Tilley, *2009 IEEE International Conference on Software Maintenance,*, IEEE CS.

The Deployment Paper

II. What is it?

A. A post-production activity.

B. Performed by developers and/or customers.

C. Has one or more of these steps:

What is it?, cont'd

1. *release*
2. *configuration*
3. *installation*
4. *activation*
5. *monitoring*

What is it?, cont'd

6. *updating*
7. *reconfiguration*
8. *redeploying*
9. *deactivation*
10. *undeploying*

III. What Gets Deployed

A. A formalized definition is *UML Component*

B. Has "*contractually specified interfaces and explicit context dependencies*".

C. Uses *well-defined resources*.

D. Is *versioned*.

What Gets Deployed?, cont’d

E. OMG defines *domain* as the target envir.

F. They define the inter-domain concepts of

1. *component discovery*
2. *component binding*

IV. Six Case Studies

A. ***Java Beans***

B. ***Linux***

C. ***.Net***

D. ***OMG, Corba***

E. ***Service-Oriented Computing***

V. Java Beans

A. Standard for server-side components.

B. Sun (now Oracle) defines process.

C. Use jar files and XML descriptor files.

Java Beans, cont'd

D. Issues:

No good "bean registry", making installation of new beans problematic

1. Very language dependent.

2. The "deploy with our free runtime" is an interesting model

VI. **Linux**

A. *RedHat Package Manager (RPM)* is cited as most common method of deployment.

B. Deploys in binary or source.

Linux, cont'd

C. Question -- are these RPM packaging elements standard for Linae?

1. *lead*

2. *signature*

3. *header*

4. *archive*

Linux, cont’d

D. Issues:

1. OS platform dependent.

2. Somewhat language independent (cf. Java Beans).

VII. **.Net**

A. Deployment unit is an *Assembly*.

B. Includes name, version, and public key.

C. Uses metadata manifests.

D. Uses platform-specific *Common Interface Language*.

.Net, cont’d

E. Issues:

1. OS platform dependent.
2. Somewhat more language independent (cf. Java Beans, Linux).
3. Anyone here have experience with it?

VIII. OMG CORBA

A. Ambitious attempt at platform independence.

B. Uses 5-step process:

OMG CORBA, cont’d

1. *installation*
2. *configuration*
3. *planning (for launch)*
4. *preparation (for launch)*
5. *launch*

OMG CORBA, cont’d

C. Uses various OMG models, e.g.,

1. Platform Independent Model (PIM).
2. CORBA Component Model (CCM).

D. Uses UML concepts, IDL mid-level definition, XML details.

OMG CORBA, cont’d

E. Issues:

1. Pretty darn complicated.
2. Still, most complete attempt at a deployment standard.

IX. Service-Oriented Computing

A. Examples include

1. Apache Axis.

2. Microsoft IIS

B. ***Question:*** Anyone have experience with these?

Service-Oriented Computing, cont’d

C. Issues:

1. It has potential.

2. Not clear yet if it’s a money maker

X. Virtualization

A. Abstract platform dependencies into a generic layer.

B. E.g., VMWare

Virtualization, cont’d

C. Issues:

1. Seems to have great potential.

2. Dearle says potentially *"profound implications"*.

XI. Not in Dearle's List, but Worth Considering

A. ***Browser-Based*** -- a glaring Dearle omission.

B. ***Debian, Fink*** and their ilk.

C. ***App Stores***, e.g., Apple and Android.

D. ***Question:*** Experience with these, anyone?

XII. Issues Revisited

A. *Binding*

1. compile time
2. link time
3. pre-runtime config
4. runtime

Issues , cont’d

B. ***Runtime environments***

1. J2EE and .Net are OK.
2. Browser = runtime environment is coming on.
3. So is virtualization.

C. ***Light Weight Containers***, e.g., *J2EE Spring*

E. *Metadata Config Files*