



Cytoscape

Phân tích và hình ảnh hóa mạng

Hướng dẫn dành cho nhà phát triển Cytoscape 3.3

John "Scooter" Morris, Tiến sĩ, UCSF



phác thảo



- Giới thiệu/Cài đặt•
“Varia/onsonatheme-Helloworld”

- Bước 1: Cytoscape và OSGi
- Bước 2: Mô hình mạng Cytoscape
- Bước 3: Mô hình bảng Cytoscape
- Bước 4: Mô hình trực quan Cytoscape
- Bước 5: Cơ chế giao diện người dùng Cytoscape
- Bước 6: Sự kiện
- Bước 7: Lệnh •

APItour

- Thảo luận: BestPractices



Giới thiệu



- John “Scooter” Morris
 - Hiện nay
 - Trợ lý Giáo sư phụ trợ, Dược phẩm/calChemistry. Giám đốc, NIHResourceforBiocomputing, Visualiza/on, và Thông tin/cs (RBVI)@UCSF
 - Kỹ sư lưu động, NIHNationalResourceforNetworkBiology (NRNB)
 - 1985-2004
 - DisinguishedSystemsKiến trúc sư: Genentech, Inc.
 - Cytoscape core team since 2006
 - Tác giả của một số plugin Cytoscape • SFLDLoader, structureViz, clusterMaker, chemViz, metanodePlugin, groupTool, commandTool, bioCycPlugin



Giới thiệu



- Bạn hy vọng sẽ đạt được điều gì hôm nay?



Những lưu ý



- Tôi cho rằng các bạn đều có năng lực Java lập trình viên
- Không thể bao quát mọi thứ
 - Bạn sẽ bị choáng ngợp ngay cả khi chúng ta đã thử
- Chúng tôi sẽ bao gồm Cytoscape 3.3 Bundled
 - Ứng dụng API
 - Simpleapps khác biệt ở nhiều điểm
- Chúng ta sẽ xây dựng mọi thứ trên con đường này
 - Dừng lại/trở lại



Thiết lập - Shell



- Tải xuống các tệp mẫu từ:

<http://www.cgl.ucsf.edu/home/scooter/Cytoscape3DevTut/setup.zip>

- Tháo dỡ



Thiết lập - Eclipse



- Cài đặt Eclipse
 - Cũng muôn:
 - m2e
- Tải xuống tệp mẫu
 - Giải nén
 - Trong

Eclipse:

- File à Import à Maven à Existing Maven Projects
- Điều hướng đến thư mục HelloWorld từ tải về ở trên



Gỡ lỗi với Eclipse



RemoteExecutionforDebugging • Chạy

Cytoscape với debug là lệnh đầu tiên và lineargument

Trong Windows

```
cytoscape.bat gỡ lỗi
```

TrongLinux/Mac

```
./cytoscape.sh debug Đảm bảo bạn
```

có thể thấy thông báo sau trong terminal: Nghe để vận chuyển

```
dt_socketat tại địa chỉ: 12345
```

LƯU Ý: phải nằm trong thư mục Cytoscape



Gỡ lỗi với Eclipse



Chạy DebuggerFromEclipse •

Nhấp vào Run và chọn DebugConfigura/ons... • Chọn

RemoteJavaApplica/ons và tạo cấu hình mới/on • Chọn/Nhập thông tin sau và nhấn Apply:

- Tên:Cytoscape3
- Dự án: myproject(yourprojectnamemaybedifferent;so"browse"tofindthe
oneyouareabouttodebug (bạn sắp sửa gỡ lỗi)
- Connec/onType:Standard(SocketAlach)
- Máy chủ: localhost
- Cổng: 12345 (hoặc cổng được chỉ định trong tệp cytoscape.sh)

• Nhấn Debug. Thao tác này sẽ khởi động Cytoscape3. Sau 20-30 giây, bạn sẽ thấy CytoscapeDesktop. •

Đặt điểm dừng trong Eclipse.

Chuyển đổi Perspec/vetoDebugmode: Window Perspec/ve Debug • Tải bundle vào container OSGI của bạn và chạy



Bước 1: Cytoscape & OSGi



- Mục tiêu thiết kế Cytoscape3

- Khả năng mở rộng
- Hiệu suất
- Tính ổn định
 - Tính ứng dụng/ ổn định •

Tính ổn định của

API • Tính mô-đun

- Được thực thi bởi OSGi



- **Wikipedia:**

OSGiframework là một hệ thống mô-đun và nền tảng dịch vụ cho Java ngôn ngữ lập trình thực hiện một mô hình thành phần hoàn chỉnh và động, một cái gì đó không tồn tại độc lậpJava/VM môi trường. Các ứng dụng hoặc thành phần (được sắp xếp theo dạng bó) để triển khai) có thể cài đặt, khởi động, dừng, cập nhật và gỡ cài đặt từ xa mà không cần khởi động lại; quản lý các gói Java/ Các lớp được chỉ định rất chi tiết. Áp dụng quản lý vòng đời (bắt đầu, dừng, cài đặt, v.v.) được thực hiện thông qua các API cho phép tải xuống từ xa các chính sách quản lý. Sổ đăng ký dịch vụ cho phép các gói phát hiện sự cố khi có dịch vụ mới hoặc dịch vụ bị xóa và điều chỉnh cho phù hợp.



OSGi



- Hướng đến dịch vụ
 - Một “gói” là đơn vị truy cập
 - Các gói có thể được bắt đầu và dừng lại một cách độc lập
 - Gói dịch vụ triển khai
 - Có thể đăng ký và không đăng ký
 - Nói chung, truy cập giữa các gói thông qua một dịch vụ
 - Thực thi tách biệt API và triển khai
 - Quy tắc là bạn có thể phụ thuộc vào APIbundles, nhưng notimplementa/onbundles



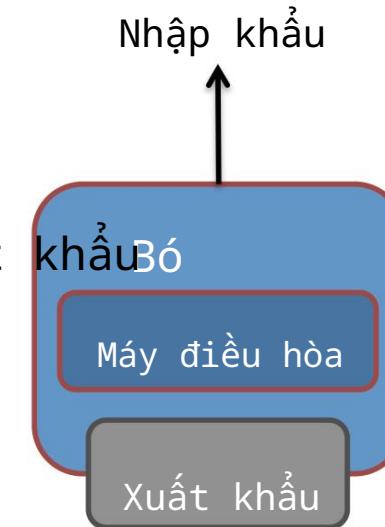
Giải phẫu của một bó



- JAR với extrametadata
- Nhập khẩu

The Java packages used by the bundle

- Các gói Java trong gói mà các gói khác được phép sử dụng (thường chỉ là API)
- Máy lạnh
 - Được kích hoạt khi bundle bắt đầu/dừng

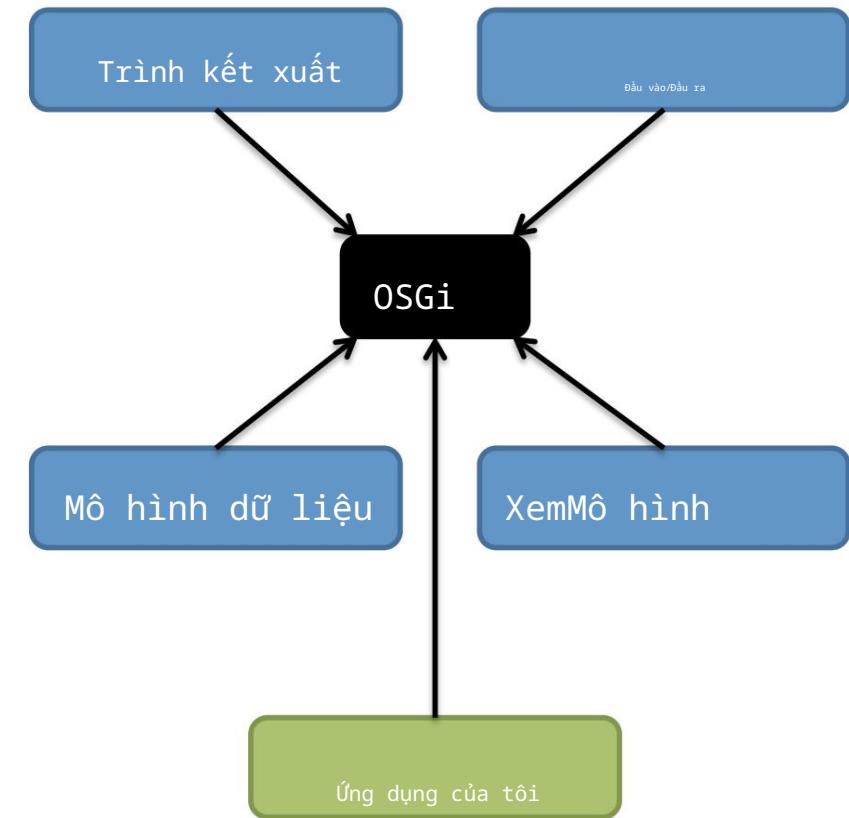




Tế bào học 3



- Nhân vi mô hướng dịch vụ
 - Hệ điều hành Gicore
 - Tải động/ dỡ bỏ các mô-đun, hay còn gọi là bó
 - Mỗi hệ thống con trong Cy3 có OSGi riêng biệt bundle(s).
- Ứng dụng cũng có thể được đóng gói thành bundle





Ví dụ: HelloWorld



- pom.xml
 - Maven project descriptor- Maven id/en/fier
 - Nhóm id• Ar/fact id
 - Phiên bản
 - OSGi id/en/fier
 - Tên biểu tượng bó
 - Mô tả nhập khẩu/ xuất khẩu
- Kích hoạt.java
 - Bundle activator

Maven project descriptor- Maven id/en/fier

- Nhóm id• Ar/fact id

Ar/fact id

- Phiên bản

- OSGi id/en/fier

- Tên biểu tượng bó

- Mô tả nhập khẩu/ xuất khẩu



Bố cục dự án Maven



- pom.xml
 - Mô tả dự án
- src/main/java
 - Mã gói
- src/test/java
 - src/kiểm tra/java
 - Mã kiểm tra
 - Không bao gồm trong bundleJAR
- src/main/nguồn tài nguyên
 - Các tệp không phải mã cần được bao gồm trong bundleJAR



Gói cốt lõi



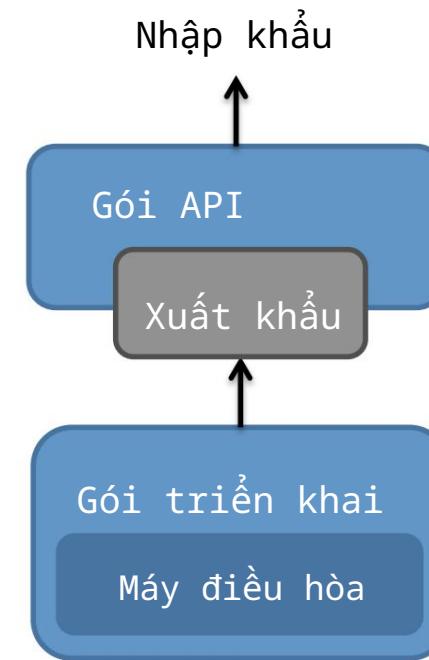
- ứng dụng/ trên • lệnh-thực thi
- nhiệm vụ cốt lõi
- đồ họa tùy chỉnh
- nguồn dữ liệu
- phương trình/các phép tính
- sự kiện
- nhóm
- io
- cách trình bày
- người mẫu
- hiện tại/trên tài sản.
- dịch vụ
- phiên họp
- đú-u/l
- xem mô hình
- bản đồ hình ảnh
- vizmap-gui
- dịch vụ web
- công việc



Gói cốt lõi



- Thường có phần chèn:
 - API (tùy chọn/tùy chỉnh)
 - Không có điện/điện áp
 - Triển khai/trên
 - At least one per API bundle
 - Không xuất khẩu
- API riêng biệt để chúng ta có thể tiếp tục thực hiện một/một mô-đun
 - Desktop application
 - Console application, for scripts

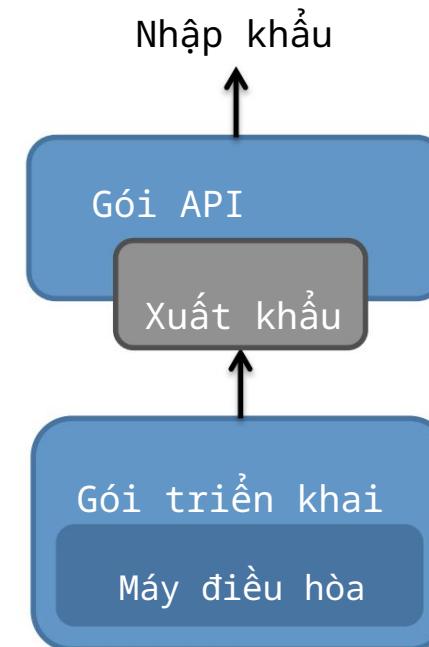




Gói cốt lõi



- Không có gì nên nhập khẩu thực hiện/onbundles
- Trừ khi nó dành cho đơn vị kiểm tra/ng





Gói cốt lõi



- Các gói tác vụ
 - api làm việc
 - work-swing-api
 - công việc-implement
 - công việc-swing-implement
 - work-headless-impl
- VizMapperbundles - vizmap-api
 - vizmap-gui-api
 - vizmap-gui-core-impl
 - vizmap-gui-impl
 - vizmap-impl



Dịch vụ OSGi

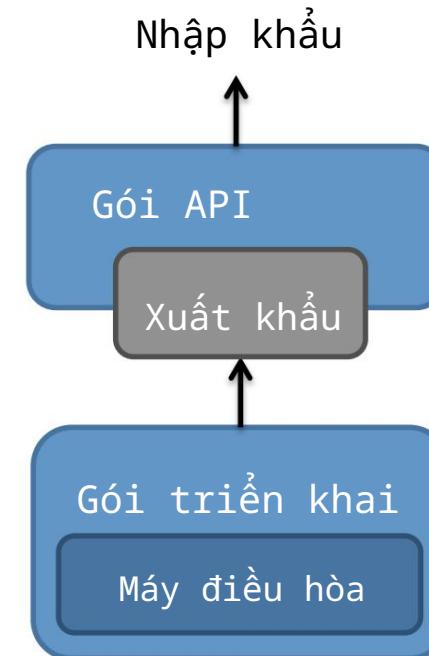


- **Dịch vụ**

- Một thể hiện của Java giao diện
- API và thực hiện các gói keo dán
- Thường được đăng ký bởi GóiAc/vator

- **InCy3:**

- Định nghĩa yanAPIbundle
- Đăng ký bởi animplementa/onbundle





Dịch vụ OSGi



- Giao diện
 - giao diện MyService
{ . }
- Thực hiện/trên
 - lớp MyServiceImpl
thực hiện MyService
{ . }
- Đúng/đúng
 - Cặp khóa-giá trị tùy ý
 - ("tiêu đề", "Tôi
Dịch vụ")
("Menu ưa thích",
"Ứng dụng")



API Cytoscape



- Có sẵn dưới dạng dịch vụ OSGi
- Hai loại chính:
 - API:ApplicationProgrammingInterface • Chỉ cần lấy và sử dụng:
`Dịch vụ MyService = getService(bối cảnh, MyService.class);`
 - SPI:Giao diện nhà cung cấp dịch vụ
 - Triển khai/mở rộng và đăng ký:
`registerService(context, new MyServiceImpl(),
MyService.class, thuộc tính);`



Dịch vụ OSGi



- Các loại dịch vụ phổ biến nhất:
 - Nhà máy
 - Tạo trường hợp mới
 - Người quản lý
 - Theo dõi, cung cấp quyền truy cập hoặc vận hành thu thập/tài sản của các đối tượng
 - U/li/es
 - Thu thập/onsofu/lityfunc/ons



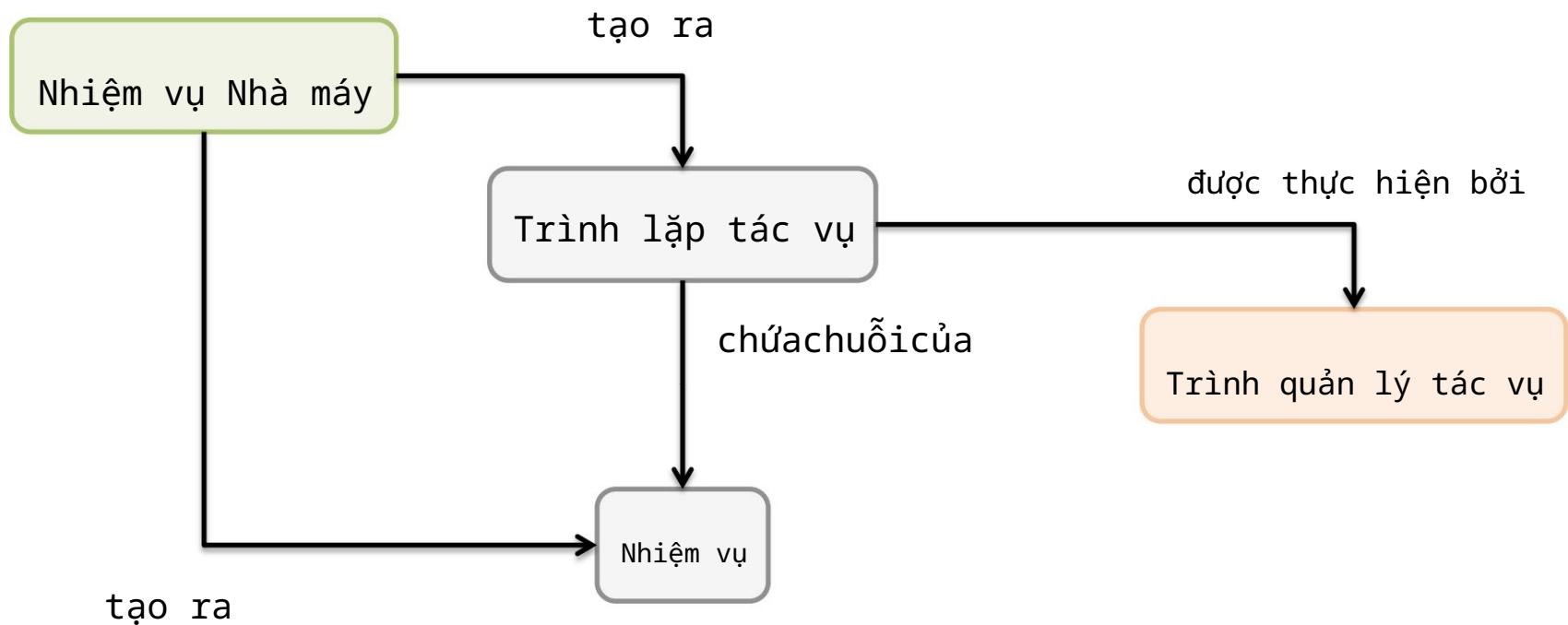
Nhiệm vụ Cytoscape



- Nhiệm vụ Nhà máy
 - Đơn vị công việc chính trong Cytoscape là “Nhiệm vụ”
 - Nhiệm vụ được tạo bởi TaskFactories



Nhiệm vụ





Nhiệm vụ Cytoscape



- **Nhiệm vụ Nhà máy**

- Đơn vị công việc chính trong Cytoscape là “Nhiệm vụ”
- Nhiệm vụ được tạo bởi TaskFactories
- TaskFactories are OSGi services
- Có thể đăng ký tại CyAc/vator của bạn:

Nhà máy TaskFactory =

```
registerService(bc, myFactory, TaskFactory.class, thuộc tính);
```

where myFactory is the taskfactory you want to register

- proper/es provide meta-data about the factory
 - Java Proper/es



Nhiệm vụ Cytoscape



- **CytoscapeTaskFactoryProper/es**
 - Proper/eshavespecialmeaninginCytoscape- Được định nghĩa trong [org.cytoscape.work.ServiceProper/es](#)

 - Keyproper/es • TITLE-
 - Nếu sử dụng như một menu, đây là menu/tle •
 - PREFERRED_MENU-Nơi này sẽ được thêm vào •
 - ENABLE_FOR-Khi menu này được bật •
 - IN_TOOL_BAR-có trên thanh công cụ không? • IN_MENU_BAR-có trên các menu cấp cao nhất không? • MENU_GRAVITY-Trọng lượng riêng của mục này.



Nhiệm vụ Cytoscape



- Ví dụ:

```
nhập org.cytoscape.work.AbstractTaskFactory; nhập  
org.cytoscape.work.TaskIterator;
```

```
lớp MyTaskFactory mở rộng AbstractTaskFactory {  
    công khai MyTaskFactory() {  
        siêu();  
    }  
  
    công khai TaskIterator createTaskIterator() {  
        trả về null; // Điền vào  
    }  
  
    boolean công khai isReady() { trả về true; }  
}
```



Nhiệm vụ Cytoscape



- Ví dụ (inCyAc/vator):

```
MyTaskFactory myFactory = new MyTaskFactory(); Properties props= new  
Properties(); // Lưu ý ký hiệu "." cho các menu xếp tầng  
props.setProperty(PREFERRED_MENU, "Apps.cddApp"); props.setProperty(TITLE,  
"Load CDD Domains for Node"); // Không phải tắt cả các nhà máy tác vụ đều là lệnh  
  
props.setProperty(COMMAND, "loadCDDDomains4node");  
props.setProperty(COMMAND_NAMESPACE, "cddApp");  
props.setProperty(IN_MENU_BAR, "true"); // Thường  
có nghĩa là mục menu thứ hai  
props.setProperty(MENU_GRAVITY, "2.0");  
registerService(bc, loadCDDDomainNodeView,  
                NodeViewTaskFactory.class, nodeViewProps);
```



Bước 1: Dự án



- Add a new Cytoscape Plugin menu
 - Menu title: Xin chào thế giới!
 - Hiện tại, không cần phải làm nhiệm vụ



Các nhà máy quan trọng



- CyNetworkFactory - mô hình-api
- CyTableFactory
 - mô hình-api
- CyGroupFactory - nhóm-api
- CyNetworkViewFactory - viewmodel-api
- VisualMappingFunctionFactory - vizmap-api
- VisualStyleFactory
 - vizmap-api



Các nhà quản lý quan trọng



- CyApplicationManager
 - Ứng dụng/trên api
 - Rất nhiều thông tin “nhà nước”
- Quản lý mạng CyNetwork
 - mô hình-api
- Quản lý CyTable
 - mô hình-api
- Quản lý bảng mạng CyNetwork
 - mô hình-api
 - Quản lý hiệp hội/trên
giữa các bảng và các đối tượng mạng
- Trình quản lý CyNetworkView
 - viewmodel-api
- Quản lý nhóm CyGroup
 - nhóm-api
- Quản lý bản đồ trực quan
 - vizmap-api



Nhà máy nhiệm vụ đặc biệt



- API nhiệm vụ cốt lõi
 - NetworkViewTaskFactory
 - Mạngbackgroundcontextmenu
 - NodeViewTaskFactory
 - Nodecontextmenu
 - EdgeViewTaskFactory
 - Edgecontextmenu



Nhiệm vụ



- Môi trường chạy/môi trường thực tế có chủ đích
 - Desktop application
 - Không có đầu, via script/ng
 - Lập trình/thực tế, qua ứng dụng
- Không phù hợp để đảm nhận nhiệm vụ sẽ là một con đường
 - Chỉ sử dụng Swing trong các nhiệm vụ nếu cần thiết



Bước 2: Mô hình Cytoscape



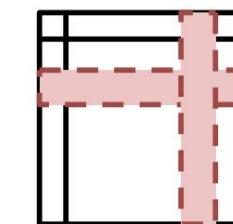
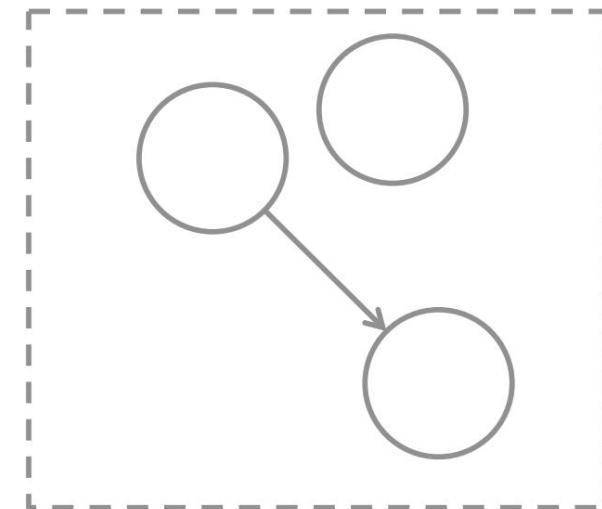
- Hai khái niệm cơ bản trong mô hình dữ liệu:
 - Mạng lưới CyNetwork
 - Bàn Cy



Các khái niệm: Mô hình dữ liệu



- Mạng lưới Cy
 - Mul/đồ thị
 - Có hướng hoặc không có hướng
 - CyNode
 - CyEdge
- Bảng Cy
 - CyColumn
 - Khóa chính
 - CyRow
- Mã số định danh
 - Phiên-duy nhất/fier

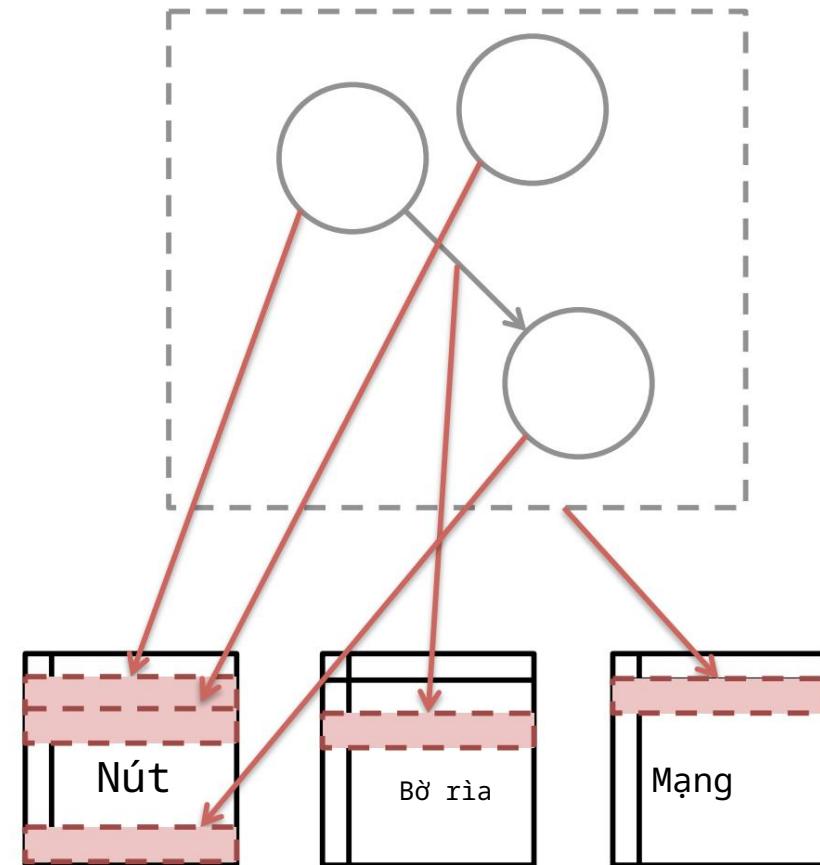




Các khái niệm: Mô hình dữ liệu



- Mạng lưới Cy
 - Mô hình/đồ thị
 - Có hướng hoặc không có hướng
 - CyNode
 - CyEdge
- Bảng Cy
 - CyColumn
 - Khóa chính
 - CyRow
 - Mã số định danh
 - Phiên-đuy nhất/fier

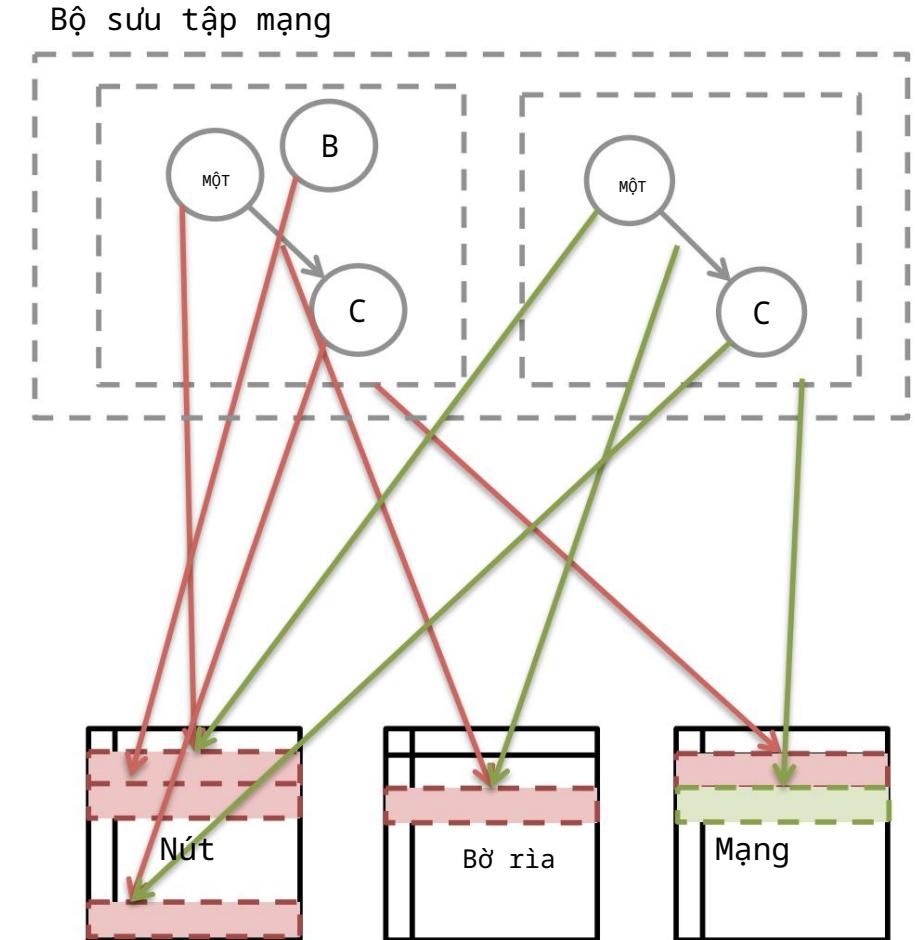




Các khái niệm: Mô hình dữ liệu



- Mạng lưới Cy
 - Can have multiple networks in a "collection"
 - Essentially a single-level hierarchy - `CyRootNetwork`
 - Top `CyNetwork` in the collection • Contains all nodes and edges
 - [org.cytoscape.model.subnetwork](#)
 - Mạng lưới CySub
 - Subgraph/một phần của `CyRootNetwork`
 - Có thể nhiều/nhiều mạng con
 - All nodes and edges in the same SUIDs as in `CyRootNetwork`
 - All `CyNetworks` are not `CyRootNetworks` are `CySubNetworks`





Các khái niệm: Mô hình dữ liệu

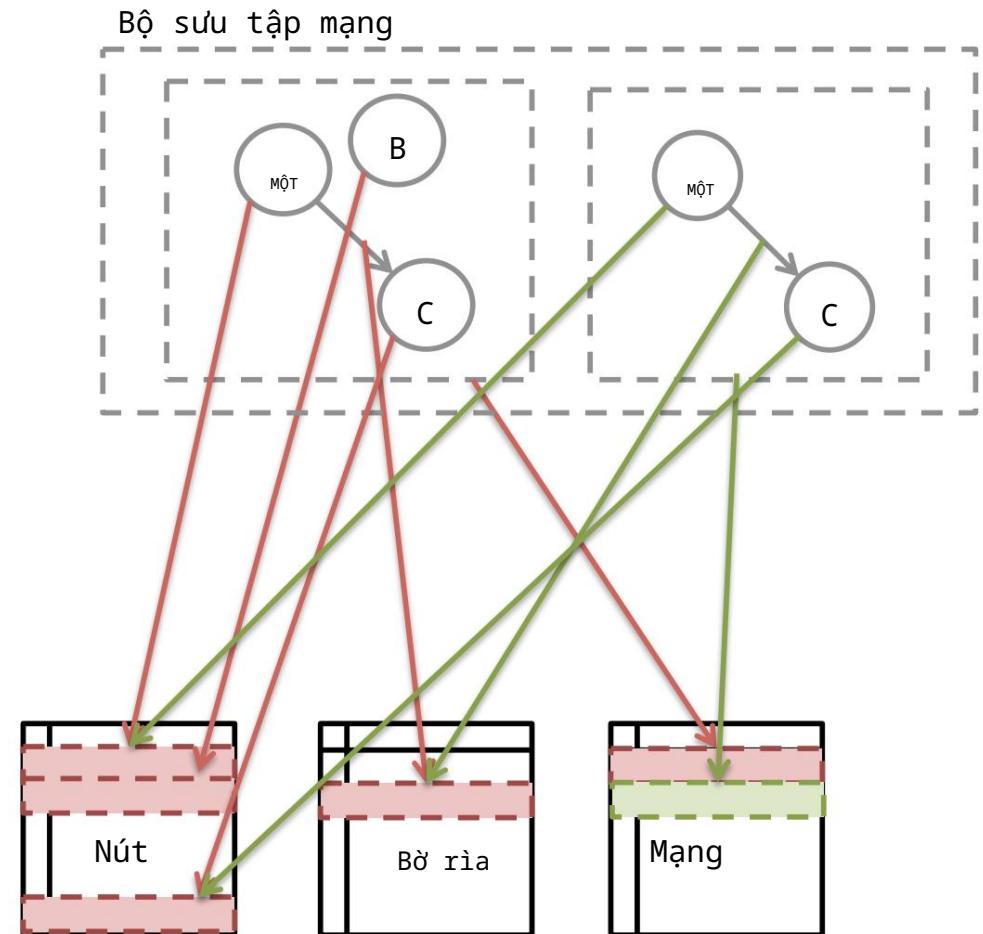


- **Mạng lưới Cy**

- Can have multiple networks in a "collection"
- CyNode and CyEdge attribute can be shared
- Membership properties cannot be shared

- **Bảng Cy**

- Thực tế có ba "kiểu" bảng:
 - Thuộc tính chia sẻ
 - Chia sẻ trong mạng lưới thu thập/trên
 - Thuộc tính cục bộ
 - Không chia sẻ
 - Bàn "Mặt tiền"
 - Xem các bảng cục bộ và bảng chia sẻ đã hợp nhất





Các khái niệm: Mô hình dữ liệu



- Mạng lưới Crea/nga

- Sử dụng [CyNetworkFactory](#) để tạo mạng và [CyNetworkManager toaddit](#)

- Commonpalern• Cần lấy chúng trong CyActivator của bạn: CyNetworkFactory

```
networkFactory = getService(bc,
    CyNetworkFactory.class);
CyNetworkManager networkManager =
    getService(bc, CyNetworkManager.class);
```

- Sau đó chuyển chúng đến taskfactory của bạn trong constructor của nó
- Sau đó chỉ cần sử dụng chúng:


```
CyNetwork newNetwork = networkFactory.createNetwork();
networkManager.addNetwork(mạng mới);
```
- Hầu hết các operations (bao gồm cả việc thêm các nút và cạnh) là trên [CyNetwork](#)



Bước 2: Dự án



- Thêm một nhiệm vụ vào TaskFactory từ Bước 1
- Nhiệm vụ nên thêm node vào mạng
 - Tôi khuyên bạn nên mở rộng Nhiệm vụ Trùu tượng
 - Nếu bạn có/tôi, hãy thêm hai nút và một cạnh giữa chúng
- LƯU Ý: Bạn sẽ phải tạo chế độ xem theo cách thủ công• GỢI Ý: bạn sẽ cần phải chỉnh sửa tệp pom.xml của mình để bao gồm headdi/onaldependency



Bước 3: Mô hình bảng



- **Bảng Cy**

- Mô hình bảng chuẩn:
 - Các cột có kiểu cố định:
 - Boolean, Chuỗi, Số nguyên, Dài, Đôi
 - Danh sách<Boolean>, Danh sách<Chuỗi>, Danh sách<Số nguyên>, Danh sách<Dài>, Danh sách<Đôi>
 - Rows are singly indexed by a key
- Các cột có thể là “ảo”
 - Essen/allyfunc/onsasalink từ một bảng vào một bảng khác



Bảng và Mạng



- Crea/ngaCyNetwork tạo ra:
 - Bảng mạng
 - LOCAL_ATTRS và HIDDEN_ATTRS cho từng mạng
 - Bảng nút
 - LOCAL_ATTRS và HIDDEN_ATTRS mỗi mạng
 - DEFAULT_ATTRSforeachnetwork
 - » Tất cả các cột trừ khóa là ảo (Kết hợp LOCAL_ATTRS và CHIA SẺ_ATTRS)
 - SHARED_ATTRSforeachcollection
 - Bàn cạnh
 - LOCAL_ATTRS, HIDDEN_ATTRS và DEFAULT_ATTRS cho từng mạng
 - SHARED_ATTRSforeachcollection



Bảng và Mạng



Người mẫu Sự vật	Bàn	Chìa khóa	Ghi chú
CyNetwork LOCAL_ATTRS		CyNetwork.SUID Standardlocaltable	
CyNetwork HIDDEN_ATTRS		CyNetwork.SUID Không hiển thị người dùng	
CyNode	ĐỊNH_CỤC_ĐỊA_VỊ	CyNode.SUID	Localtable not shared across networks in the same collection
CyNode	CHIA SẺ_ATTRS	CyNode.SUID	Sharedtable. One for each network collection.
CyNode	MẶC ĐỊNH_ATTRSCyNode.SUID		The façade table. Essentially all virtual columns point to LOCAL_ATTRS and SHARED_ATTRS tables
CyNode	ÂN_TÂN_CẬP	CyNode.SUID	Thuộc tính cục bộ không được hiển thị cho người dùng
Tiếng Việt	ĐỊNH_CỤC_ĐỊA_VỊ	CyEdge.SUID	Localtable not shared across networks in the same collection
Tiếng Việt	CHIA SẺ_ATTRS	CyEdge.SUID	Sharedtable. One for each network collection.
Tiếng Việt	MẶC ĐỊNH_ATTRSCyEdge.SUID		The façade table. Essentially all virtual columns point to LOCAL_ATTRS and SHARED_ATTRS tables
Tiếng Việt	ÂN_TÂN_CẬP	CyEdge.SUID	Thuộc tính cục bộ không được hiển thị cho người dùng



Mô hình bảng



- Cột chuẩn
 - CyNetwork.TÊ N
 - Tên của nút, cạnh hoặc mạng
 - CyNetwork.ĐÃ CHỌN
 - Nếu đúng, đối tượng này được chọn
 - CyRootNetwork.SHARED_NAME
 - Trong bảng SHARED_ATTRTables • Shared(root)nameoftheobject



Mô hình bảng



- Cũng có thể tạo bảng của riêng bạn

- Đối tượng mạng bị ràng buộc •

- Key luôn là SUID(Dài) • Nên

- được đăng ký với CyNetworkTableManager

- Có thể dễ dàng kéo từ CyNetwork:

```
CyNetwork.getTable(Class<? extends CyIdentifiable> type, không gian tên String );
```

- Phím tắt để nhận hàng:

```
CyNetwork.getRow(Mục nhập CyIdentifiable , không gian tên String );
```

- Không ràng buộc

- Keyisanyvalidtype•

- Nên được đăng ký với CyTableManager



Truy cập vào bảng



- Các cột Test/ngfor
 - Phải kiểm tra sự tồn tại của cột trước khi truy cập nếu
`(table.getColumn(String columnName) != null)`
- Tạo/cột mới
 - Cột phải được gõ
 - Cột danh sách phải bao gồm loại danh sách • Gezngrows
 - Từ bảng:
`lấyRow(Khóa đối tượng)`
 - Từ một mạng lưới:
`lấyRow(CyIden/fiableentry, Stringnamespace)`



Truy cập vào bảng



- Dữ liệu Gezng

- Tất cả dữ liệu được truy cập thông qua các hàng:

```
<T> T CyRow.get(String columnName, Class<? mở rộng T> loại)
```

```
Số nguyên I = row.get("clusterNumber", Integer.class);
```

- Trong đó “type” là kiểu cột. Nó là lỗi nếu kiểu này sai rồi

- Dữ liệu Sezng

```
CyRow.set(Chuỗi columnName, giá trị T)
```

- Where T is the column type

- Lưu ý: thêm các nút và cạnh automa/cally thêm các hàng tương ứng



Bước 3: Dự án



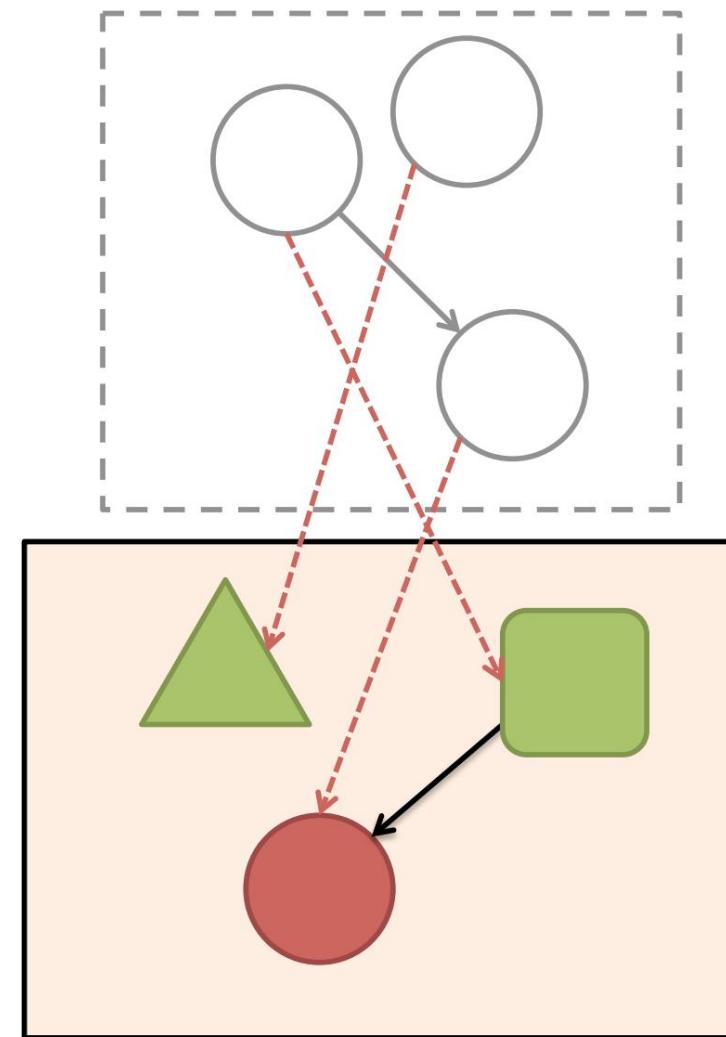
- Trong mạng bạn đã tạo trước đó:
 - Đổi tên của node hoặc node
 - Tạo hai cột nút mới:
 - HelloàListOfStrings
 - WorldàDouble
 - Thêm dữ liệu vào các cột mới



Bước 4: Xem mô hình



- Xem Mạng CyNetwork
 - Xem<CyNode>
 - Xem<CyEdge>
- Thuộc tính trực quan
 - Ví dụ:
 - VỊ TRÍ _NÚT_X
 - CHIỀU RỘNG CẠNH
 - CHIỀU CAO MẠNG





Xem mô hình



- **>Create network view**

`CyNetworkViewFactory.createNetworkView(CyNetwork network);` -

Cần lấy CyNetworkViewFactory trong CyActivator của bạn

- **Will create views for all nodes and edges**

- **Get network view**

`CyNetworkViewManager.getNetworkViews(CyNetwork network);` - Cần

lấy CyNetworkViewManager trong CyActivator của bạn

- Lưu ý bạn có thể thu thập/xem lại-if there can be multiple views per network

- **Get node and edge views**

Xem<CyEdge>edgeView= `CyNetworkView.getNodeView(CyNode node);`

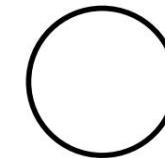
Xem<CyNode>nodeView= `CyNetworkView.getEdgeView(CyEdge edge);`



Thuộc tính trực quan



- Từ điển trực quan
 - Phân cấp thuộc tính trực quan
 - Con cái thừa hưởng giá trị riêng/giá trị riêng từ cha mẹ
- Nút
 - SƠN_NODE
 - NODE_BORDER_SƠN
 - NODE_FILL_COLOR

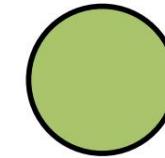




Thuộc tính trực quan



- Từ điển trực quan
 - Phân cấp thuộc tính trực quan
 - Con cái thừa hưởng giá trị riêng/giá trị riêng từ cha mẹ
- Nút
 - SƠN_NODE
 - NODE_BORDER_SƠN
 - NODE_FILL_COLOR





Thuộc tính trực quan



- Từ điển trực quan
 - Phân cấp thuộc tính trực quan
 - Con cái thừa hưởng giá trị riêng/giá trị riêng từ cha mẹ
- Nút
 - SƠN_NODE
 - NODE_BORDER_SƠN
 - NODE_FILL_COLOR

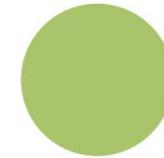




Thuộc tính trực quan



- Từ điển trực quan
 - Phân cấp thuộc tính trực quan
 - Con cái thừa hưởng giá trị riêng/giá trị riêng từ cha mẹ
- Nút
 - SƠN_NODE
 - NODE_BORDER_SƠN
 - NODE_FILL_COLOR





Thiết lập Thuộc tính trực quan



- Visualproper/escanbedirectlysetin Xem giao diện:

```
view.setLockedValue(VisualProperty<? extends T> vp, giá trị v)
```

.nơi View is a node view (View<CyNode>) hoặc can edge view (View<CyEdg

- Thường lấy VisualProperty từ BasicVisualLexicon
- Các loại là quan trọng. Cần phải đảm bảo rằng Visan là loại phù hợp cho VisualProperty
- Ví dụ:

```
nodeView.setLockedValue(BasicVisualLexicon.NODE_FILL_COLOR,  
Màu sắc.XANH DƯƠNG);
```



Phong cách trực quan



- [VisualMappingFunc/on](#)
 - Bản đồ giữa một giá trị CyColumn và giá trị VisualProperty - Ví dụ: cột "name" được ánh xạ tới NODE_LABEL
- [Kiểu dáng trực quan](#)
 - Thu thập/thu thập VisualMappingFunc/ons
- Ba loại bản đồ:
 - Truyền qua
 - Thường được sử dụng cho nhãn
 - Rời rạc
 - Dữ liệu danh mục
 - Con/nuous
 - Ánh xạ phạm vi đến phạm vi Gradient màu nút
 - Kích thước nút



Xây dựng bản đồ trực quan



- Quản lý bản đồ trực quan
 - Quản lý tất cả các kiểu dáng trực quan
 - Get it in your CyAc/vatora service
- Gezing the visual style for a network:

```
Kiểu VisualStyle =  
    vmm.getVisualStyle(CyNetworkView networkView);  
    ..nơi vmm là the VisualMappingManager
```

- Tạo VisualStyle (thường là tạo bản sao)
 - Sử dụng VisualStyleFactory.createVisualStyle(style);-
Get it in your CyAc/vatora service



Xây dựng bản đồ trực quan



- Nhận VisualMappingFunc/onFactory mong muốn trong `yourCyAc/vator`:

```
VisualMappingFunctionFactory vmfFactoryC =  
    getService(bc, VisualMappingFunctionFactory.class,  
              "(mapping.type=continuous)");  
  
VisualMappingFunctionFactory vmfFactoryD =  
    getService(bc, VisualMappingFunctionFactory.class,  
              "(mapping.type=discrete)");  
  
VisualMappingFunctionFactory vmfFactoryP =  
    lấyDịch vụ(bc, VisualMappingFunctionFactory. lớp,  
              "(mapping.type=passthrough)");
```

- Lưu ý rằng đây là một điều khác biệt. Chúng tôi đang sử dụng lại đối số bộ lọc để lấy Dịch vụ



Xây dựng bản đồ trực quan



- Truyền qua:

```
// Sử dụng ánh xạ truyền qua String
ctrAttrName1 = "SUID"; PassthroughMapping
pMapping = (PassthroughMapping)
    vmfFactoryP.createVisualMappingFunction(ctrAttrName1, String.class,
                                                BasicVisualLexicon.NODE_LABEL);

vs.addVisualMappingFunction(pMapping);

// Thêm kiểu mới vào VisualMappingManager
vmm.addVisualStyle(so với);

// Áp dụng kiểu trực quan cho NetworkView
so với.áp dụng(myNetworkView);
myNetworkView.updateView();
```



Xây dựng bản đồ trực quan



- Liên quan/liên quan:

```
// Đặt bản đồ màu của nút thành thuộc tính "Degree"
Ánh xạ ContinuousMapping = (ContinuousMapping)
    vmfFactoryC.createVisualMappingFunction("Độ",
                                                Integer.class,
                                                Từ điển hình ảnh cơ bản.NODE_FILL_COLOR);

// Xác định các điểm
Giá trị kép val1 = 2d;
BoundaryRangeValues<Paint> brv1 = new
    BoundaryRangeValues<Paint>(Màu.ĐỎ, Màu.XANH LÁ, Màu.XANH LÁ);
Giá trị kép val2 = 12d;
BoundaryRangeValues<Paint> brv2 = new
    BoundaryRangeValues<Paint>(Color.YELLOW, Color.YELLOW, Color.BLACK); // Đặt các điểm mapping.addPoint(val1,
brv1);
mapping.addPoint(val2, brv2);

// thêm ánh xạ vào phong cách trực quan
vs.addVisualMappingFunction(ánh xạ);
```



Quyền ưu tiên thuộc tính trực quan



- How Visual Property is determined for a View:
 1. Locked value from View (aka bypass)
 2. Mapped value from Visual Style
 3. Giá trị mặc định cho Visual Style
 4. Giá trị mặc định cho Visual Property



Bước 4: Dự án



- Sửa đổi ứng dụng/on của bạn để tạo một chế độ xem ini/alview• Thay đổi hình dạng của hai nút của bạn• Tạo kiểu trực quan:
 - Sử dụng giá trị đầu tiên trong cột Hello để làm nhãn nút
 - Sẽ cần phải sử dụng VisualBypass, không phải passthrough người lập bản đồ
 - Sử dụng giá trị trong cột World để thiết lập màu sử dụng acon/nuousmapper



Bước 5: Giao diện người dùng



- Nhiệm vụ phải hoạt động ở cả hai
 - Chế độ không đầu (ienongui) •
Nhiệm vụ được thực thi như thế nào?
 - Chế độ GUI
 - Tách SwingUI?
- Lệnh: headlessmode
- Có thẻ điều chỉnh: xử lý đối số



Lệnh



- Xuất khẩu TaskFactories to:
 - Hộp thoại lệnh
 - Giao diện REST
- Đơn giản
 - Thêm vào TaskFactory của bạn đúng/các thuộc tính:
 - COMMAND_NAMESPACE
 - Tên chung của ứng dụng của bạn.
 - Tất cả các lệnh của bạn sẽ được nhóm lại theo • LỆNH -
 - Bản thân lệnh
- Lập luận?

LƯU Ý: bạn cần cẩn thận về những gì TaskFactories bạn xuất (không phải guionly)



- Có thể điều chỉnh:

- `areJavaAnnotations`

- `automaticallygenerateGUI-`

- `automaticallyexposecommandarguments` • Ví dụ: import

```
org.cytoscape.work.Tunable;  
nhập org.cytoscape.work.AbstractTask;  
lớp MyClass mở rộng AbstractTask {  
    @Tunable (mô tả="Giá trị số nguyên của tôi")  
    giá trị int công khai;  
}
```



- Các kiểu cơ bản - int, double, float, String, long, boolean
 - Tệp,

URL • Các lớp cho các Tunable phức tạp hơn

- ListSingleSelection, ListMultipleSelection
 - BoundedDouble, BoundedFloat, BoundedInteger,
Giới hạn dài
-
- Ví dụ:
 @Tunable (description="Chọn từ danh sách")
 công khai ListSingleSelection màu=
 new ListSingleSelection("đỏ", "xanh lam", "xanh lục");



- Chỉ có lệnh điều chỉnh
 - Được sử dụng cho select/onofnodes, edges, androws
 - Mạng lưới CyNetwork
 - Lớp học U/lity
 - EdgeList, NodeList, RowList



Có thể điều chỉnh



- Các tham số có thể điều chỉnh thường được sử dụng
 - ngữ cảnh: `limittunabletocertaincontext("gui", "nongui", "both")` - phụ thuộc vào `:dependencybetween tunables`
 - `gravity:controlorderofpanels-`
`groups:grouptunablepanelstogether-`
`listenForChange:listoftunablesthatwillupdate`
có thể điều chỉnh được
 - Công cụ/p



Có thể điều chỉnh



- Phương pháp tiếp cận của GelersandSelers
 - Cũng có thể sử dụng rõ ràng các từ gelers và selers
 - Cho phép kiểm soát các giá trị

```
@Tunable (mô tả="Kiểm tra")  
public int getTest() { giá trị trả về; }  
công khai void setTest(int v) { giá trị = v; }
```
 - Rất hữu ích cho ini/aliza/onvaluesandreas/ngto thay đổi



- Lặt Vặt

- Có thể có các lớp ngữ cảnh với mul/pletunables

- Các giá trị có thể điều chỉnh:

`@ContainsTunables`

ngữ cảnh MyContext công khai;

- Resul/ngUIwillincludecontextTunables

- Cung cấp một/một hộp thoại

- Cung cấpTiêu đề:

`@Cung cấpTiêu đề`

công khai String getTitle() {trả về "MyTitle";}



Tin nhắn trạng thái



- Hai cách để thông báo cho người dùng về trạng thái:
 - org.cytoscape.work.TaskMonitor
 - Đã truyền một đối số để chạy() phương thứcTasks• setTitle() và showMessage() cung cấp các thông báo trạng thái
 - Tin nhắn cũng được ghi lại trong Lịch sử nhiệm vụ của Cytoscape
 - setProgress() cập nhật thanh tiến trình
 - org.cytoscape.application.CyUserLog
 - Cơ sở ghi nhật ký chung cho tin nhắn người dùng
 - Sử dụng(orcuse) Log4J
 - Tin nhắn được đăng nhập vào Lịch sử nhiệm vụ của Cytoscape



Bước 5: Dự án



- Sửa đổi nhiệm vụ của bạn:
 - Nhập hình dạng để tạo các nút của bạn
 - Nhập màu cao và thấp cho phạm vi của bạn
- Xuất lệnh cho nhiệm vụ của bạn



Bước 6: Sự kiện



- Triết lý Cytoscape:
 - Effect lower layers by method invocation
 - Effect upper layers by event handling
- Các sự kiện cháy ở lớp dưới được xử lý bởi lớp trên lớp.

Tìm kiếm các gói ".events": -

[org.cytoscape.model.events](#)

- [org.cytoscape.view.model.events](#)

- [org.cytoscape.application.events](#)



Sự kiện



- Đăng ký danh sách người nghe sự kiện của bạn dưới dạng dịch vụ:
 Người nghe MyListener = MyListener mới();
 registerService(người nghe, NetworkAddedListener.class);
- Lớp của bạn chỉ cần triển khai
 handleEvent phù hợp:

```
lớp MyListener triển khai NetworkAddedListener {  
    công khai void handleEvent(NetworkAddedEvent ev) {  
        // xử lý dữ liệu của bạn  
    }  
}
```



Sự kiện (Lựa chọn)



- Nghe cho chính mình/trên -
(chúng ta thực sự không cố làm điều này khó khăn.-
.nhưng chúng ta đã thành công)
- Programma/callyselec/ngnodesoredges :
 - SetCyNetwork.SELECTEDtoTrueintheappropriate
bảng mặc định
- Ngheforselec/on:
 - Listenforchangestorows ([RowsSetListener](#))



Bộ nghe hàng



- Có lẽ bạn không muốn -
The networks you care about, hoặc-
`CyNetworkTableManager`
.trong hàm tạo của bạn
- Triển khai `RowsSetListener`
- Sự kiện `RowsSet`:
 - nguồn là `CyTable` có chứa cách hàng đã thay đổi -
`getColumnRecords(String column)` return a collection
of `RowSetRecords`
 - Cuối cùng, `eachRowSetRecord` là `row, column,`
và giá trị

```

lớp công khai NetworkSelectionLinker triển khai RowsSetListener {
    // Định nghĩa các biến

    công khai NetworkSelectionLinker(CyRootNetwork rootNetwork, CyEventHelper eventHelper) {
        this.rootNetwork = rootNetwork;
        this.eventHelper = eventHelper; this.viewManager
        = clusterManager.getService(CyNetworkViewManager.class);
    }

    công khai void handleEvent(RowsSetEvent e) {
        nếu (e.containsColumn(CyNetwork.SELECTED) || bỏ qua Lựa chọn)
            trở lại;

        CyNetworkView currentNetworkView = clusterManager.getNetworkView(); ignoreSelection = true; Bản
        đồ<CyNetwork, Boolean> stateMap =
        new HashMap<CyNetwork, Boolean>(); cho (Mạng con CySubNetwork: rootNetwork.getSubNetworkList()) {

            nếu (e.getSource().equals(subNetwork.getTable(CyNode.class, CyNetwork.LOCAL_ATTRS))) {
                đối với (RowSetRecord bản ghi: e.getColumnRecords(CyNetwork.SELECTED)) { Long uid =
                    record.getRow().get(CyIdentifiable.SUID, Long.class); Giá trị Boolean = (Boolean)record.getValue();
                    đối với (CySubNetwork sub2: rootNetwork.getSubNetworkList()) {

                        nếu (subNetwork.equals(sub2) || sub2.getDefaultNodeTable().getRow(uid) == null)
                            Tiếp tục;
                        sub2.getDefaultNodeTable().getRow(uid).set(CyNetwork.SELECTED, giá trị);
                    }
                }
            }

            } nếu (viewManager.viewExists(subNetwork)) {
                đối với (CyNetworkView xem: viewManager.getNetworkViews(subNetwork)) { nếu (!
                    view.equals(currentNetworkView)) { view.updateView(); }
                }
            }
        }

        eventHelper.flushPayloadEvents(); bỏ
        quaSelection = false;
    }
}

```



Bước 6: Dự án



- Add `onselectListener` to your app
 - Khi anode được chọn, hãy thay đổi hình dạng



Bước 7: Lệnh



- Cách thức truy cập chức năng/tính chất liên kết lỏng lẻo
 - Lỗi
 - Ứng dụng
- Ý tưởng chung:
 - Các lệnh được xuất sang CommandTool
 - Công cụ->CommandLineDialog
 - Các ứng dụng khác có thể thực hiện các lệnh đó bằng cách sử dụng gói org.cytoscape.command



Lệnh



Không gian tên có sẵn: cdd

hóa học

cụm

cụm hình ảnh

yêu cầu

cạnh

gpm1

nhóm

bố trí

mạng lưới

nút

máy phân tích rinalyzer

tiếp theo

phiên họp

bộ ứng dụng

cấu trúc Viz

bàn

xem

bản đồ vizmap

đường dẫn wiki

mạng lưới trợ giúp

Các lệnh có sẵn:

structureVizalign Thực hiện chèn cấu trúc theo trình tự Aonongroupofstructures. structureViz chú thích RIN Chú thích phần dư tương tác Aonnetwork(RIN) với attributeofthe
dư lượng tương ứng trong Chimera.

cấu trúc Vizclose

structureViz createRIN Tạo phần còn lại tương tác trên mạng (RIN) từ các mô hình hiện tại trong Chimera. cấu trúc Vizexit
Đóng tất cả các mô hình mở và cấu trúc exit Chimera Vizlaunch
LaunchChimera.

cấu trúc Vizlistmodels Liệt kê hiện đang mở Chimera models. cấu trúc Vizopen
OpennewstructuresinChimera cấu trúc Vizsend Send a command to Chimera.

cấu trúc Vizset Thay đổi cấu trúc VizseWngs cấu trúc Viz
showDialog Hiển thị hộp thoại điều hướng phân tử cấu trúc Viz syncColors Đồng
bộ hóa màu sắc giữa các phần còn lại của cấu trúc và các nút mạng.

helpstructure Cấu trúc

Vizopen Các đối số của Vizopen:

edgeList=[edgeColumn: giá trị | tên cạnh,...] | tất cả | đã chọn | chưa chọn: List of edges to open structures for modbaseID=<Chuỗi>: Modbase models to fetch
network=hiện tại | [cột: giá trị | tên mạng]: Mạng cho các nút/
cạnh đã chọn nodeList=[nodeColumn: giá trị | tên nút,...] | tất cả | đã chọn | chưa chọn: List of nodes to open structures for
pdbID=<Chuỗi>: PDB ID to fetch showDialog=true | false: Hiển thị hộp thoại MolecularStructureNavigator mở cấu trúc trong Chimera structureFile=<Tên>:
Structurefile

mạng lưới trợ giúp

Các lệnh có sẵn:

networkadd Thêm các nút và các cạnh của mạng (chúng phải nằm trong tập hợp hiện tại Aon) networkaddeedge Thêm một cạnh giữa hai
nút networkaddnode Thêm một nút mới vào một mạng networkclone Tạo một bản
sao của mạng hiện tại networkcreate Tạo một mạng mới networkcreatea
Rribute Tạo một Uriattribute (cột) trong bảng mạng networkcreateempty Tạo một
mạng trống networkdelete Xóa các nút sắp xếp các cạnh
khỏi một mạng networkdeselect Bỏ chọn các nút sắp xếp các cạnh trong mạng networkdestroy Hủy (xóa) một mạng
networkexport Xuất một mạng và chế độ xem của nó thành một tệp networkget
Trả về một mạng networkgeta Rribute Lấy giá trị cho một mạng a Uriattribute
networkgetproper+ es Lấy giá trị thuộc tính trực quan cho một mạng networkhide Ẩn
các nút sắp xếp các cạnh trong mạng networkimportfile Nhập một
mạng từ một tệp networkimport url Nhập một mạng từ một URL networklist Liệt kê
tất cả các mạng có sẵn networklista Rribute
Liệt kê tất cả các Uriattribute (cột) cho các mạng networklistproper+es Liệt kê tất cả các
tệp tải mạng Aes trực quan thích hợp Tải tệp mạng (ví dụ XGMML) url tải mạng Tải tệp mạng (ví dụ
XGMML) từ url

networkrename Đổi tên anetwork networkselect

Chọn các nút được sắp xếp theo các cạnh trong mạng

networksetaRribute Đặt giá trị trong bảng mạng networksetcurrent Đặt

mạng hiện tại networksetproper+es Đặt mạngvisualproperAes

networkshow Hiển thị các nút và cạnh ẩn



Thực hiện lệnh



1. Xem lệnh có khả dụng không
 1. Sử dụng `org.cytoscape.command.AvailableCommands`
2. GetaTaskManager
3. PopulateCreateaTaskIterator sử dụng `CommandExecutorTaskFactory`
4. Sử dụng TaskObserver để nhận kết quả

```

// Lớp Quản lý của tôi //
Từ http://github.com/RBVI/StEMAPApp
lớp công khai StEMAPManager
{ CyServiceRegistrar cuối cùng serviceRegistrar;
CyEventHelper cuối cùng eventHelper;
CommandExecutorTaskFactory commandTaskFactory = null;
SynchronousTaskManager taskManager = null;
AvailableCommands availableCommands = null;

public StEMAPManager(cuối cùng CyServiceRegistrar cyRegistrar)
{ this.serviceRegistrar = cyRegistrar;
}

công khai <S> S getService(Lớp<S> serviceClass) {
    trả về serviceRegistrar.getService(serviceClass);
}

public void executeCommand(String namespace, String lệnh,
                           Map<String, Object> args, người quan sát TaskObserver) {
    nếu (commandTaskFactory == null)
        commandTaskFactory = getService(CommandExecutorTaskFactory. class);
    nếu (availableCommands == null)
        availableCommands= getService(AvailableCommands.class);
    nếu (taskManager == null)
        taskManager = getService(SynchronousTaskManager.class); nếu
(availableCommands.getNamespace(namespace) == null ||
availableCommands.getCommands(namespace).contains(command)) throw new
RuntimeException("Không tìm thấy lệnh "+namespace+" "+command);
    TaskIterator ti = commandTaskFactory.createTaskIterator(namespace, command, args, Observer); taskManager.execute(ti);

}

```

```

// Tải tệp PDB vào UCSF Chimera // Từ http://
github.com/RBVI/StEMAPApp

public void loadPDB(String pdbPath, String extraCommands) { Bản đồ<Chuỗi, Đối tượng> args =
    new HashMap<>(); nếu (pdbPath != null) args.put("structureFile",
    pdbPath); nếu không thi
    args.put("pdbID", getPDB());

    args.put("showDialog", "true"); // Giả sử
    rằng lớp gọi thực hiện TaskObserver
    // StructureViz sẽ cung cấp cho chúng ta một chuỗi văn bản chứa // tên và số
    của mô hình đã mở executeCommand("structureViz",
    "open", args, this);

    thử
    { // Đợi mọi thứ xử lý
        Thread.sleep(500);
    bắt (Ngoại lệ e) {}

    nếu (extraCommands != null) { args =
        new HashMap<>(); args.put("lệnh",
        extraCommands); executeCommand("structureViz",
        "gửi", args, null);
    }
}
}

```



Bước 7: Dự án



- Viết một ứng dụng mới
 - Thực hiện lệnh bạn đã xuất trong Bước 5
 - Cần phải thiết lập các giá trị cho các biến có thể điều chỉnh.



Mã mẫu



- Sampleapps -

<https://github.com/cytoscape/cytoscape-samples>

- Có thể sử dụng mẫu cho ứng dụng của riêng bạn

- Realappexample: - SIREN:

Đăng ký mạng lưới điều tiết- <http://baderlab.org/>

[PegahKhosravi/SIREN](https://github.com/PegahKhosravi/SIREN)

- <https://github.com/BaderLab/SirenApp>

- Một số ứng dụng trên appstore liên kết đến nguồn của chúng mã(ví dụ DynNetwork)



Mã mẫu



- Kho lưu trữ UCSFRBVI
 - <https://github.com/RBVI/>



- <http://chian/.ucsd.edu/cytoscape-3.2.1/API/> app-api
 - SimpleappAPI (để giúp Cy2plugin developers; không dành cho bundleapps)
- API nhiệm vụ cốt lõi
 - Các tác vụ cấp cao thường được sử dụng - ví dụ: tải mạng/kiểu/bảng, áp dụng bộ lọc



- mô hình-api
 - Mạng, mô hình bảng
- sự kiện-api
 - Mô hình sự kiện
- công việc-api
 - Nhiệm vụ, TaskFactory
- bố trí-api
 - Định nghĩa bố cục



- trình bày/trên-api
 - Visualpropertydefini/ons
 - .viewmodel-api
 - Sezngvisualproper/es
- vizmap-api
 - Visualmapping
- swing-u/l
 - GUIu/li/es;egfileload/savedialog;colorchooser hộp thoại



- equa/ons-api
 - Định nghĩa CyTableequa/ons(likeExcelfunc/ons)
- group-api
 - Làm việc với CyGroups (hay còn gọi là metanodes)
- io-api
 - Định nghĩa nhà nhập khẩu/xuất khẩu; luồng đọc
- (swing-)applica/trên-api
 - Truy cập trạng thái và sự kiện cấp hệ thống (ví dụ: UIpanels, thanh công cụ, menu, mainJFrame)



- thuôc tính-api
 - Truy cập/xác địnhhệ thốngproper/es; Truy cậpmứcđộphiênproper/es
- dịch vụ-api
 - Tóm tắtCyAc/vator;dịch vụ(không)đăng ký/trên;dịch vụlistenerđăng ký/trên
- phiên-api
 - Truy cập tên tệp phiên hiện tại; chụp ảnh nhanh phiên hiện tại



- lệnh-thực thi-api
 - Hỗ trợ cho execu/ngcommandsfromtasks
- nhóm-api
 - Hỗ trợ cho CyGroups, bao gồm thu gọn/mở rộng, tổng hợp thuộc tính/trên và trực quan hóa/trên/trên



Thực hành tốt nhất



- Các gói phải giảm thiểu những gì chúng xuất
 - Đừng làm bất cứ điều gì nếu không có ai yêu cầu và bạn chưa sẵn sàng cam kết lâu dài
- Bundle each variant should do as little work as possible
 - Tốt nhất là chỉ cần đăng ký dịch vụ
 - Do expensive initialization as lazily as possible
 - E.g. during menu activation / when



Lặt Vặt



- Ứng dụng/thông tin về trạng thái/trên
 - Giao diện người dùng
 - Xử lý rất nhiều yêu cầu dịch vụ



Nhân trợ giúp



- cytoscape-helpdesk@googlegroups.com
- cytoscape-discuss@googlegroups.com
- scooter@cgl.ucsf.edu



Slide và Giải pháp



- Slides có sẵn tại
 - [hlp://www.cgl.ucsf.edu/home/scooter/
Cytoscape3DevTut/ISMBslides.pdf](http://www.cgl.ucsf.edu/home/scooter/Cytoscape3DevTut/ISMBslides.pdf)
- Mysolu/onsavailableat: -
[hlp://www.cgl.ucsf.edu/home/scooter/
Cytoscape3DevTut/solu/ons.zip](http://www.cgl.ucsf.edu/home/scooter/Cytoscape3DevTut/solu/ons.zip)



Kết luận



- Bây giờ bạn đã sẵn sàng để viết những điều không tầm thường

Cytoscapeapps • Hướng

dẫn này là một phần của Cytoscape

thang phát triển: [hlp://](http://wiki.cytoscape.org/Cytoscape_3/AppDeveloper/)

wiki.cytoscape.org/Cytoscape_3/AppDeveloper/

[Thang ứng dụng Cytoscape](#)

- Bạn đã đi được một nửa chặng đường của thang rồi! • Nhiều API

chưa được đề cập

- Cộng đồng mã nguồn mở: ~~lấy~~ cắp mượn từ người khác
 - Luôn luôn ghi nhận nguồn



Có câu hỏi nào không?

