

The Art of Visualising Software Architecture



## ...the architecture diagrams don't match the code





# Any recommendations for software for drawing software architecture but not MS Visio?



11:11 AM - 16 Apr 2015





































I help software teams understand software architecture, technical leadership and the balance with agility

## Software architecture needs to be more

# accessible

coding architecture

Software Architecture

Technical leadership by coding, coaching, collaboration, architecture sketching and just enough up front design The Art of Visualising Software Architecture

coding-----architecture

Communicating software architecture with *sketches*, diagrams and the C4 model

Simon Brown



Simon Brown



## I code too \$\frac{1}{1}\$ (-) \$\psi 0\$

## The problem

#### Financial Risk System

#### 1. Context

A global investment bank based in London, New York and Singapore trades (buys and sells) financial products with other banks (*"counterparties"*). When share prices on the stock markets move up or down, the bank either makes money or loses it. At the end of the working day, the bank needs to gain a view of how much risk of losing money they are exposed to, by running some calculations on the data held about their trades. The bank has an existing Trade Data System (TDS) and Reference Data System (RDS) but needs a new Risk System.

#### 1.1. Trade Data System

The Trade Data System maintains a store of all trades made by the bank. It is already configured to generate a file-based XML export of trade data to a network share at the close of business (5pm) in New York. The export includes the following information for every trade made by the bank:

• Trade ID, Date, Current trade value in US dollars, Counterparty ID

#### 1.2. Reference Data System

The Reference Data System stores all of the reference data needed by the bank. This includes information about counterparties (other banks). A file-based XML export is also generated to a network share at 5pm in New York, and it includes some basic information about each counterparty. A new reference data system is due for completion in the next 3 months, and the current system will eventually be decommissioned. The current data export includes:

Counterparty ID, Name, Address, etc...

#### 2. Functional Requirements

- 1. Import trade data from the Trade Data System.
- 2. Import counterparty data from the Reference Data System.
- 3. Join the two sets of data together, enriching the trade data with information about the counterparty.
- 4. For each counterparty, calculate the risk that the bank is exposed to.
- Generate a report that can be imported into Microsoft Excel containing the risk figures for all counterparties known by the bank.
- 6. Distribute the report to the business users before the start of the next trading day (9am) in Singapore.
- 7. Provide a way for a subset of the business users to configure and maintain the external parameters used by the risk calculations.

"Financial Risk System" architecture kata Simon Brown | @simonbrown



"Design a solution & draw some pictures to describe it"

## Abstraction is about reducing detail

rather than creating a different representation





The Shopping List



Boxes & No Lines

FUNCTIONAL VIEW





Stormtroopers



The Airline Route Map



Generically True





Missing technology details



### Deployment vs Execution Context



Homeless Old C# Object (HOCO)



Choose your own adventure



Should have used a whiteboard!



Eh?

Challenging? Level of detail Lutere to stop different Who is the audience backgrands Implementation - easy to get bogged dan Type of diagrams Notaction Documenting assumptions

What's been challenging about the exercise?

7 Challerging Needed to ask questions/ make assumptions Temptation to focus on detail Luker do me stop? How much detail? Talked about more than the diagravs What notation? \_boxes - arrows (10) Challenging? Verifying our own assumptions Expressing the solution - communicating it is a clear way -use of notation - easy to mix levels of obstraction - how much detail?

## People expect to present their designs and therefore information is still stuck in their heads



## The diagram isn't self-evident, but we'll explain it

## Review the diagrams

3 things we like about the diagrams

3 things we think would improve the diagrams

(focus on the communication of the solution rather than the solution itself)

NO ANNOTATION SHOULD USE POST ITS ON FLOWS shal dog CAN FALL MORE CGLUVr OFF COLORS mean? NOT SURE OF MIXES TRANSITION Manger Cot Ferstarres LONFLICTING Between DIFPERENT DARF GROST Objects VS LEVELS OF LEVELS OF What ebout DIAGRAMS -DETAIL DETAIL IN the different dictions PRESENTATION what with ARE attows! WHAT DO shape unes boundary THE SHAPE UML 15 GODD WHAT DO MEAN Z BUT NOT LINES RE-EVERY ONE KNOWS PINKC mean PRESENT2 DATA 7 (DNTRO Not sure (i) What's the DEP.7) DIFFERENT what this ARE THE LEVELS IN DB-like ARROWS THE SAME RIGHT WAY icon? 15 DIAHRAM Zouros 2.

# Who here uses UML on a regular basis?



## 9 out of 10 people don't use UML

(in my experience)

# I do use UML

#### (activity, class, sequence, collaboration, state)





in Children

(1 m) (1 m) (1 m)

() uti

Figure 2-2 Design Level Component/Block diagram

~


In my experience, software teams aren't able to effectively visualise the software architecture of their systems



## We can **Visualise our process**...

# ...but not our software





# Moving fast requires good communication



### Titles

Short and meaningful, numbered if diagram order is important

### Lines

Make line style and arrows explicit, add annotations to lines to provide additional information

### Layout

Sticky notes and index cards make a great substitute for drawn boxes, especially early on

## Some tips for

effective sketches





effective sketches





It's usually difficult to show the entire design on a **Single** diagram

Different **VIEWS** of the design can be used to manage complexity and highlight different aspects of the solution

Software architecture deals with abstraction, with decomposition and composition, with style and esthetics. To describe a software architecture, we use a model composed of multiple views or perspectives.

Architectural Blueprints—The "4+1" View Model of Software Architecture by Philippe Kruchten The description of an architecture—the decisions made—can be organized around these four views, and then illustrated by a few selected *use cases*, or *scenarios* which become a fifth view. The architecture is in fact partially evolved from these scenarios as we will see later.



We apply Perry & Wolf's equation independently on each view, i.e., for each view we define the set of elements to use (components, containers, and connectors), we capture the forms and patterns that work, and we capture the rationale and constraints, connecting the architecture to some of the requirements.



## Viewpoints

and perspectives



Viewpoint	Definition
Logical	The logical representation of the system's functional structure, normally pre- sumed to be a class model (in an object-oriented systems development context). Our Functional viewpoint is a development of this "4+1" viewpoint, renamed to make its content clear (because you could have a number of logical aspects to an architecture).
Process	The concurrency and synchronization aspects of the architecture. Our Concur- rency viewpoint is a development of this "4+1" viewpoint, renamed to avoid confusion with business process modeling.
Development	The design-time software structure, identifying modules, subsystems, and layers and the concerns directly related to software development. Our Development viewpoint is based on this "4+1" viewpoint.
Physical	The identification of the nodes on which the system's software will be executed and the mapping of other architectural elements to these nodes. Our Deployment viewpoint is a development of this "4+1" viewpoint.

## Do the **names** of those views make sense?

Conceptual vs Logical Process vs Functional Development vs Physical Development vs Implementation Physical vs Implementation Physical vs Deployment

Why is there a separation between the logical and development views?



## "the model-code gap"

**Model-code gap.** Your architecture models and your source code will not show the same things. The difference between them is the *model-code gap*. Your architecture models include some abstract concepts, like components, that your programming language does not, but could. Beyond that, architecture models include intensional elements, like design decisions and constraints, that cannot be expressed in procedural source code at all.

Consequently, the relationship between the architecture model and source code is complicated. It is mostly a refinement relationship, where the extensional elements in the architecture model are refined into extensional elements in source code. This is shown in Figure 10.3. However, intensional elements are not refined into corresponding elements in source code.

Upon learning about the model-code gap, your first instinct may be to avoid it. But reflecting on the origins of the gap gives little hope of a general solution in the short term: architecture models help you reason about complexity and scale because they are abstract and intensional; source code executes on machines because it is concrete and extensional.

## Do the diagrams reflect the





Workspace last modified: Wed 09 Mar 2016 11:55 UTC





Component diagram for Financial Risk System - Batch Process Workspace last modified: Wed 09 Mar 2016 11:55 UTC

## As an industry, We lack a COMMON VOCABULARY with which to think about, describe and communicate software architecture

#### Software System



A SOFTWARE SYSTEM is made up of one or more CONTAINERS, each of which contains one or more COMPONENTS, which in turn are implemented by one or more Classes.

## Runtime and behaviour

(sequence and collaboration diagrams of elements in the static model) Business process and workflow

#### Data (entity relationship diagrams)

## Static model

(software systems, containers, components and classes) etc...

### Infrastructure

(physical, virtual, containerised hardware; firewalls, routers, etc) Deployment (mapping of containers to infrastructure)

## The C4 model



## System Context

The system plus users and system dependencies



## Containers

The overall shape of the architecture and technology choices



## Components

Components and their interactions within a container



Classes (or Code)

Component implementation details





**Diagrams** are **maps** that help a team navigate a complex codebase

# Think about the target audience



## A common set of abstractions is more important than

a common notation



#### News

#### © C5 Alliance plans Microsoft events in Channel Islands

Channel Island cloud provider, C5 Alliance are organising two breakfast events in both Jersey and Guernsey, named 'Leveraging Microsoft Technologies for Regulatory Compliance'. The breakfast briefings are due to include demonstrations of the latest Microsoft technologies and how they are combined. The briefings will cover Microsoft CRM process driven forms, SharePoint Workflow & Collaboration and SQL Server Data Warehousing technology. C5 Alliance, who work with a number of clients, both financial after...

Posted Today

#### Jersey residents set to have choice in fibre broadband

Sure customers will soon be able to access Jersey's fibre network following the reaching of an agreement between Sure and JT that finalises the commercial arrangements for access to the network. The agreement means that JT has gone some way to fulfilling the second condition of the eight that were set out in the States of Jersey's funding arrangements for the network, as agreed by the Treasury Minister, Senator Philip Ozouf. "This is excellent news for our broadband customers who have been extremely pati...

Posted Yesterday

#### Logicalis Group taking over Jersey cloud provider

Logicalis Group, the international IT solutions and managed services provider, has announced the acquisition of Jersey's iConsult Limited, a privately owned Jersey company and provider of desktop and mail hosted solutions to the small medium businesses (SMB) market within the Channel Islands. Through their data facility in Jersey the company services over 800 users on the Islands, mainly in the financial and professional services sectors. Their main offering is a hosted desktop solution, using primarily ...

Posted 18 Oct 2013

More...

#### Local events

#### Van Nikkhoo - Growth Funding

Topics of discussion will be: Growth capital, Funding cycles, Investment decisions plus Valuation and exits. With over 29 years of industry experience in various senior capacities internationally, Ivan is a Managing Director at Siemer & Associates and a...

Pomme d'Or Hotel, St Heller, Jersey 29 Oct 2013 et 17:30

#### Tech Tribes Talks

The third set of Tech Tribes talks are ready to rock your world!After a very successful July event at the Royal Yacht we've decided to go back for our October talks. We have a great line up of speakers and we take great pleasure in inviting you to atte...

The Royal Yacht, St Heller 24 Oct 2013 at 17:30

#### 🗊 The Internet of

#### **Everything and Gigabit**

#### Jersey

"The Internet of everything" Currently, there are an estimated 10 to 15 billion "things" connected to the Internet and this is predicted to grow to 50 billion by 2020. How will this change our lives? What infrastructure will we need? What opportun...

The Grand Hotel, St Heller, Jersey Tomorrow at 17:15

#### 2014 2013 2012

#### Talks by local speakers

#### 🛐 Ted talk

#### Agile software

Software architecture and

## techtribes.je







techtribes.je - Context

Container diagram (level 2)



Context diagram (level 1) Container diagram (level 2)

## Component diagram

Class diagram



Class diagram (level 4)



## A simple notation

(whiteboard and sticky note friendly, supplemented with colour coding)

### Anonymous User

[Person] Anybody on the web.

### techtribes.je

[Software System]

techtribes.je is the only way to keep up to date with the IT, tech and digital sector in Jersey and Guernsey, Channel Islands.

### Web Application

[Container: Apache Tomcat 7.x]

Allows users to view people, tribes, content, events, jobs, etc from the local tech, digital and IT sector.

### **Twitter Connector**

[Component: Spring Bean + Twitter4j]

Retrieves profile information and tweets (using the REST and Streaming APIs).





Shapes and colour can add an additional layer of information

# 

Enterprise context User interface mockups and wireframes Domain model Sequence and collaboration diagrams Business process and workflow models Infrastructure model Deployment model The description of an architecture—the decisions made—can be organized around these four views, and then illustrated by a few selected *use cases*, or *scenarios* which become a fifth view. The architecture is in fact partially evolved from these scenarios as we will see later.



We apply Perry & Wolf's equation independently on each view, i.e., for each view we define the set of elements to use (components, containers, and connectors), we capture the forms and patterns that work, and we capture the rationale and constraints, connecting the architecture to some of the requirements.

### 4+1 architectural view model

Philippe Kruchten

Software Systems Architecture

Working with Stakeholders Using Viewpoints and Perspectives

NICK ROZANSKI • EOIN WOODS

Second Edition

### Software Systems Architecture

Working with Stakeholders Using Viewpoints and Perspectives (2nd Edition)

Nick Rozanski and Eoin Woods
# C4 is **not** a design process

Up front design VS retrospectively drawing diagrams



# What's the easiest solution to create software architecture diagrams?

I don't want to bother with adhering to UML or finding the proper symbol for the right type of component. I might want a cloud, or 5 boxes in a larger box. Or some other shape. And I'd like a nice variety of arrows to choose from. Something to whip together a diagram fast, but equally important, it should look pretty. Something professional, that could go in a publication or on a company's web site. Certainly free is better, but I think I'm willing to pay once to not have to fight the whatever tool I use to document software anymore.

Any suggestions?

http://quora.com/Whats-the-easiest-solution-tocreate-software-architecture-diagrams



Any **general purpose diagramming tool** can be used to create software architecture diagrams

## Do building architects use Microsoft Visio?



# Sketches get out of date, so why not auto-generate the diagrams?

## Spring PetClinic

https://github.com/spring-projects/spring-petclinic/

		Sprir s o u r
Views	JSP with custom tags	Thymeleaf
+ Bootstrap (CSS)	&& webjars	&& Dandelion
Controller		Spring @MVC annotations
		Bean Validation
Service		@Cacheable
		@Transactional
_		Spring Data JPA
Repository	3 profile	es default (JPA)
		jdbc

https://speakerdeck.com/michaelisvy/ spring-petclinic-sample-application

	org.springframework.samples.petclinic
$\overline{\mathbf{w}}$	💼 model
	C & BaseEntity
	C & NamedEntity
	C & Owner
	C & Person
	C & Pet
	C & PetType
	C & Specialty
	C a Vet
	C a Vets
	C a Visit
$\overline{\mathbf{w}}$	repository
	🔻 🛅 jdbc
	C a JdbcOwnerRepositoryImpl
	© ∘ JdbcPet
	C & JdbcPetRepositoryImpl
	Subscription of the second
	C & JdbcVetRepositoryImpl
	C & JdbcVisitRepositoryImpl
	① & OwnerRepository
	B PetRepository
	b VetRepository
	b VisitRepository
v	service
	ClinicService
	ClinicServiceImpl
v	■ util
	CallMonitoringAspect
_	C EntityUtils
v	• web
	CrashController
	Controller
	PetController
	PetTypeFormatter
	PetValidator
	S VetController



An auto-generated UML class diagram

#### Software Reflexion Mo-Bridging the Gap between Source and

1

Cl

1 Intr

Software e

software s

Box and ar

are often :

though the

soning abo

can be dan

always ina

Current

high-level

derived me

their very

source. Al

by these re

from the m

ple of this i

Figure 1,

sufficiently

ent way. T

of interest

call graph

source cod

between t

model is th

engineer's

agree with

terprets th

modifies t

tional refle

flexion" from

<sup>1</sup>The old

We have

source.

Gail C. Murphy and David Notkin

Dept. of Computer Science & Engineering University of Washington Box 352350 Seattle WA, USA 98195-2350 {gmurphy, notkin}@cs.washington.edu

#### Abstract

Software engineers often use high-level models (for instance, box and arrow sketches) to reason and communicate about an existing software system. One problem with high-level models is that they are almost always inaccurate with respect to the system's source code. We have developed an approach that helps an engineer use a high-level model of the structure of an existing software system as a lens through which to see a model of that system's source code. In particular, an engineer defines a high-level model and specifies how the model maps to the source. A tool then computes a software reflexion model that shows where the engineer's high-level model agrees with and where it differs from a model of the source.

The paper provides a formal characterization of reflexion models, discusses practical aspects of the approach, and relates experiences of applying the approach and tools to a number of different systems. The illustrative example used in the paper describes the application of reflexion models to NetBSD, an implementation of Unix comprised of 250,000 lines of C code. In only a few hours, an engineer computed several reflexion models that provided him with a useful, global overview of the structure of the NetBSD virtual memory subsystem. The approach has also been applied to aid in the understanding and experimental reengineering of the Microsoft Excel spreadsheet product.

<sup>0</sup>Permission to make digital/hard copies of all or part of this material without fee is granted provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the Association for Computing Machinery, Inc. (ACM). To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGSOFT '95 Washington, D.C., USA ©1995 ACM 0-89791-716-2/95/0010...\$3.50

#### Introduction

Software engineers often think about an existing software system in terms of high-level models. Box and arrow sketches of a system, for instance, are often found on engineers' whiteboards. Although these models are commonly used, reasoning about the system in terms of such models can be dangerous because the models are almost always inaccurate with respect to the system's source.

Current reverse engineering systems derive high-level models from the source code. These derived models are useful because they are, by their very nature, accurate representations of the source. Although accurate, the models created by these reverse engineering systems may differ from the models sketched by engineers; an exam-

<sup>\*</sup>This research was funded in part by the NSF grant CCR-8858804 and a Canadian NSERC post-graduate scholarship.

# Diagramming tools see COQLE

rather than components

# What is a "Component"?

What are the architecturally significant elements?





A UML class diagram showing architecturally significant elements



#### A component diagram, based upon the code

# The code is the embodiment of the architecture

In practice, architecture is embodied and recoverable from code, and many languages provide architecturelevel views of the system.

> A Survey of Architecture Description Languages by Paul C. Clements

# Is the architecture *in* the code?

#### Context



#### People

Security groups/roles in configuration files, etc.

### Software Systems

Integration points, APIs, known libraries, credentials for inbound consumers, etc.

#### Containers

IDE projects/modules, build output (code and infrastructure), etc. **Components** Extractable from the code if an architecturally-evident coding style has been adopted.

#### Containers



#### People

Security groups/roles in configuration files, etc.

### Software Systems

Integration points, APIs, known libraries, credentials for inbound consumers, etc.

#### Containers

IDE projects/modules, build output (code and infrastructure), etc. **Components** Extractable from the code if an architecturally-evident coding style has been adopted.

#### Components



#### People

Security groups/roles in configuration files, etc.

### Software Systems

Integration points, APIs, known libraries, credentials for inbound consumers, etc.

#### Containers

IDE projects/modules, build output (code and infrastructure), etc.

#### Components

Extractable from the code if an architecturally-evident coding style has been adopted.



# "architecturally-evident coding style"

# Architecturally-evident coding styles include:

Annotations/attributes (@Component, [Component], etc) Naming conventions (\*Service) Namespacing/packaging (com.mycompany.system.components.\*) Maven modules, OSGi modules, Java 9 and Jigsaw, JavaScript module patterns, ECMAScript 6 modules, microservices, etc

Extract as much of the software architecture from the code as possible, and supplement where necessary

# Create an architecture description language **USING CODE**

## Structurizr for Java Structurizr for .NET

GitHub This repository Search	Explo	ore Features Enterprise Pricing	Sign up Sign in
Structurizr / java	quests o 🦛 Pulse 📊 Graphs	⊙ Watc	h 41 ★ Star 159 ∛ Fork 60
Structurizr for Java			
242 commits	<b>₽ 1</b> branch	🟷 5 releases	6 contributors
Branch: master - New pull request	New file Find f	HTTPS - https://github.	com/stru 🚔 😃 Download ZIP
simonbrowndotje Some doc tweaks.			Latest commit 5b71952 4 days ago
docs	Some doc tweaks.		4 days ago
gradle	Gradle Support: cleanup workspace, co	mmit gradle wrapper, remove ant/ivy	a year ago
structurizr-annotations/src/com/structu.	. Some refactoring and an initial version	of a Javadoc component finder	a year ago
structurizr-client	Added a simplified constructor.		3 months ago
structurizr-core	Some minor tweaks.		5 days ago
in structurizr-dot	Tweaks following merge.		3 months ago
structurizr-examples	Tweaks to getting started example.		4 days ago
structurizr-spring	Merged the JHipster component finder i	nto the Spring component finder.	9 days ago
E .gitignore	Some quick experimenting with an expo	ort to DOT.	4 months ago
	Initial commit		2 years ago
E README.md	Tweaks to getting started example.		4 days ago
E build.gradle	Tweaks following merge.		3 months ago
gradle.properties	Renamed components and messing wit	h Maven repo publication.	a year ago
E) gradlew	Trimmed down the Gradle config and st	opped tracking the IDEA project	a year ago
gradlew.bat	Gradle support		a year ago
settings.gradle	Merged the JHipster component finder	nto the Spring component finder.	9 days ago
swagger.txt	Added the ability to customise where the	e diagram metadata is placed (	3 months ago

EE README.md

Structurizr Create web-based software architecture diagrams using code.

#### Structurizr for Java

GitHub This repository	Search Explore	e Features Enterpris	e Pricing	Sign up Sign in
structurizr / dotnet	🍸 Pull requests 0 🧄 Pulse 📊 Graphs		• Watch 4	★ Star 18 V Fork 1
Structurizr for .NET				
T 29 commits	β۶ <b>1</b> branch	$\bigcirc$ 0 releases		1 contributor
Branch: master - New pull	request New file Find file	e HTTPS - https:/	//github.com/stru	Download ZIP
😨 simonbrowndotje Fixed some broken links. Latest commit 3d5aac1 2 days ago				
Structurizr.Core	Well, they do say namng is hard			2 days ago
Structurizr.CoreTests	Well, they do say namng is hard			2 days ago
Structurizr.Examples	Added enough code to get a software architecture r	model for the "Conto		3 days ago
docs	Some doc tweaks.			4 days ago
.gitignore	Initial commit			9 days ago
	Initial commit			9 days ago
README.md	Fixed some broken links.			2 days ago
Structurizr.sln	Renamed some folders and added some skeleton of	code for the component f		3 days ago
CO DEADNE and				

E README.m

0

•

#### Structurizr Create web-based software

architecture diagrams using code.



#### Structurizr for .NET

This GitHub repository is a .NET library to create software architecture models that are compatible with Structurizr, a SaaS to create web-based software architecture diagrams. It's an implementation of the "C4 software architecture model", as <u>described</u> in Simon Brown's FREE The Art of Visualising Software Architecture book. In a nutshell:

- 1. Create a software architecture model using .NET code, either manually or by extracting information from an existing codebase.
- 2. Upload the model (as a JSON document) to Structurizr using the web API.
- 3. Visualise and share the resulting software architecture diagrams (example).
  - Upload the software architecture model to Structurizr using the web API

### Structurizr

#### Create web-based software architecture diagrams using code.

Getting Started with Java Getting started with .NET



### GOTO Copenhagen 2016

(promotional code for 1 month free Enterprise Plan; use by 15th April 2016)

What is Structurizr?



Software System

#### Structurizr Create web-based software architecture diagrams using code.





3. Explore, visualise and share your software architecture model with web-based software architecture diagrams Structurizr is a web-based tool that creates diagrams

for you, based upon the software architecture model and views defined in your code

### What is Structurizr?

```
class GettingStarted
    static void Main(string[] args)
    {
        Workspace workspace = new Workspace("My model", "This is a model of my software system.");
        Model.Model model = workspace.Model;
        Person user = model.AddPerson("User", "A user of my software system.");
        SoftwareSystem softwareSystem = model.AddSoftwareSystem("Software System", "My software system.");
        user.Uses(softwareSystem, "Uses");
        ViewSet viewSet = workspace.Views;
        SystemContextView contextView = viewSet.CreateContextView(softwareSystem);
        contextView.AddAllSoftwareSystems();
        contextView.AddAllPeople();
        Styles styles = viewSet.Configuration.Styles;
        styles.Add(new ElementStyle(Tags.SoftwareSystem) { Background = "#a4b7c9", Color = "#000000" });
        styles.Add(new ElementStyle(Tags.Person) { Background = "#728da5", Color = "#ffffff" });
        StructurizrClient structurizrClient = new StructurizrClient("key", "secret");
        structurizrClient.PutWorkspace(1234, workspace);
    }
}
                                                                                        Software System
              User
                                                      -Uses
                                                                                            [Software System]
              [Person]
    A user of my software system.
                                                                                          My software system.
```

```
static void Main(string[] args)
```

Workspace workspace = new Workspace("Financial Risk System", "A simple example C4 model based upon the financial risk system arc Model.Model model = workspace.Model;

// create the basic model
SoftwareSystem financialRiskSystem = model.AddSoftwareSystem(Location.Internal, "Financial Risk System", "Calculates the bank's

```
Person businessUser = model.AddPerson(Location.Internal, "Business User", "A regular business user");
businessUser.Uses(financialRiskSystem, "Views reports using");
```

Person configurationUser = model.AddPerson(Location.Internal, "Configuration User", "A regular business user who can also config configurationUser.Uses(financialRiskSystem, "Configures parameters using");

SoftwareSystem tradeDataSystem = model.AddSoftwareSystem(Location.Internal, "Trade Data System", "The system of record for trade
financialRiskSystem.Uses(tradeDataSystem, "Gets trade data from");

SoftwareSystem referenceDataSystem = model.AddSoftwareSystem(Location.Internal, "Reference Data System", "Manages reference data
financialRiskSystem.Uses(referenceDataSystem, "Gets counterparty data from");

SoftwareSystem emailSystem = model.AddSoftwareSystem(Location.Internal, "E-mail system", "Microsoft Exchange");
financialRiskSystem.Uses(emailSystem, "Sends a notification that a report is ready to");
emailSystem.Delivers(businessUser, "Sends a notification that a report is ready to", "E-mail message", InteractionStyle.Asynchro

SoftwareSystem centralMonitoringService = model.AddSoftwareSystem(Location.Internal, "Central Monitoring Service", "The bank-wid financialRiskSystem.Uses(centralMonitoringService, "Sends critical failure alerts to", "SNMP", InteractionStyle.Asynchronous).Ad

SoftwareSystem activeDirectory = model.AddSoftwareSystem(Location.Internal, "Active Directory", "Manages users and security role



## Spring PetClinic

https://github.com/spring-projects/spring-petclinic/

		Sprir s o u r
Views	JSP with custom tags	Thymeleaf
+ Bootstrap (CSS)	&& webjars	&& Dandelion
Controller		Spring @MVC annotations
		Bean Validation
Service		@Cacheable
		@Transactional
_		Spring Data JPA
Repository	3 profile	es default (JPA)
		jdbc

https://speakerdeck.com/michaelisvy/ spring-petclinic-sample-application

	org.springframework.samples.petclinic
$\overline{\mathbf{w}}$	💼 model
	C & BaseEntity
	C & NamedEntity
	C & Owner
	C & Person
	C & Pet
	C & PetType
	C & Specialty
	C a Vet
	C a Vets
	C a Visit
$\overline{\mathbf{w}}$	repository
	🔻 🛅 jdbc
	C a JdbcOwnerRepositoryImpl
	© ∘ JdbcPet
	C & JdbcPetRepositoryImpl
	Subscription of the second
	C & JdbcVetRepositoryImpl
	C & JdbcVisitRepositoryImpl
	① & OwnerRepository
	B PetRepository
	b VetRepository
	b VisitRepository
v	service
	ClinicService
	ClinicServiceImpl
v	■ util
	CallMonitoringAspect
_	C EntityUtils
v	• web
	CrashController
	Controller
	PetController
	PetTypeFormatter
	PetValidator
	S VetController

```
**
* This is a C4 representation of the Spring PetClinic sample app
* (https://github.com/spring-projects/spring_petclinic/).
*
* Use the examples/springpetclinic.sh file to run this example -
* you'll need a compiled version of the app on the CLASSPATH.
*/
public class SpringPetClinic {
```

#### public class SpringPetClinic {

// create the basic model (the stuff we can't get from the code)

SoftwareSystem springPetClinic = model.addSoftwareSystem("Spring PetClinic", "Allows employees to view
Person user = model.addPerson("Clinic Employee", "An employee of the clinic");
user.uses(springPetClinic, "Uses");

```
Container webApplication = springPetClinic.addContainer(
```

```
"Web Application", "Allows employees to view and manage information regarding the veterinarians
Container relationalDatabase = springPetClinic.addContainer(
```

```
"Relational Database", "Stores information regarding the veterinarians, the clients, and their
user.uses(webApplication, "Uses", "HTTP");
webApplication.uses(relationalDatabase. "Reads from and writes to". "JDBC. port 9001"):
```

```
// and now automatically find all Spring @Controller, @Component, @Service and @Repository components
ComponentFinder componentFinder = new ComponentFinder(
```

webApplication, "org.springframework.samples.petclinic",

new SpringComponentFinderStrategy(),

new JavadocComponentFinderStrategy(new File("/Users/simon/Documents/sandbox/spring-petc componentFinder.findComponents();

// connect all of the repository components to the relational database

```
>/**
 * This is a C4 representation of the Spring PetClinic sample app
 * (https://github.com/spring-projects/spring-petclinic/).
 *
```

\* Use the examples/springpetclinic.sh file to run this example \* you'll need a compiled version of the app on the CLASSPATH.

#### public class SpringPetClinic {

\*/

#### public static void main(String[] args) throws Exception {

Workspace workspace = **new** Workspace("Spring PetClinic",

"This is a C4 representation of the Spring PetClinic sample app (https://github.com/spring-pro Model model = workspace.getModel();

// create the basic model (the stuff we can't get from the code)
SoftwareSystem springPetClinic = model.addSoftwareSystem("Spring PetClinic", "Allows employees to view
Person user = model.addPerson("Clinic Employee", "An employee of the clinic");
user.uses(springPetClinic, "Uses");

#### Container webApplication = springPetClinic.addContainer(

"Web Application", "Allows employees to view and manage information regarding the veterinarians
Container relationalDatabase = springPetClinic.addContainer(

"Relational Database", "Stores information regarding the veterinarians, the clients, and their user.uses(webApplication, "Uses", "HTTP"); webApplication.uses(relationalDatabase, "Reads from and writes to", "JDBC, port 9001");

// and now automatically find all Spring @Controller, @Component, @Service and @Repository components
ComponentFinder componentFinder = new ComponentFinder(

webApplication, "org.springframework.samples.petclinic",

new SpringComponentFinderStrategy(),

new JavadocComponentFinderStrategy(new File("/Users/simon/Documents/sandbox/spring-petc componentFinder.findComponents();

```
// connect the user to all of the Spring MVC controllers
webApplication.getComponents().stream()
    .filter(c -> c.getTechnology().equals("Spring MVC Controller"))
    .forEach(c -> user.uses(c, "Uses"));
```

// connect all of the repository components to the relational database

## "Component Finder" with pluggable strategies, implemented using reflection & static analysis (e.g. Java Annotations, .NET Attributes, type name ends with "Controller", type extends class x, type implements interface Y, supplement model with type-level comments

from source code, etc)

// and now automatically find all Spring @Controller, @Component, @Service and @Repository components
ComponentFinder componentFinder = new ComponentFinder(

webApplication, "org.springframework.samples.petclinic",

new SpringComponentFinderStrategy(),

new JavadocComponentFinderStrategy(new File("/Users/simon/Documents/sandbox/spring\_petc componentFinder.findComponents();

// connect all of the repository components to the relational database webApplication.getComponents().stream()

```
.filter(c -> c.getTechnology().equals("Spring Repository"))
```

.forEach(c -> c.uses(relationalDatabase, "Reads from and writes to"));

```
// finally create some views
ViewSet viewSet = workspace.getViews();
SystemContextView contextView = viewSet.createContextView(springPetClinic)
contextView.addAllSoftwareSystems();
contextView.addAllPeople();
```

```
ContainerView containerView = viewSet.createContainerView(springPetClinic);
containerView.addAllPeople();
containerView.addAllSoftwareSystems();
containerView.addAllContainers();
```

```
ComponentView componentView = viewSet.createComponentView(webApplication);
componentView.addAllComponents();
componentView.addAllPeople();
componentView.add(relationalDatabase):
```

// and now automatically find all Spring @Controller, @Component, @Service and @Repository components
ComponentFinder componentFinder = new ComponentFinder(

webApplication, "org.springframework.samples.petclinic"

new SpringComponentFinderStrategy(),

new JavadocComponentFinderStrategy(new File("/Users/simon/Documents/sandbox/spring/spring\_petc componentFinder.findComponents();

// connect all of the repository components to the relational database
webApplication.getComponents().stream()

.filter(c -> c.getTechnology().equals("Spring Repository"))

.forEach(c -> c.uses(relationalDatabase, "Reads from and writes to"));

```
// finally create some views
ViewSet viewSet = workspace.getViews();
SystemContextView contextView = viewSet.createContextView(springPetClinic);
contextView.addAllSoftwareSystems();
contextView.addAllPeople();
```

```
ContainerView containerView = viewSet.createContainerView(springPetClinic);
containerView.addAllPeople();
containerView.addAllSoftwareSystems();
containerView.addAllContainers();
```

```
ComponentView componentView = viewSet.createComponentView(webApplication);
componentView.addAllComponents();
componentView.addAllPeople();
componentView.add(relationalDatabase);
```
// and now automatically find all Spring @Controller, @Component, @Service and @Repository components
ComponentFinder componentFinder = new ComponentFinder(

```
webApplication, "org.springframework.samples.petclinic",
```

```
new SpringComponentFinderStrategy(),
```

new JavadocComponentFinderStrategy(new File("/Users/simon/Documents/sandbox/spring/spring\_petc componentFinder.findComponents();

```
// connect the user to all of the Spring MVC controllers
webApplication.getComponents().stream()
    .filter(c -> c.getTechnology().equals("Spring MVC Controller"))
    .forEach(c -> user.uses(c, "Uses"));
```

```
// connect all of the repository components to the relational database
webApplication.getComponents().stream()
    .filter(c -> c.getTechnology().equals("Spring Repository"))
    .forEach(c -> c.uses(relationalDatabase, "Reads from and writes to"))
```

```
// finally create some views
ViewSet viewSet = workspace.getViews();
SystemContextView contextView = viewSet.createContextView(springPetClinic);
contextView.addAllSoftwareSystems();
contextView.addAllPeople();
```

```
ContainerView containerView = viewSet.createContainerView(springPetClinic);
containerView.addAllPeople();
containerView.addAllSoftwareSystems();
containerView.addAllContainers();
```

```
ComponentView componentView = viewSet.createComponentView(webApplication);
componentView.addAllComponents();
componentView.addAllPeople();
componentView.add(relationalDatabase);
```

### Spring PetClinic - System Context



Spring PetClinic

[Software System] Allows employees to view and manage information regarding the veterinarians, the clients, and their pets.

#### **Spring PetClinic - Containers**





Spring PetClinic - Web Application

- Components



## **Spring PetClinic - Web Application - Components**



Close

Х



<pre>matter</pre>		
<pre>g-petclinic / src / main / java / org / springframework / samples / petclinic / service / ClinicService.java mothewing test methods used should blooud blooud blooud blooud blood bl</pre>	ch:	master - 🗄 🖻
<pre>mtheduity test methodsused should/should/sou herdbuirs introducto</pre>	ng	-petclinic / src / main / java / org / springframework / samples / petclinic / service / ClinicService.java
<pre>numbulos is in the second second</pre>	mic	haelisvy test methods:used should/shouldNot 5c9ab6 on 16 Jan
<pre>Numes (38 stoc) 1.67 K8 Rev Prior Control Rev Prior Rev Prior</pre>	ontri	butors 🏭 👧
<pre>/*  * Copyright 2002-2013 the original author or authors.  *  *   * Licensed under the Apache License, Version 2.0 (the "License");  * you may not use this file except in compliance with the License.  * Tou may obtain a copy of the License at  *  *   *   *   *   *   *   *   *   *</pre>	lin	es (38 sloc) 1.67 KB Raw Blame History 🖵 🧳 📋
<pre>2 * Copyright 2002-2013 the original author or authors. * ticensed under the Apache License, Version 2.0 (the "License"); * you may obtain a copy of the License at * http://www.apache.org/licenses/LICENSE-2.0 * thuless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" PASIS, * WITHOUT WARAWTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License is distributed on an "AS IS" PASIS, * WITHOUT WARAWTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * Linitations under the License. */ package org.springframework.samples.petclinic.service; import org.springframework.samples.petclinic.model.Owner; import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Vet; impo</pre>	1	/*
<pre>4 * Licensed under the Apache License, Version 2.8 (the "License"); * you may obtain a copy of the License at * these required by applicable law or agreed to in writing, software * this://www.apache.org/licenses/LICENSE-2.0 * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS DS* BASIS, * WTHOUT MARAWITES DK CONDITIONS OF AWY KIND, either express or inplied. * See the License for the specific language governing permissions and * Limitations under the License. */ package org.springframework.samples.petclinic.service; Import org.springframework.samples.petclinic.model.Omer; import org.springframework.samples.petclinic.model.Net; import org.springframework.samples.petclinic.model.Net; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Vet; void saw</pre>	2	* Lopyright 2002-2013 the original author or authors. *
<pre>5 *you may not use this file except in compliance with the License. *You may obtain a copy of the License at * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software = distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARAWITES OR CONDITIONS OF ANY KND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. */ package org.springframework.samples.petclinic.service; import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Vet; i</pre>	4	* Licensed under the Apache License, Version 2.0 (the "License");
<pre>6 * You may obtain a copy of the License at 7 8 http://www.apache.org/licenses/LICENSE-2.0 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9</pre>	5	st you may not use this file except in compliance with the License.
<pre>* http://www.apache.org/licenses/LICENSE-2.0 *  thess required by applicable law or agreed to in writing, software idistributed under the License is distributed on an "AS IS" BASIS, idistributed under the License is distributed on an "AS IS" BASIS, * WTHOONT WARAWTIES BO CONDITIONS OF AWY KND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. */ package org.springframework.samples.petclinic.service; import org.springframework.samples.petclinic.model.owner; import org.springframework.samples.petclinic.model.Ver; import org.springframework.samples.petclinic.model.Ver; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Visit; //* * Mostly used as a facade so all controllers have a single point of entry * # Quurthor Michael Isvy */ public interface ClinicService { Collection<pettypes collection<petry="" dataac<="" dataaccessexception;="" findownerbyid(int="" findpettype()="" findpettypes()="" id)="" owner="" pet)="" sawepet(pet="" td="" throws="" void=""><td>6</td><td>* You may obtain a copy of the License at</td></pettypes></pre>	6	* You may obtain a copy of the License at
<pre>* Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRWITES OR CONDITIONS OF AVY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. */ package org.springframework.samples.petclinic.service; import org.springframework.doo.DataAccessException; import org.springframework.samples.petclinic.model.Merr; import org.springframework.samples.petclinic.model.PetType; import org.springframework.samples.petclinic.model.PetType; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Vet; void savelow returnes of the second state conset second state c</pre>	8	<pre>http://www.apache.org/licenses/LICENSE-2.0</pre>
<pre>9 * Unless required by applicable law or agreed to in writing, software 1 * distributed under the License is distributed on an "AS IS" BASIS, * WTHOOT MARAWTIES OR COMDITIONS OF ANY KIND, either express or implied. 3 * See the License for the specific language governing permissions and * limitations under the License. */ package org.springframework.samples.petclinic.service; 1 import org.springframework.samples.petclinic.model.owner; 1 import org.springframework.samples.petclinic.model.Pet; 1 import org.springframework.samples.petclinic.model.Pet; 1 import org.springframework.samples.petclinic.model.Vet; 1 import org.springframework.samples.petclinic.model.Vet; 1 import org.springframework.samples.petclinic.model.Vet; 1 import org.springframework.samples.petclinic.model.Vet; 1 import org.springframework.samples.petclinic.model.Visit; /** * Mostly used as a facade so all controllers have a single point of entry */ public interface ClinicService { Collection-PetType&gt; findPetTypes() throws DataAccessException; Void savePet(Pet pet) throws DataAccessException; void saveVisit(Visit visit) throws DataAccessException; Collection-Vet&gt; findVetS() throws DataAccessException; Void saveOwner(Owner owner) throws DataAccessException; Collection-Vet&gt; findVetS() throws DataAccessExcep</pre>	9	*
<pre>i distributed under the License is distributed on an "AS IS" BASIS,  * WITHOUT WARAWITES OR CONDITIONS OF ANY KIND, either express or inplied.  * See the License for the specific language governing permissions and  * limitations under the License.  */ package org.springframework.samples.petclinic.service;  import org.springframework.dao.DataAccessException; import org.springframework.samples.petclinic.model.Owner; import org.springframework.samples.petclinic.model.Owner; import org.springframework.samples.petclinic.model.Owner; import org.springframework.samples.petclinic.model.Net; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Visit; /**  * Mostly used as a facade so all controllers have a single point of entry  *  * gauthor Michael Isvy  */ public interface ClinicService {  Collection-PetType&gt; findPetTypes() throws DataAccessException;  Void saveVisit(Visit visit) throws DataAccessException;  void saveVisit(Visit visit) throws DataAccessException;  collection-Vet&gt; findOwnerByLastMame(String LastMame) throws DataAccessException;  void saveOwner(Owner owner) throws DataAccessException;  collection-Owner&gt; findOwnerByLastMame(String LastMame) throws DataAccessException;  void saveOwner(Owner owner) throws DataAccessException;  collection-Owner&gt; findOwnerByLastMame(String LastMame) throws DataAccessException; </pre>	10	* Unless required by applicable law or agreed to in writing, software
<pre>* See the License for the specific language governing permissions and * See the License for the specific language governing permissions and * limitations under the License. */ package org.springframework.samples.petclinic.service; import org.springframework.dao.DataAccessException; import org.springframework.samples.petclinic.model.owner; import org.springframework.samples.petclinic.model.Net; import org.springframework.samples.petclinic.model.Net; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Vet; infort org.springframework.samples.petclinic.model.Vet; infort org.springframework.samples.petclinic.model.Vet; infort org.springframework.samples.petclinic.model.Vet; infort org.springframework.samples.petclinic.model.Vet; infort org.springframework.samples.petclinic.model.Vet; infort org.springframework.samples.petclinic.model.Vet; infort org.springfra</pre>	11	* distributed under the License is distributed on an "AS IS" BASIS,
<pre>* limitations under the License. */ package org.springframework.samples.petclinic.service; import java.util.Collection; import org.springframework.doo.DataAccessException; import org.springframework.samples.petclinic.model.Owner; import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Pet;; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Visit; /**  * Mostly used as a facade so all controllers have a single point of entry  *  * Gauthor Michael Isvy  */ public interface ClinicService {  Collection-PetType&gt; findPetTypes() throws DataAccessException;  Owner findOwnerById(int id) throws DataAccessException;  void saveOwner(Owner owner) throws DataAccessException;  collection-Vet&gt; findOwnerByLastName(String lastName) throws DataAccessException; </pre>	13	* minious manyowitco on complitumo or ANN Kind, element express or implied. * See the License for the specific language governing permissions and
<pre>*/ package org.springframework.samples.petclinic.service; import java.util.Collection; import org.springframework.samples.petclinic.model.Owner; import org.springframework.samples.petclinic.model.Net; import org.springframework.samples.petclinic.model.Pett; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Visit; /**  * Mostly used as a facade so all controllers have a single point of entry  *  * @author Michael Isvy  *  * @author Solute.compenters findPetFypes() throws DataAccessException;  Owner findOwnerById(int id) throws DataAccessException;  void saveVisit(Visit visit) throws DataAccessException;  void saveOwner(Owner owner) throws DataAccessException;  void</pre>	14	* limitations under the License.
<pre>beckage org.springframework.samples.petclinic.service; import java.util.Collection; import org.springframework.doo.DataAccessException; import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Vet; out saveOuterPetType&gt; findPetTypes() throws DataAccessException; void saveOuter(Owner owner) throws DataAccessException; void saveOwner(Owner owner) throws DataAccessException; collection-Owner&gt; findOwnerByLastName(String lastName) throws DataAccessException;</pre>	15	*/
<pre>import java.util.Collection; import org.springframework.dao.DataAccessException; import org.springframework.samples.petclinic.model.Owner; import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Vet; import microscopy of throws DataAccessException; void saveOuter/Petpet findVets() throws DataAccessException; void saveOuter(Owner owner) throws DataAccessException; collection-Owner&gt; findOwnerByLastName(String lastName) throws DataAccessException;</pre>	16	<pre>package org.springframework.samples.petclinic.service;</pre>
<pre>import org.springframework.do.DataAccessException; import org.springframework.samples.petclinic.model.Owner; import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Visit; /** * Mostly used as a facade so all controllers have a single point of entry * @author Michael Isvy */ public interface ClinicService { Collection<pettype> findPetTypes() throws DataAccessException; Owner findOwnerById(int id) throws DataAccessException; Pet findPetById(int id) throws DataAccessException; void savePet(Pet pet) throws DataAccessException; Collection<vet> findOvter owner) throws DataAccessException; Collection<vet> findOvter owner) throws DataAccessException; Collection<vet> findOvter owner) throws DataAccessException; Collection<vet> findOvterByLatName(String LastName) throws DataAccessException;</vet></vet></vet></vet></vet></vet></vet></vet></vet></vet></vet></vet></pettype></pre>	18	import java.util.Collection:
<pre>import org.springframework.dao.DataAccessException; import org.springframework.samples.petclinic.model.Demer; import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Visit; /** * Mostly used as a facade so all controllers have a single point of entry * * @author Michael Isvy */ public interface ClinicService { Collection<pettype> findPetTypes() throws DataAccessException; Owner findOwnerById(int id) throws DataAccessException; Pet findPetById(int id) throws DataAccessException; void savePet(Pet pet) throws DataAccessException; collection<vet> findOwner DataAccessException; void saveOwner(Owner owner) throws DataAccessException; collection<owner> findOwnerByLatName(String lastName) throws DataAccessException; collection<owner> findOwnerByLatName(String lastName) throws DataAccessException; collection<owner> findOwnerByLatName(String lastName) throws DataAccessException; collection<owner> findOwnerByLatName(String lastName) throws DataAccessException; collection<owner> findOwnerByLastName(String lastName) throws DataAccessException; collection</owner></owner></owner></owner></owner></vet></pettype></pre>	19	
<pre>import org.springframework.samples.petclinic.model.Dwner; import org.springframework.samples.petclinic.model.Petry import org.springframework.samples.petclinic.model.Petrype; import org.springframework.samples.petclinic.model.Vetry import org.springframework.samples.petclinic.model.Vetry index.petcletery findVetry findVetry findVetry DataAccessException; void saveOwner(Owner owner) throws DataAccessException; Collection-Qwner&gt; findOwnerByLastName(String LastName) throws DataAccessException;</pre>	20	<pre>import org.springframework.dao.DataAccessException;</pre>
<pre>2 Import org.springframework.samples.petclinic.model.Pet; import org.springframework.samples.petclinic.model.Vet; import org.springframework.setRetersers.setRete</pre>	21	<pre>import org.springframework.samples.petclinic.model.Owner;</pre>
<pre>import org.springframework.samples.petclinic.model.Vet; import org.springframework.samples.petclinic.model.Visit; /** * Mostly used as a facade so all controllers have a single point of entry * autor Michael Isvy */ public interface ClinicService { Collection<pettype> findPetTypes() throws DataAccessException; Owner findOwmerById(int id) throws DataAccessException; Pet findPetById(int id) throws DataAccessException; void savePet(Pet pet) throws DataAccessException; Collection<vet> findOvter SutaAccessException; Collection<vet> findVets() throws DataAccessException; void saveOwner(Owner owner) throws DataAccessException; Collection<vet> findOvterByLatName(String lastName) throws DataAccessException;</vet></vet></vet></pettype></pre>	22	<pre>import org.springriamework.samples.petclinic.model.PetType:</pre>
<pre>import org.springframework.samples.petclinic.model.Visit; /**  * Mostly used as a facade so all controllers have a single point of entry  *  * @author Michael Isvy  */ public interface ClinicService {  Collection<pettype> findPetTypes() throws DataAccessException;  Owner findOwnerById(int id) throws DataAccessException;  Pet findPetById(int id) throws DataAccessException;  void savePet(Pet pet) throws DataAccessException;  void savePet(Pet pet) throws DataAccessException;  void saveOwner(Owner owner) throws DataAccessException;  Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException;</owner></pettype></pre>	24	<pre>import org.springframework.samples.petclinic.model.Vet;</pre>
<pre>/**     /**     /**     /**     * Mostly used as a facade so all controllers have a single point of entry     *     *     @uthor Michael Isvy     */     public interface ClinicService {         Collection<pettype> findPetTypes() throws DataAccessException;         Owner findOwmerById(int id) throws DataAccessException;         Owner findOwmerById(int id) throws DataAccessException;         void savePet(Pet pet) throws DataAccessException;         void savePet(Pet pet) throws DataAccessException;         void saveVisit(Visit visit) throws DataAccessException;         void saveOwner(Owner owner) throws DataAccessException;         void saveOwner(Owner owner) throws DataAccessException;         Collection<owner> findOwmerByLastName(String lastName) throws DataAccessException;         Void SaveOwner(Owner&gt; findOwnerByLastName(String lastName) throws DataAccessException;         Void saveOwner(DataAccessException;         Void saveOwner(DataAccessException;         Void saveOwner(DataAc</owner></pettype></pre>	25	<pre>import org.springframework.samples.petclinic.model.Visit;</pre>
<pre>/** /** /** * Mostly used as a facade so all controllers have a single point of entry * Mostly used as a facade so all controllers have a single point of entry * @ * Mostly used as a facade so all controllers have a single point of entry * @ */ public interface ClinicService { Collection<pettype> findPetTypes() throws DataAccessException; Owner findOwnerById(int id) throws DataAccessException; Pet findPetById(int id) throws DataAccessException; void savePet(Pet pet) throws DataAccessException; Collection<vet> findOwner DataAccessException; Void saveOwner(Owner owner) throws DataAccessException; Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException; </owner></vet></pettype></pre>	26	
<pre>* Mostly used as a facade so all controllers have a single point of entry * * Mostly used as a facade so all controllers have a single point of entry * * @author Michael Isvy */ public interface ClinicService { Collection<pettype> findPetTypes() throws DataAccessException; Owner findOwnerById(int id) throws DataAccessException; Pet findPetById(int id) throws DataAccessException; void savePet(Pet pet) throws DataAccessException; Collection<vet> findOwner DataAccessException; Collection<vet> findVets() throws DataAccessException; Collection<vet> findVets() throws DataAccessException; Collection<owner collection<owner="" dataaccessexception;="" owner)="" throws=""> findOwnerByLastName(String lastName) throws DataAccessException;</owner></vet></vet></vet></pettype></pre>	28	/**
<pre>* * * * * * * * * * * * * * * * * * * *</pre>	29	* Mostly used as a facade so all controllers have a single point of entry
<pre># (Bauthor Michael Isvy #/ public interface ClinicService { Collection<pettype> findPetTypes() throws DataAccessException; Owner findOwnerById(int id) throws DataAccessException; Pet findPetById(int id) throws DataAccessException; void savePet(Pet pet) throws DataAccessException; Collection<vet> findVets() throws DataAccessException; Collection<vet> findVets() throws DataAccessException; Collection<owner> findOwnerByLastName(String LastName) throws DataAccessException;</owner></vet></vet></pettype></pre>	30	*
public interface ClinicService {         Collection <pettype> findPetTypes() throws DataAccessException;         Owner findOwnerById(int id) throws DataAccessException;         Pet findPetById(int id) throws DataAccessException;         void savePet(Pet pet) throws DataAccessException;         Collection<vet> findVets() throws DataAccessException;         void saveVisit(Visit visit) throws DataAccessException;         Collection<vet> findVets() throws DataAccessException;         void saveOwner(Owner owner) throws DataAccessException;         Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException;</owner></vet></vet></pettype>	31	* @author Michael Isvy
Collection <pettype> findPetTypes() throws DataAccessException; Owner findOwnerById(int id) throws DataAccessException; Pet findPetById(int id) throws DataAccessException; void savePet(Pet pet) throws DataAccessException; Collection<vet> findVets() throws DataAccessException; Collection<vet> findVets() throws DataAccessException; Collection<owner dataaccessexception;<br="" owner)="" throws="">Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException;</owner></owner></vet></vet></pettype>	33	public interface ClinicService {
Collection <pettype> findPetTypes() throws DataAccessException; Owner findOwnerById(int id) throws DataAccessException; Pet findPetById(int id) throws DataAccessException; void savePet(Pet pet) throws DataAccessException; void saveVisit(Visit visit) throws DataAccessException; Collection<vet> findVets() throws DataAccessException; Void saveOwner(Owner owner) throws DataAccessException; Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException;</owner></vet></pettype>	34	
90         Owner findOwnerById(int id) throws DataAccessException;         91         92         93         94         95         95         96         97         98         98         99         90         91         92         93         94         95         96         97         98         98         99         90         90         91         92         93         94         94         95         96         97         98         99         90         91         92         94         95         96         97         98	35	Collection <pettype> findPetTypes() throws DataAccessException;</pettype>
Pet findPetById(int id) throws DataAccessException; void savePet(Pet pet) throws DataAccessException; void saveVisit(Visit visit) throws DataAccessException; Collection <vet> findVets() throws DataAccessException; Void saveOwner(Owner owner) throws DataAccessException; Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException;</owner></vet>	36 37	Owner findOwnerById(int id) throws DataAccessExcention:
Pet findPetById(int id) throws DataAccessException; void savePet(Pet pet) throws DataAccessException; void saveVisit(Visit visit) throws DataAccessException; Collection <vet> findVets() throws DataAccessException; void saveOwner(Owner owner) throws DataAccessException; Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException;</owner></vet>	38	which randomeroyacyane adj chrona busineeeaacheeptanij
<pre>void savePet(Pet pet) throws DataAccessException; void saveVisit(Visit visit) throws DataAccessException; Collection<vet> findVets() throws DataAccessException; void saveOwner(Owner owner) throws DataAccessException; Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException; ind</owner></vet></pre>	39	<pre>Pet findPetById(int id) throws DataAccessException;</pre>
<pre>42 42 43 44 44 45 46 46 47 47 46 47 47 47 48 48 49 49 40 40 40 40 40 40 40 40 40 40 40 40 40</pre>	40 41	<pre>void savePet(Pet pet) throws DataAccessException;</pre>
<pre>void saveVisit(Visit visit) throws DataAccessException; Collection<vet> findVets() throws DataAccessException; void saveOwner(Owner owner) throws DataAccessException; Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException; Second Second Second</owner></vet></pre>	42	
Collection <vet> findVets() throws DataAccessException; void saveOwner(Owner owner) throws DataAccessException; Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException;</owner></vet>	43 44	<pre>void saveVisit(Visit visit) throws DataAccessException;</pre>
<pre>void saveOwner(Owner owner) throws DataAccessException; Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException; </owner></pre>	45	Collection <vet> findVets() throws DataAccessException;</vet>
<pre>B8 S0 Collection<owner> findOwnerByLastName(String lastName) throws DataAccessException; S0 S0</owner></pre>	47	<pre>void saveOwner(Owner owner) throws DataAccessException;</pre>
50	48 49	Collection <dwner> findOwnerByLastName(String lastName) throws DataAccessException;</dwner>
	50	

0

Explore Features Enterprise Pricing

© 2015 GitHub, Inc. Terms Privacy Security Contact Help

GitHub This repository

Status API Training Shop Blog About Pricing

Sign up Sign in



# Diagrams are maps







## Creating the model as code provides opportunities...

Once you have a model, you can export that model and visualise it however you like...







### What can the different users do with which software





Build pipeline integration keeps software architecture models up-to-date



Visualising software architecture is still very much an *art*, but it's 2016 and time to stop using tools like Microsoft Visio!

Inanks.



simon.brown@codingthearchitecture.com @simonbrown on Twitter