

# Coding for Education

## Coding for the last one billion



Sayamindu Dasgupta  
[sayamindu@laptop.org](mailto:sayamindu@laptop.org)

“Our focus is not the next billion, but  
the last billion”

*Nicholas Negroponte*

# Things I will cover in the talk

- Sugar
  - Sugar HIG (Human Interface Guidelines)
  - Things that make up Sugar
  - Activities
  - Journal
  - Collaboration
- Rainbow
  - How to keep it happy and how to stay away from it

# A question

- Does any of the items below make sense to a kid in rural India, who has never been to a city ?
  - Files
  - Folders
  - Desktops



Sugar tries to address this challenge

# Sugar is something...

- That can make your teeth go bad



But Sugar is also....

# *Advertisement time*

“The first serious attempt to create a user interface that is based on both cognitive and social constructivism”

# Core Sugar UI Principles

Abstraction

Focus

Familiarity

# Abstraction

- Less culture specific
  - *No confusion about a “desktop”*
- Less resource usage
  - No fancy icons/UI elements to draw

# Focus

- Run everything is fullscreen
  - *No distractions*
- Focus on a core set of tools
  - Centered around tasks

# Familiarity

- Attempt to use universal, familiar languages and metaphors for representation
  - *Everyone usually knows what a “diary” or a “journal” is*
- People-centric, activity-centric
  - Attempt to mimic the real life of kids

# Components of the Sugar UI

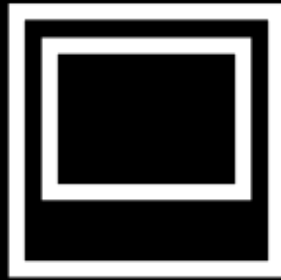
People



# Activities



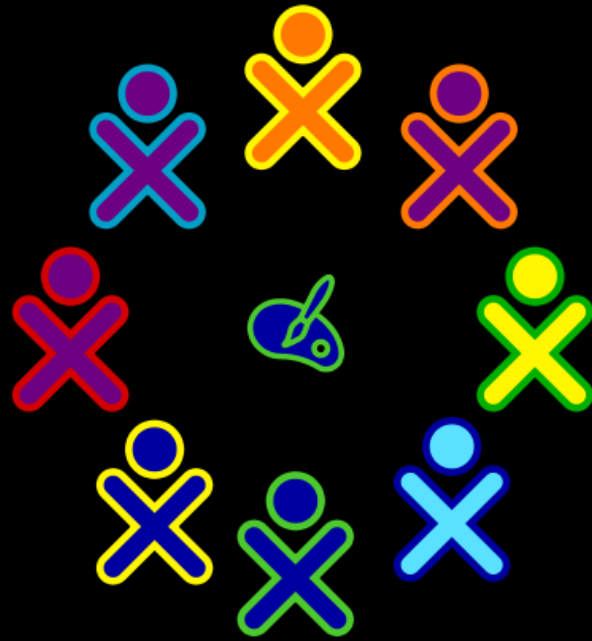
# Objects



# Community

Community forms a central theme in Sugar's interaction model

# People work on activities together



to produce objects

# The technology behind Sugar



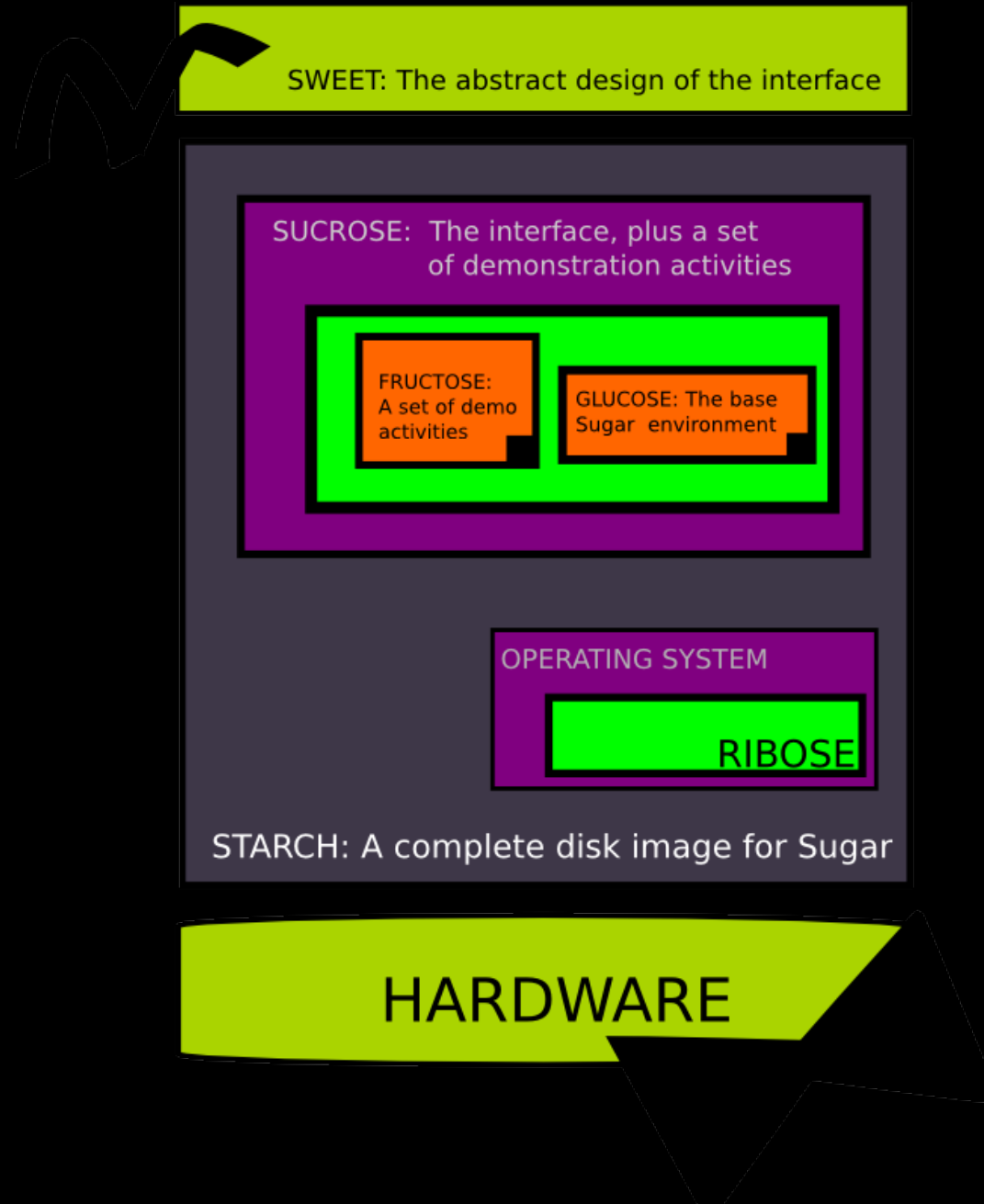
# The Sugar Taxonomy

- **Sweet:** The abstract design of the interface
  - the taste of sugar
- **Glucose:** The base Sugar environment
  - the fundamental, simple sugar used by all life forms
- **Fructose:** A set of demonstration activities
  - the main sugar in fruit, which is how we're supposed to get our sugar
- **Sucrose:** The interface, plus a set of demonstration activities
  - "table sugar", the kind you buy in the store. It consists of glucose and fructose, combined

# The Sugar Taxonomy (..contd)

- **Ribose:** The base Linux distribution being used by Sugar
  - the sugar used by all lifeforms to control their hardware, in the form of RNA. It's important, but not sweet
- **Starch(es):** A complete disk image for Sugar
  - starch is composed of multiple sugars bonded together)
- **Honey:** activities developed by 'in the wild' vendors
  - a sweet fluid made by bees, mostly Fructose

# ..and this is how it all fits together



# Activities: The technical perspective

- Packaged as “XO” bundles (zip files)
- Usually written in Python. Alternatives include
  - Smalltalk (Etoys, Scratch)
  - HTML/Javascript (SocialCalc)
  - C/C++ (X activity)
- Usually uses PyGTK or PyGame for the UI

# Activities: Directory Layout

- Toplevel MANIFEST file listing all the files and directories
- activity/activity.info containing the metadata

# activity.info

[Activity]

name = Image Viewer

bundle\_id = org.laptop.ImageViewerActivity

class = ImageViewerActivity.ImageViewerActivity

icon = activity-imageviewer

activity\_version = 4

show\_launcher = no

mime\_types =

image/bmp;image/gif;image/jpeg;image/png;image/tiff

license = GPLv2+

# Activities: the actual code

- The value of the class field in *activity.info* defines the name of the class which contains the runnable activity code
- This class has to have “sugar.activity.Activity” as a baseclass

# Example

```
from sugar.activity import activity

class ImageViewerActivity(activity.Activity):
    def __init__(self, handle):
        activity.Activity.__init__(self, handle)
        self.view = ImageView.ImageView()
        self.set_canvas(self.view)
        self.view.show_all()
```

# Activity: Handling Data

- All objects created by the laptop's owner is kept in a journal
- The journal is presented through a diary or a blog like interface

# Screenshot time

Search my journal

Anything

Anyone

Anytime

Type a title

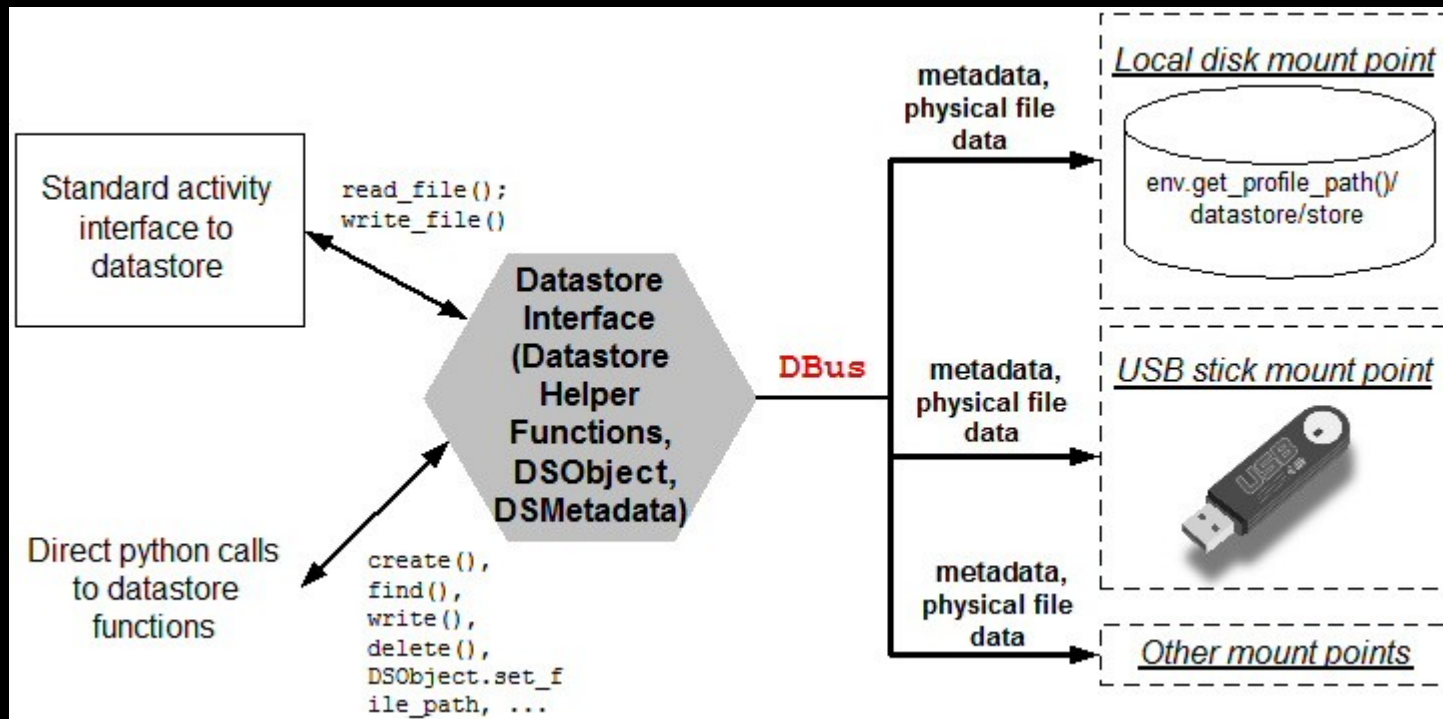
Add new entry

- ▶ Added 8 objects to  **My Homework** ▶ 5 minutes ago
- ▶ Took 4 photos of  **Rain Forest** ▶ with    yesterday
- ▶ Painted a picture of  **Our school** ▶ with  2 days ago
- ▶ Read  **History of Thailand** ▶ with   3 days ago
- ▶ Wrote  **History of Thailand** ▶ 1 week ago
- ▶ Painted a picture of  **My Cat** ▶ with  1 week ago
- ▶ Looked at  **A Photo of my Cat** ▶ with  1 week, 2 days ago
- ▶ Wrote  **History of Thailand** ▶ 1 week ago, 3 days ago
- ▶ Painted a picture of  **Our school** ▶ with  1 week ago, 3 days ago
- ▶ Looked at  **Uruguay - Wikipedia, the fr...pedia** ▶ with  1 week, 2 days ago

# Writing to the journal

```
class AnnotateActivity(activity.Activity):
...
    def write_file(self, file_path):
        logging.debug('WRITING FILE ...')
        #save some metadata
        self.metadata['current_page'] = '3'
        #save the file itself
        f = open(file_path, 'w')
        try:
            f.write("Hello, Children of the World")
        finally:
            f.close()
```

# The Journal Backend: Datastore



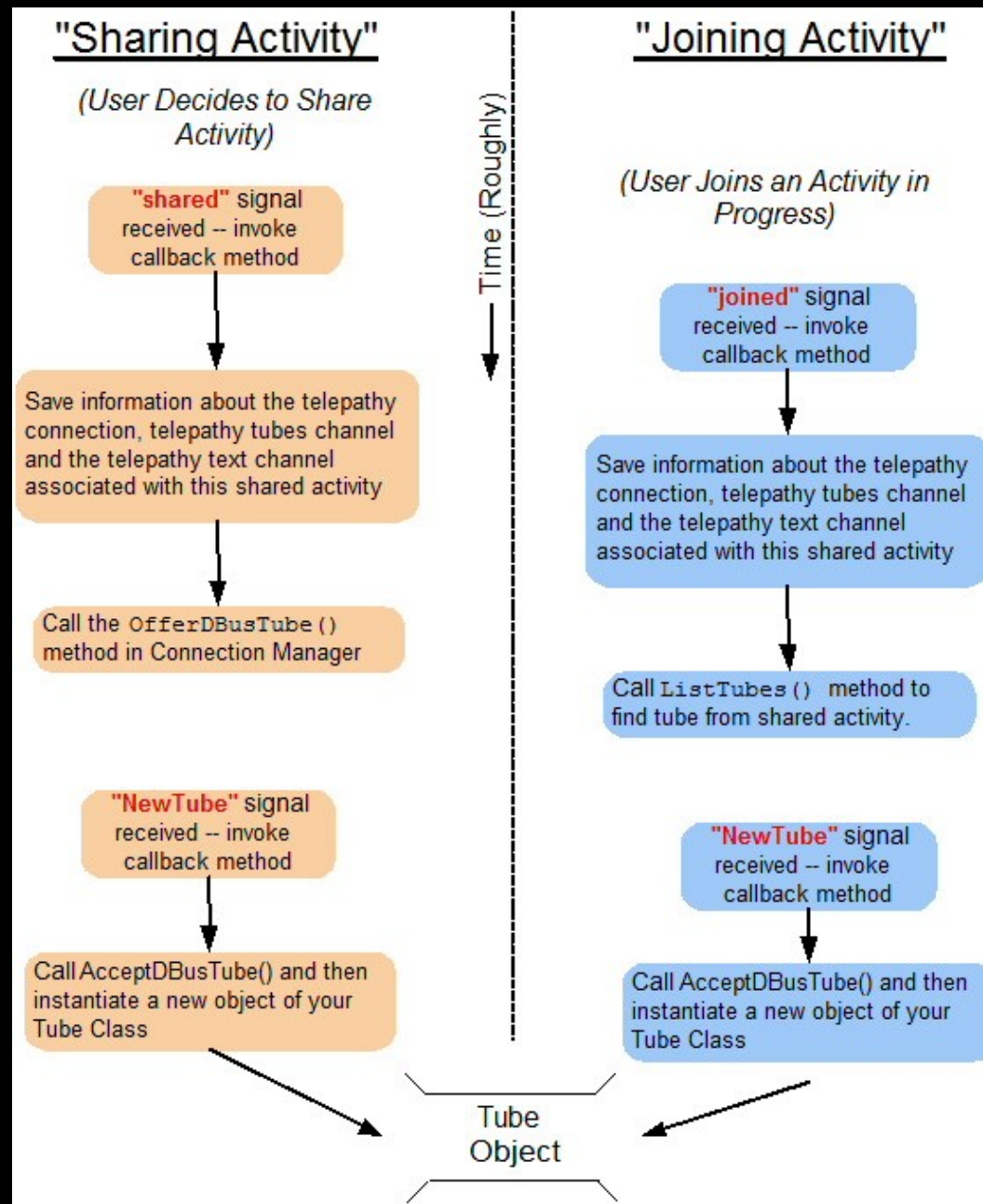
# Sugar Collaboration

- Collaboration in Sugar activities are implemented over Telepathy
  - Telepathy is a framework for real time conversations (eg: IM, voice chat, etc)
  - Telepathy uses Connection Managers to implement the backend protocols, in case of Sugar, the two CMs are Gabble and Salut
  - The CM objects in Sugar are managed by the *Presence Service*
- A shared activity is implemented via a XMPP MUC (Multi-User-Chatroom)

# Sugar Collaboration (..contd)

- A text channel is provided by the PS between each shared activity
- On top of the Text Channel, PS provides a Tube channel for data transfer and more
- Tubes can be
  - Dbus Tubes
  - Stream Tubes

# Activity Sharing



# Best way to learn about collaboration

```
$ git clone git://dev.laptop.org/projects/hellomesh
```

# Sugar Security: Rainbow

- Implementation of the Bitfrost specification
- Works by isolating activities from one another
- Isolation is (currently) implemented by generating a new uid (and perhaps a new gid) for each program it is asked to run

# Keeping Rainbow Happy

- Home directory is not persistent
- Writeable areas of the filesystem are subdirectories of `$SUGAR_ACTIVITY_ROOT`, and *not* `$HOME`
- “Standard” dotfiles and dotdirectories may not be present, and even if they are, they may not be useful

# Learning more

- [http://wiki.laptop.org/go/Sugar\\_almanac](http://wiki.laptop.org/go/Sugar_almanac)
  - Incredible “how do I” snippets
- <http://sugarlabs.org/>
  - For all things Sugary
- [http://wiki.laptop.org/go/Summer\\_of\\_Code/Ideas](http://wiki.laptop.org/go/Summer_of_Code/Ideas)
  - Interesting ideas you can adopt
- <http://wiki.laptop.org>
  - For the rest of the stuff

# Questions ??

(you can ask stuff about Microsoft as well)  
There will be no manhandling ;-)



That's all folks ;-)