

# Help! Evil services are killing my objects

Asked 16 years, 1 month ago   Modified 14 years, 8 months ago

Viewed 382 times

---



2

When I believed in American dream about encapsulation and polymorphism, intrusion of Web Services washed my objects off with RPC calls...



When I cherished my resurrected PONO's, ugly army of barbarians called proxy objects conquered my lands...



Later, peace seemed to come back with DDD and NHibernate on the server side, but the SilverLightning hit my castle, now there's hunger again, delicious lazy loading is only in my memories, and for years now my poor objects have to consume stale services again...

And I am full of fear...the world is talking more and more about some other terrifying procedural monsters...they call them "Workflows"...

How can I save my objects? Literally, I do not provide anyone any services. I am building a simple small system. I don't want to use services to find my data. I do not want to use services to talk from my web interface to my web interface...as I don't want to use snail mail to talk to my colleagues.

Any ideas? Did you manage to save your objects? Did you manage to save more than your domain model? (hopefully you managed the latter...)

**Update:** If this was not clear... We have a killed architecture because everything is using web service based. There was a fashion "OO - is dead", services rule. In SOA, it is still quite hard to focus on objects when everything is focusing on verbs ("operation contracts"). I feel it is hard to take care of your design.

oop

architecture

domain-driven-design

service

Share

edited Apr 13, 2010 at 0:03

Improve this question

Follow

community wiki

5 revs, 3 users 71%

badbadboy

---

What did you smoke? Did you really want an answer to, err, whatever that was? – [Stein G. Strindhaug](#) Nov 20, 2008 at 17:56

---

1 i'm thinking "way too much acid" - what are you babbling about? – [Steven A. Lowe](#) Nov 20, 2008 at 17:56

---

@Steven - you are saying "too much acid"? If something doesn't make sense to you, or doesn't make sense at all, or is a joke, or you didn't get the joke, or maybe it was not a

good joke - I wouldn't say "too much acid". Thanks for encapsulation btw. – [Paul Kapustin](#) Nov 20, 2008 at 18:40

---

@Stein - ingen snus i dag. Yes, I wanted the answer. Will update the question. – [Paul Kapustin](#) Nov 20, 2008 at 18:41

---

it was a joke, badbadboy, and probably not a good one. though your rant was funny, i really didn't understand the problem and question – [Steven A. Lowe](#) Nov 20, 2008 at 19:05

---

## 4 Answers

Sorted by:

Highest score (default)



3



Beware you foolish mortals. The Entity That Is has indeed fed on your polymorphed objects. But this also means that you have inherited the Big Slimy Interface that lurks in the dark. So you can retire your puny barbarians (by proxy if you wish).



And yes, thanks to the Entity That Is, your objects got lazy and have their garbage collected. So their joy is only temporary because their life is immediate ended when they move out of scope. And not a single one of them can get away.

If you show fear to the Entity That Is, dead is only a destructor away. So be careful when you ride the waves of the workflow, because they are as unpredictable as the average market stock.

Your objects are never save for the Entity That Is. Persistence can back them up for a while but eventually

all wil fail as the last clock cycle has rung. Fortunately thanks to persistance, your objects can be send to better places where they can multiply and live in peace.

The Entity That Is, is strict but fair, so if you use the propret command, your virtual doors to other realities wil open and allow swift and reliable trafic.

Good luck and honour the Entity That Is, you may not always agree with it, but its rule is the law and death the only penalty.

Share Improve this answer

answered Nov 20, 2008 at 19:04

Follow

community wiki

Toon Krijthe

---

@Gamecat: you made me laugh :)) – Paul Kapustin Nov 20, 2008 at 23:36

---



1

i *think* you're complaining about verb-centered designed when using SOA. If so, that is not a requirement of SOA, but it is a temptation.



"Anywhere in a normal OO application that you would do something, change that something to a web service" is probably overkill



the best uses for SOA that i've seen just replace the data access layer with an SOA layer, plus they expose a few

high-level 'public' operations like registering a new user and so on

one could make a web service out of every class method, but this would be ridiculous in most cases...

Share Improve this answer

answered Nov 20, 2008 at 19:07

Follow

community wiki

[Steven A. Lowe](#)

---

yes, I complain about this. So what should CreateUser() return? Id? DTO? We have 35 web services, and these DTO's all over... we have a lot of layers pulling out these DTO's, and all layers are verb-centric. CreateUser()...pass id here, pass id there..kind of inferred anemic domain model...

– [Paul Kapustin](#) Nov 20, 2008 at 19:15

---

Though, no one to blame except myself of course. Domain model was described by M.Fowler many years ago. Services are good. You just need to know how to consume them..

– [Paul Kapustin](#) Nov 20, 2008 at 19:20

---

No ids -- use business keys! Its a dead giveaway that you aren't really abstracting the underlying repository.

– [Jennifer Zouak](#) Apr 13, 2010 at 0:08



1

It seems like the two ain't compatible when scaling. Amazon apparently lost plenty of time and money due to issues with versioning and their object models. SOA layers seem to work better if there is no dependant object



definition leaving the consumer of the services to map EACH call to their own domain model..... hmm.....



Share Improve this answer

answered [Nov 20, 2008 at 19:12](#)

Follow

community wiki  
[Quibblesome](#)



0

Clearly you haven't take your abstraction pills this morning. Now, take your nice medicine and you'll feel better in a little while...



Share Improve this answer

answered [Nov 20, 2008 at 18:18](#)

Follow



community wiki  
[Kluge](#)

---

do you mean I am too abstract or do you mean there were too few abstractions in the code to support OO in SOA?

– [Paul Kapustin](#) Nov 20, 2008 at 18:45

---

@badbadboy: Don't read too much into my answer. Your clever metaphors made me laugh, and I was just trying for a quick, funny response, which in retrospect, probably wasn't that funny... – [Kluge](#) Nov 20, 2008 at 20:28

---

@Kluge: thanx. I laughed reading your answer too :) Just that I first got only sarcastic comments and that's why I was wondering :) – [Paul Kapustin](#) Nov 20, 2008 at 23:35

---

