

# Distributed and Collaborative Software Evolution Analysis with Churrasco

---

Marco D'Ambros  
Michele Lanza

Science of Computer Programming Journal  
Abril 2010

Pedro Araujo Pires

# Introdução

---

- Análise da evolução de um software é uma atividade complexa
- Modelar a evolução de sistemas implica:
  - na obtenção de dados de repositórios de software, como Git ou SVN
  - no *parsing* dos dados para extrair as informações relevantes
  - popular os modelos que serão usados para as análises.
- **Ferramentas de apoio à análise da evolução de um software devem esconder essas tarefas dos usuários.**

# Introdução

---

- Além disso, essas ferramentas devem quebrar a complexidade das informações analisadas.
- Qualquer ferramenta de análise de evolução de software deveria ter as seguintes características:
  - Modelo flexível
  - Acessibilidade
  - Armazenamento incremental de resultados
  - Suporte para colaboração

# Churrasco

---

- Esconde as tarefas de obtenção e processamento de dados dos usuários.
- Possui um mecanismo de persistência extensível e flexível.
- Permite análise colaborativa ao permitir que os usuários insiram anotações nos dados analisados.
- Armazena os resultados em um banco de dados central, permitindo que outros usuários tenham acesso a resultados anteriores.

# Churrasco *Components*

---

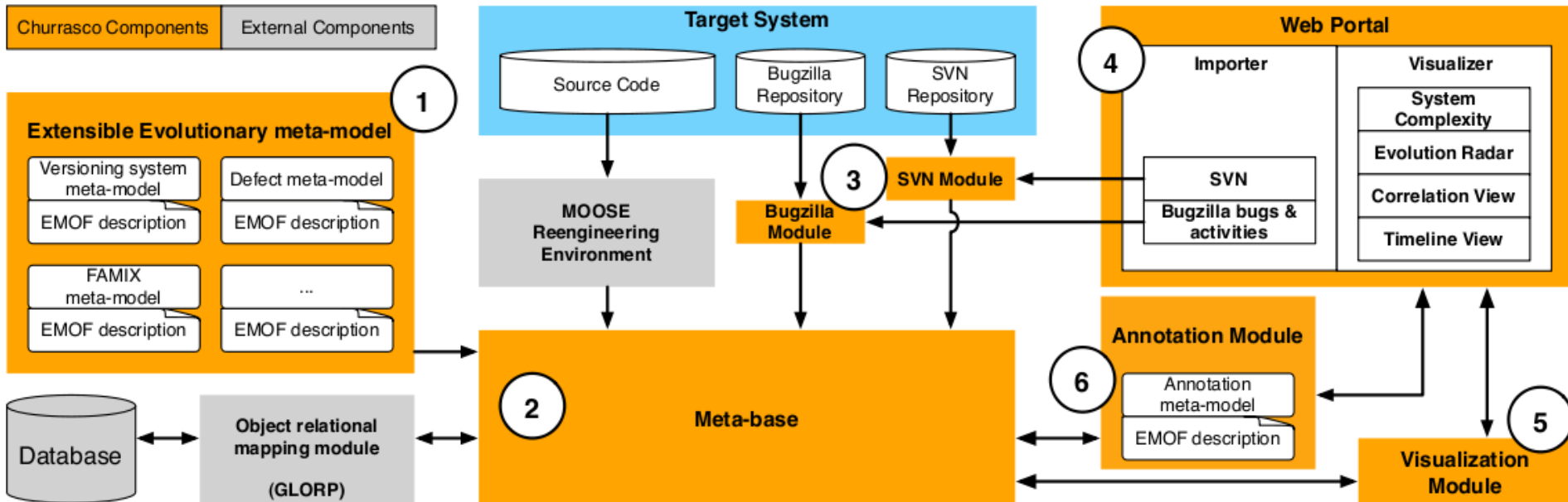
- Meta-modelo evolucionário extensível
  - Descreve a representação interna da evolução dos sistemas de software.
- Meta-base
  - Provê persistência relacional para ler/escrever do banco de dados
  - Utiliza Moose para gerar uma representação do código fonte baseada no modelo FAMIX.
- Módulos Bugzilla e SVN
  - Busca e processa dados de repositórios Bugzilla e SVN.

# Churrasco *Components*

---

- Módulo de visualização
  - Permite a análise de sistemas criando visualizações que utilizam Scalable Vector Graphics (SVG) interativos.
  
- Módulo de anotações
  - Permite análise colaborativa ao permitir adicionar anotações nas entidades sendo analisadas.
  - Comunicação com o módulo de visualizações, para retratar as anotações dentro das visualizações.

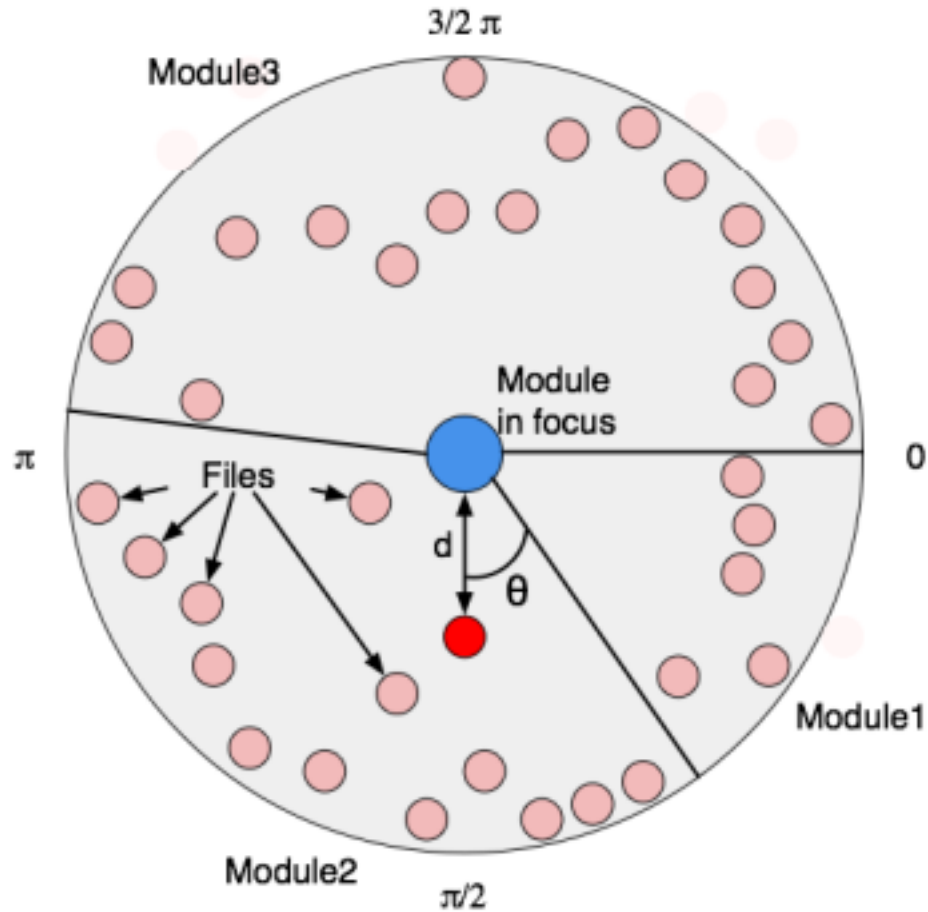
# Churrasco - Arquitetura



# Churrasco - Visualizações

---

## Evolution Radar

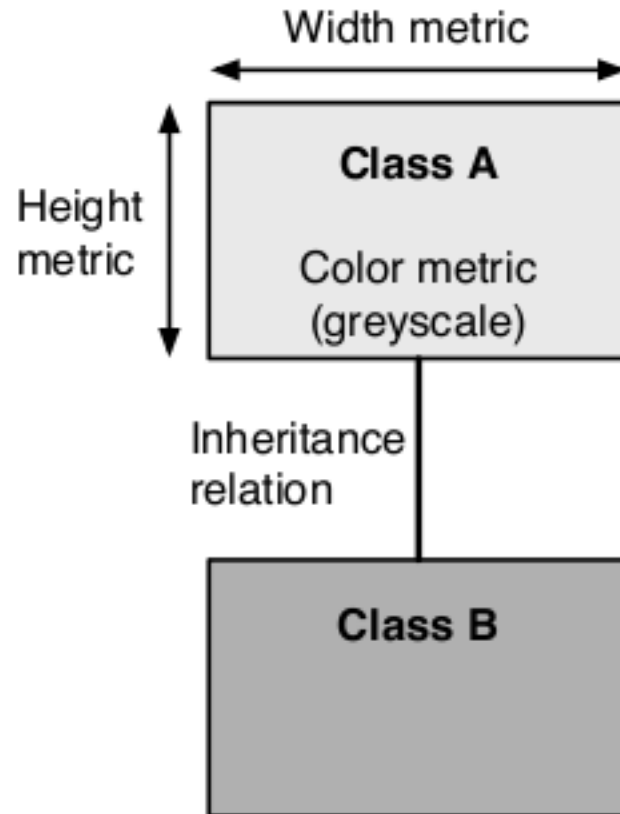




# Churrasco - Visualizações

---

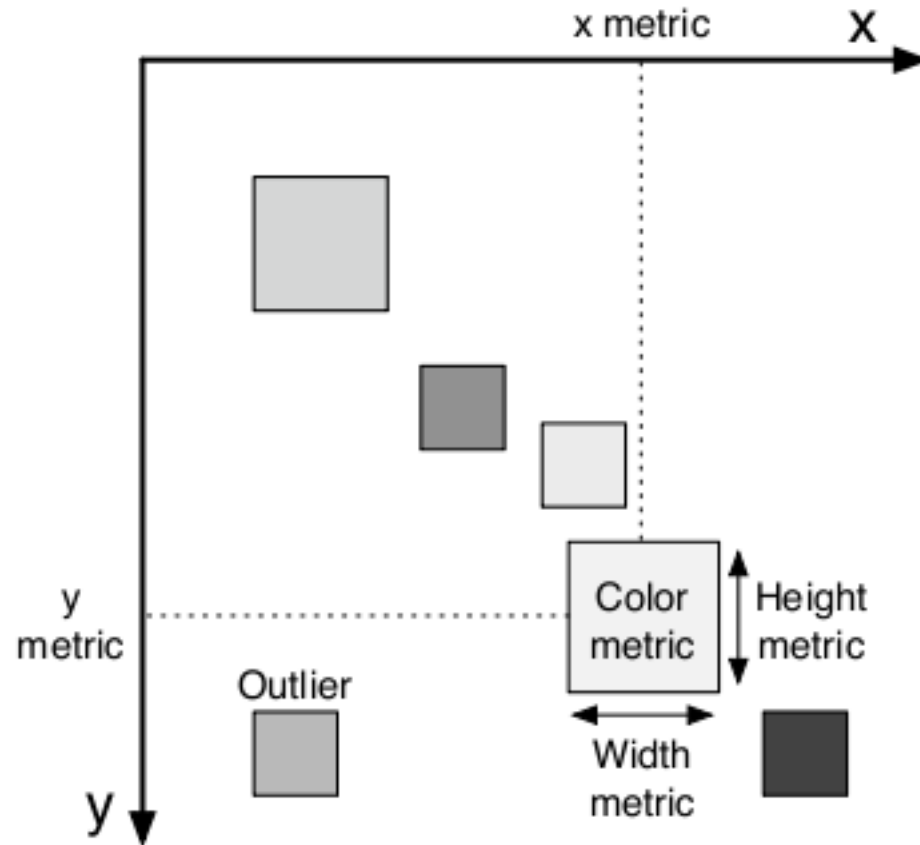
## System Complexity



# Churrasco - Visualizações

---

## Correlation View



# Churrasco in action

---

## Cenário:

- Michele e Marco estudando ArgoUML
- Michele analisa o sistema inteiro
- Marco foca no pacote *model*

# Churrasco in action

---

## System Complexity View

1. Marco cria uma anotação dizendo que a classe *FacadeMDRImpl*, do pacote *model*, apresenta sintomas de *bad design*:
  - 350 métodos
  - 3400 LOC
  - 3 atributos
  - Única classe que implementa a interface *Facade*
2. Michele adiciona uma segunda anotação dizendo que a observação de Marco é válida para o sistema inteiro, e que a classe *FacadeMDRImpl* é a que possui maior número de métodos no sistema inteiro.

# Churrasco in action

---

## System Complexity View

3. Marco vê que várias classes na hierarquia da classe *Factory* implementam a interface *Factory* e também herdam de classes que pertencem à hierarquia da *AbstractModelFactory*.
  4. Essas informações não estão disponíveis para Michele, que as descobre selecionando as entidades anotadas por Marco, e lendo as anotações.
- Esses problemas de design sempre estiveram presentes no sistema?

# Churrasco in action

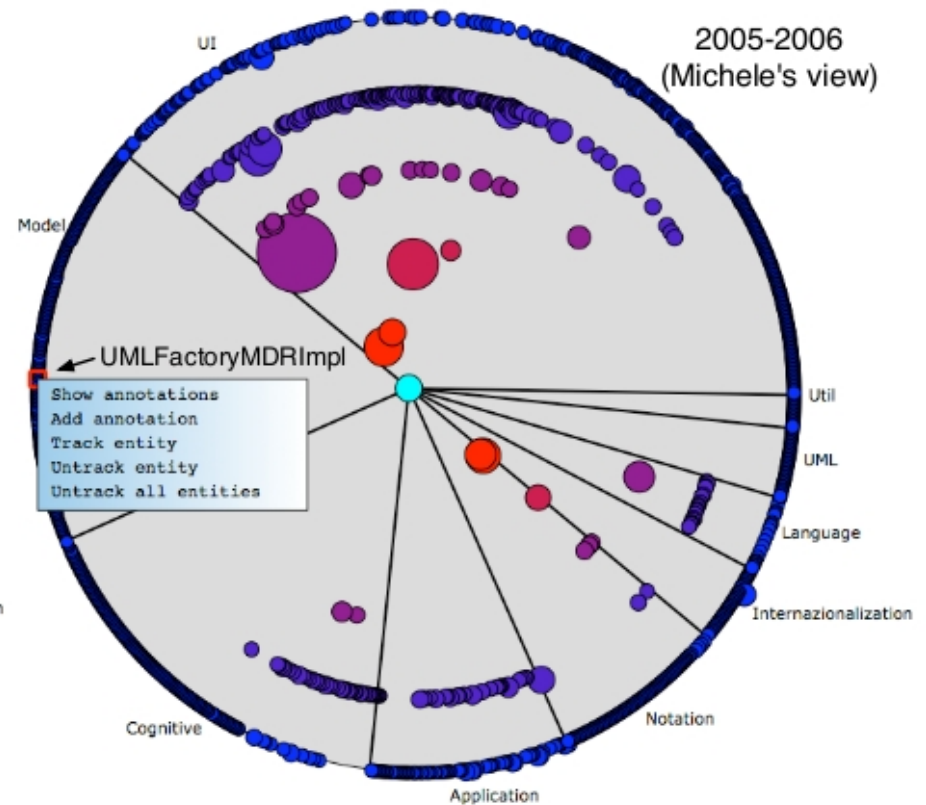
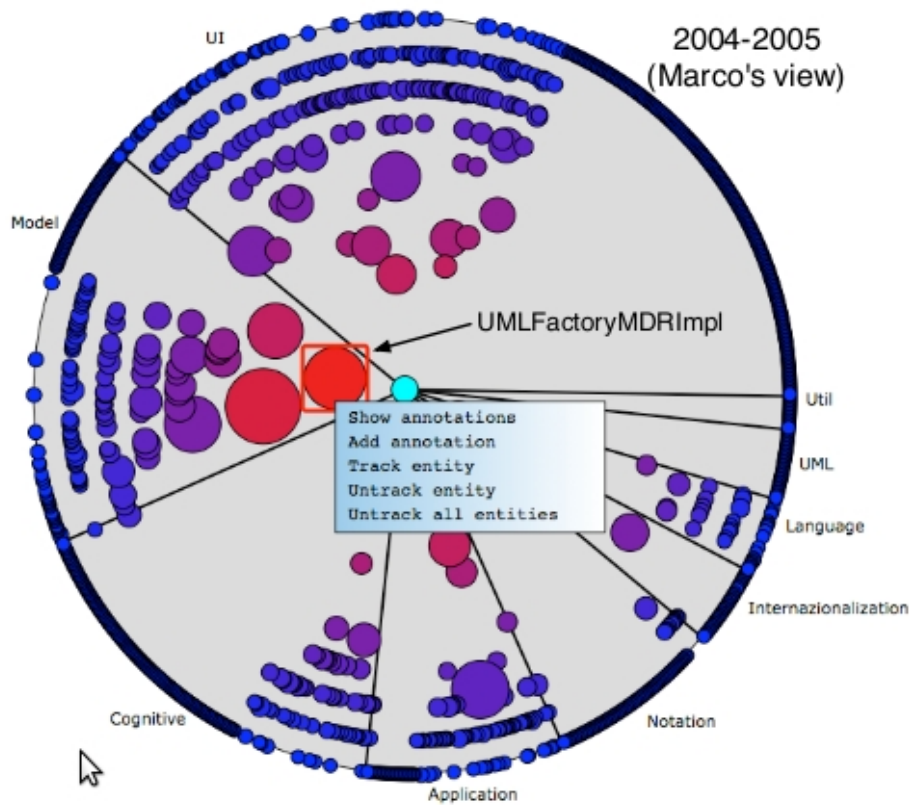
---

## Evolution Radar

- Representa as dependências de um módulo em um período específico de tempo.
  - Proximidade com o centro indica maior acoplamento com o módulo representado.
  - Tamanho indica número de mudanças no intervalo sendo analisado.
- Marco está olhando para o período 2004-2005 e Michele para 2005-2006.

# Churrasco in action

## Evolution Radar



# Churrasco in action

---

## Evolution Radar

1. Marco seleciona a classe *UMLFactoryMDRImpl*, que é a maior e a mais próxima do centro.
2. Marco adiciona uma anotação dizendo que essa classe é perigosa, dado que ela possui um alto acoplamento com um módulo diferente (*Diagram*) do qual ela faz parte (*Model*).
3. Michele vê as anotações de Marco e percebe que na sua visualização a classe *UMLFactoryMDRImpl* não está acoplada com o módulo *Diagram*. Ele então adiciona uma anotação dizendo que a classe não é perigosa, pois o acoplamento diminui ao longo do tempo.



# Churrasco in action

---

Este cenário mostra que:

- O conhecimento adquirido através da análise da evolução de um sistema pode ser construído de forma incremental.
- Diferentes usuários em diferentes locais podem colaborar.
- Várias técnicas de visualização podem ser combinadas para melhorar a análise.

Porém:

- Só houveram dois participantes,
- que eram os desenvolvedores da ferramenta,
- e que possuíam conhecimento prévio sobre o sistema analisado.

# Experimentos

---

## Objetivos:

- Avaliar se o Churrasco provê suporte para colaboração durante análise da evolução de um sistema.
- Testar a usabilidade da ferramenta.
- Testar a escalabilidade da ferramenta em termos do número de participantes.

# Experimento 1

---

## Tarefa:

Utilizar as visualizações *System Complexity* e *Correlation View* para:

1. Descobrir classes onde é necessário algum esforço de reengenharia (e explicar porquê).
2. Descobrir classes que impactam muitas partes do sistema caso modificadas, e explicar porquê.

## Sistema:

JMol, um visualizador 3D de estruturas químicas.

- ~ 900 classes

# Experimento 1

---

## Anotações:

Class	NOA	NOM	Annotation
JmolSimpleViewer	0	8	"This is a strange hierarchy. There is only one subclass per superclass (all with many method and few attributes)."
JMolViewer	0	135	"Strange: 134 abstract methods, only 1 concrete, only 1 subclass."
Viewer	54	465	"This class seems to be the "thing" in the system, at least in terms of functionality", "Strong dependency with Eval.", "High fan out (25) and many LOC (>1k).", "High number of access to foreign data."
Eval	34	198	"This class should probably be broken down.", "Very strong dependency with Viewer."
JMol	60	25	"This class has the largest fan out (78). Probably part of the core of the system.", "High coupling, low cohesion", "13 protected methods and no child!"
PngEncoder	23	26	"17 protected attributes, completely useless since there's no child!"
BondIterator	5	5	"There are ca. 6 classes with Iterator logic. The implementation is strange. I would expect them to be in some hierarchy."
Graphics3D	101	166	"This can be probably broken down. It's an implementation of a 3d engine."

# Experimento 1

---

## Respostas:

Statement	Strongly disagree	Disagree	Neither	Agree	Strongly Agree
Churrasco is easy to use			1	3	2
System Complexity view is useful				2	5
Correlation view is useful			1	1	5
Churrasco is a good means to collaborate					7
Collaboration is important in reverse engineering			1	5	1

# Resultados

---

- Participantes acharam as visualizações úteis para a realização das tarefas.
- Churrasco escalou bem com 8 participantes acessando o mesmo modelo através do portal web, sem problemas de performance.

# Experimento 2

---

## Cenário:

- Alunos de graduação desenvolveram aplicativo web em smalltalk.
- Última semana de desenvolvimento foi somente para refatoração.
- Grupo de 4 alunos:
  - 1 líder, identifica potenciais classes para refatoração.
  - 3 alunos responsáveis por ver em detalhes as indicações do líder.

## Tarefas:

- Usar as visualizações *System Complexity* e *Correlation View* para identificar quais partes do sistema deveriam ser refatoradas.

# Experimento 2

---

<b>Class ElementModel</b>
What is the difference between Element Model and Element? Are both hierarchies replicated?
One is the model that manages the functionalities of the element, the other one manages the displaying of the element (it is a proxy pattern).
One is for the layout behavior while the other is for the widget behavior.

<b>Class WBLBorderLayoutModel</b>
This layout seems to have more behavior than the others, even though it has the same number of attributes. Maybe it is doing too much and it should be a composite layout?
It has a lot of complex operations which being detached can raise the complexity much more. As you say it has functionalities that can be put in more than 1 class.
The layout is complex. Dividing it into several classes will require too much time and effort.



# Experimento 2

---

Statement	Strongly disagree	Disagree	Neither	Agree	Strongly Agree
Churrasco is a good means to collaborate				1	3
Collaboration helped me in understanding the system				3	1
The proposed methodology (leaders) helps in structuring the collaborative effort				3	1
Reading other users' annotations eases the given tasks				2	2
Quantify (1-10) the added value of the collaborative support provided by the tool	1-2	3-4	5-6	7-8	9-10
				2	2

# Resultados

---

- Colaboração ajudou os participantes a entenderem melhor o sistema.
- O uso de anotações tornou mais fácil a tarefa de encontrar potenciais candidatos à refatoração.

# Conclusões

---

## Principais *features* do Churrasco:

- Possui um *meta-model* flexível.
- Como todo o processo é feito através do portal web, é acessível através de qualquer navegador, sem necessidade de instalação ou configuração.
- Resultados são armazenados em um banco de dados central, gerando resultados incrementais das análises feitas.
- Provê suporte para análises colaborativas, como foi mostrado com os experimentos.
- Aplicações web introduzem novas formas de interação com sistemas, abrindo assim novas portas para a pesquisa.

# Trabalhos futuros

---

- Extender a ferramenta.
  - Extrair informações de listas de emails
- Fazer mais experimentos.
  - Experimento quantitativo sobre a utilidade do Churrasco.
  - Experimento qualitativo (entrevistas) sobre a utilidade e usabilidade do Churrasco.