Discussion of Steimann's Essay: "The Paradoxical Success of Aspect-Oriented Programming"

Gary T. Leavens Department Of Computer Science, Iowa State University

A THE REAL PROPERTY AND A DESCRIPTION OF A

00PSLA, October 26, 2006

Supported in part by the US NSF

Outline

ander andere eine der 199 besten und die eine Antonio and der Antonio and ander 1 beste 195 von der Antonio and

- Summary
- Important points
- Perspective: what questions?
- Questionable points
- Conclusion

Summary: Definition and Paradox

- Need definition of AOP languages
 Definition offered ≈ implicit invocation
 Paradox:
 - Ok in moderation, but then no contributionHurts modularity if more developed

Summary: Modularity of AOP

Information hiding (Parnas) "Main concern" (5.1.2): Strong coupling of aspects to base Implicit interface, often not public Impairs independent development Explicit interfaces for aspects in base: Less obliviousness More scattering

Summary: Locality of AOP

naya analaya ka mada ili 👬 kana 🖉 tari 20 ali ali makalari ka kanalarika da ali ma kanala 🖬 ka 🕯 ka kanala kanala kanala kanala kanala kanala kanala

Implicit invocation

 less direct control flow

 Context exposure

 more global access to variables



Summary: Utility of AOP

Generated code weaving Forming new components (glue code) Dismantle components Reassemble to form new components Observed uses: Logging Security Runtime assertion checking

Summary: Prospects for AOP

na a ann an an an Alain Inaca 🖬 tha an a Alainna ann an Anna ann an Alainn a bhana 🕯 ann a Alainn an ann an Alainn an Al

"Tremendous success"
Cachet
OOP is old, need something new
Paradox (section 8):
Aims to "modularize crosscutting concerns"
"Its very nature ... breaks modularity"

Important Points: What is AOP?

are, and a second of the base of the late of the late of the base of the base of the late of the

Focus on mechanisms (vs. goals)
Need definition of AOP languages
Mechanisms
Independent of goals
Implicit invocation definition sensible

Important Points: Modularity of AOP

Interfaces are crucial
Explicit interfaces for aspects in base:
Less obliviousness
More scattering

Important Points: Utility of AOP

AOP especially useful as noted
Generated code weaving

Aids modularity of generator
AspectJ = Assembly language of 2000's

Glue code is important

Important Points: Prospects for AOP

ander andere se ander 1914 have of the loss of

AOP has been a success

TRANSPORTER AND INCOME AND ADDRESS OF A DESCRIPTION OF A



Perspective

Flon's axiom (SIGPLAN, Oct. 1975):

"There does not now, nor will there ever, exist a programming language in which it is the least bit hard to write bad programs."



and a second second

Flon's axiom (SIGPLAN, Oct. 1975): Can write bad programs in any language

CONTRACTOR OF STATES AND ADDRESS OF STATES AND ADDRESS AND ADDRE

October 25, 2006



Perspective: What are the right questions?

Flon's axiom (SIGPLAN, Oct. 1975): Can write bad programs in AOP languages
Can we write good ones?
Without scattering and tangling of crosscutting concerns
Without excess coupling of aspects to base

Perspective: All or Nothing?

Modularity
System is modular or not
Obliviousness
Mechanisms are oblivious or not

Perspective: Questions of Degree



October 25, 2006

Perspective: Perfect AOP Measures

and a subscription of the second s









Perspective: Compromise



October 25, 2006

Perspective: N-Dimensions of Modularity



Questionable Points: Coupling of Aspects and Base

and a submitted a state base of the last of the last development of the last of the submitted base of the submitted base of the last of the submitted base of the last of the submitted base of the

Section 5.1.2:

Code "moved out of its context ..."
"it must take (a reference to) the context that it depends on with it,"
"thereby establishing coupling"

How often?

How much coupling?

Questionable Points: Reducing Coupling

XPIs (Griswold et al.)
 Adding interfaces for crosscutting
 Extra indirection reduces coupling
 Open Modules (Aldrich)
 Explicit interface to advice
 Reduces obliviousness

Questionable Points: Paradox

and a submitted a constant and a stand of the standard and a standard the standard block at the standard block a

Due to all-or-nothing usage of: Modularity Quantification and obliviousness AOP's contribution Due to multiple senses of "modularity" Information hiding (lack of coupling) Encapsulation (lack of scattering)

Questionable Points: Lack of Imagination

"I find it difficult to imagine ..." (4.2)
"I cannot see how ..." (4.3)
"it may be my lack of imagination" (5.1.7)

Questionable Points: Lack of Imagination

any a california a construction of the base of the second definition of the second second second second a second s

Most technology nay-sayers wrong Some promising directions Tool support (Eclipse AJDT) XPIs Open Modules Static analysis Model-driven architecture Annotations

Conclusions

Thought-provoking Read it! ■ Agree: AOP ≈ implicit invocation Perspective on "paradox": N-dimensions of modularity Each a scale, not all-or-nothing Compromise! Research: how to do better



Time for Discussion



October 25, 2006