edX Insights and XBlocks
Open platforms for educational innovation

Piotr Mitros
Chief Scientist, edX
What is edX?

Research
Understanding learning

Quality
Improve learning

Access
Educate 1 billion people
What is edX Learning Sciences?

- MITx platform
- Open-ended grading
- Insights
- Crowdsourcing

- We're hiring!
STEM Resource Finder

Our STEM Resource Finder features some of the best of our free, open source educational activities, models and software tools. You can search by keyword or filter by subject, grade level and type to find the right resources for your learning goals.

By Subject

- Biology
- Chemistry
- Earth and Space Science
- Engineering
- Mathematics
- Physics

By Grade Level

- Elementary School
- Middle School
- High School
- Higher Education
In the beginning...
A diode is highly non-linear!

\[ I = I_s \left( e^{\frac{V}{N}} - 1 \right) \]

SPEAKER 1: Well, the diode is indeed not a linear system.

If, for example, I choose a voltage here and I choose twice that voltage over here, then the curve responding at this voltage is very small, but the curve responding here is much more than twice the current over there. Therefore, this is not a linear system.
A diode is highly non-linear!

\[ I = I_S \left( e^{\frac{V}{V_T}} - 1 \right) \]

SPEAKER 1: Well, the diode is indeed not a linear system.
If, for example, I choose a voltage here and I choose twice that voltage over here, then the curve responding at this voltage is very small, but the curve responding here is much more than twice the current over there.
Therefore, this is not a linear system.
A diode is highly non-linear.

\[ I = I_d \left( e^{\frac{V}{V_T}} - 1 \right) \]

SPEAKER 1: Well, the diode is indeed not a linear system.

If, for example, I choose a voltage here and I choose twice that voltage over here, then the curve responding at this voltage is very small, but the curve responding here is much more than twice the current over there. Therefore, this is not a linear system.
A diode is highly non-linear.

\[ I = I_d \left( e^{\frac{V}{V_T}} - 1 \right) \]

SPEAKER 1: Well, the diode is indeed not a linear system.

If, for example, I choose a voltage here and I choose twice that voltage over here, then the curve responding at this voltage is very small, but the curve responding here is much more than twice the current over there.

Therefore, this is not a linear system.
A diode is highly non-linear!

\[ I = I_D \left( e^{\frac{V_D}{V_T}} - 1 \right) \]

SPEAKER 1: Well, the diode is indeed not a linear system.
If, for example, I choose a voltage here and I choose
twice that voltage over here, then the curve responding at
this voltage is very small, but the curve responding here
is much more than twice the current over there.
Therefore, this is not a linear system.
A diode is highly non-linear:

\[ I = I_d \left( e^{\frac{V}{V_T}} - 1 \right) \]

SPEAKER 1: Well, the diode is indeed not a linear system.
If, for example, I choose a voltage here and I choose
twice that voltage over here, then the curve responding at
this voltage is very small, but the curve responding here
is much more than twice the current over there.
Therefore, this is not a linear system.
Goal: Lightweight pedagogical innovations

- Ridiculously easy to use
- You don't need your sysadmin
  - Cannot take down the LMS
  - Hostable and hosted
- Build a community:
  - Reuse in courses
  - Open standard (not just edX)
(Contrast to Services)

- LTI, Tincan, ...

Still want to support, but:

- Hosting, “weight”
- Reliability
- Reuse
Goal: Lightweight pedagogical innovations

- Ridiculously easy to use
- You don't need your sysadmin
  - Cannot take down the LMS
  - Hostable and hosted
- Build a community:
  - Reuse in courses
  - Open standard (not just edX)
import json
from mitxmako.shortcuts import render_to_response, render_to_string
from x_module import XModule
from lxml import etree

class Module(XModule):
    id_attribute = 'filename'

    def get_state(self):
        return json.dumps({})

@classmethod
    def get_xml_tags(c):
        return ['html']

def get_html(self):
    if self.filename is None:
        xmltree = etree.fromstring(self.xml)
        textlist = [''.join([xmltree.text] + [etree.tostring(i) for i in xmltree] + [xmltree.tail])
                   for i in xmltree]
        textlist = [i for i in textlist if type(i) == str]
        return ''.join(textlist)
    try:
        filename = 'html/' + self.filename
        return self.filestore.open(filename).read()
    except:  # For backwards compatibility. TODO: Remove
        return render_to_string(self.filename, {'id': self.item_id})

    def __init__(self, system, xml, item_id, state=None):
        XModule.__init__(self, system, xml, item_id, state)
        xmltree = etree.fromstring(xml)
        self.filename = None
        filename_1 = xmltree.xpath('/html/@filename')
        if len(filename_1) > 0:
            self.filename = str(filename_1[0])
class HtmlFields(object):
    def __init__(self, html_contents=None, scope=Scope.content):
        self.html_contents = html_contents
        self.scope = scope

    def get_field(self, name):
        if name in self.html_contents:
            return self.html_contents[name]
        return None

class HtmlField(DictField, XMLElement):
    def __init__(self, name, html_content=None, module=Module):
        self.name = name
        self.html_content = html_content
        self.module = module

    def get_field(self, name):
        if name in self.html_content:
            return self.html_content[name]
        return None

class HtmlDescriptor(HttpFields, XMLElement, EditingDescriptor):
    def __init__(self, html_content=None, module=Module):
        self.html_content = html_content
        self.module = module

    def get_field(self, name):
        if name in self.html_content:
            return self.html_content[name]
        return None

@classmethod
def get_field(cls, name, html_content, module=None):
    return HtmlFields(html_content).get_field(name)

@classmethod
def get_field_from_xml(cls, name, html_content, module=None):
    return HtmlField(name, html_content, module).get_field(name)

@classmethod
def get_field_from_html(cls, name, html_content, module=None):
    return HtmlDescriptor(html_content, module).get_field(name)
def definition_to_xml(self, resource_fs):
    """If the Content is valid xml, write them to filename.xml. Otherwise, write just the .html filename" (meta-attrs="...[" to filename.xml and the html to filename.html..."
    try:
        return etree.fromstring(self.data)
    except etree.XMLSyntaxError:
        pass

    # Not proper format. Write html to file, return an empty tag
    pathname = os.path.dirname(resource_fs.path)
    filename = os.path.basename(resourceFs.path)
    filepath = os.path.join(pathname, filename)
    resourceFs.openFile(filepath, w) as file:
        file.write(self.data.encode('utf-8'))
    # write out the relative name
    relname = os.path.basename(
    elt = etree.Element('html', namespace)
    elt.set('name', relname)
    return elt

@property
def editable_metadata_fields(set):  
    """Reduce any metaData from the editable fields which have their own editor or shouldn't be edited by user."""
    subset = super(HtmlDescriptor, self).editable_metadata_fields
    if 'empty' in subset:
        del subset[empty]
    return subset

class AboutDescriptor(HtmlDescriptor):
    """These pieces of course content are treated as HtmlModules but we need to overload where the templates are located in order to be able to create new ones"
    template_dir_name = "about"

class StatisticDescriptor(HtmlDescriptor):
    """These pieces of course content are treated as HtmlModules but we need to overload where the templates are located in order to be able to create new ones"
    template_dir_name = "statistic"

class CourseInfoDescriptor(HtmlDescriptor):
    """These pieces of course content are treated as HtmlModules but we need to overload where the templates are located in order to be able to create new ones"
    template_dir_name = "courseinfo"
XModules → XBlocks

```python
from string import Template
from .core import XBlock, String, Scope
from .fragment import Fragment

class HtmlBlock(XBlock):
    """Render content as HTML.

    The content can have $PLACEHOLDERS, which will be substituted with values from the context.
    """

    content = String(help="The HTML to display", scope=Scope.content, default=u"<b>DEFAULT</b>")

def fallback_view(self, view_name, context):
    return Fragment(Template(self.content).substitute(**context))
```
• Normalization
• Self-assessment
• Randomized problems
• Stop feature
• Multistage adaptive testing
Missing piece

Standard way to process data
Open Learning Analytics: An Integrated and Modularized Platform (2011)

George Siemens, Dragan Gasevic, Caroline Haythornthwaite, Shane Dawson, Simon Buckingham Shum, Rebecca Ferguson, Erik Duval, Katrien Verbert, Ryan Baker
Off-line analysis

Source: RELATE (d. pritchard, d. seaton, et. al.)
Dashboards
Responsive System

Analyze what the student has done

Respond with a hint, resource, etc.
Similar Goals

• Ridiculously easy to use
• You don't need your sysadmin
  – Cannot take down the LMS
  – Hostable and hosted
• Data access without PII
• Build a community
  – Open standard (not just edX)
  – Reuse each other's work
SOA
Used to automatically generate student, instructor, and admin dashboard. Embeddable to show student progress, etc.

Query interface to other modules

Queriable data sources
- Read replicas of LMS databases
- Read replicas of other module's databases

User clickstream and other logged events

Discovery Interface

view
Serves HTML to users

query
Serves data via SOA

period task

event handler

add properties

Module Data Store
- MongoDB Database
- Filesystem-like Abstraction
- Cache
LMS Servers

Core Server
- 99.9% uptime
- Student progress
- Basic grades
- Etc.

Aux server
- Good uptime
- IRT
- Teaching analytics

Prototype server
- Best-effort uptime
- Experimental analytics
- Research projects
<table>
<thead>
<tr>
<th>Recommender</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRT</td>
</tr>
<tr>
<td>Concept Map</td>
</tr>
<tr>
<td>Data Export</td>
</tr>
<tr>
<td>SNA</td>
</tr>
<tr>
<td>Data Module</td>
</tr>
<tr>
<td>Data Module</td>
</tr>
<tr>
<td>Data Module</td>
</tr>
<tr>
<td>Data Module</td>
</tr>
<tr>
<td>Dummy Module</td>
</tr>
<tr>
<td>Sakai</td>
</tr>
<tr>
<td>Moodle</td>
</tr>
<tr>
<td>LC</td>
</tr>
<tr>
<td>edX</td>
</tr>
</tbody>
</table>

Build on a common set of queries, event properties, and database tables.

Provides a common set of queries and event properties.
Data module (dummy)

```python
@query()
def get_grades(course):
    ''' Dummy data module. Returns grades
    ...

    grades = 3*numpy.random.randn(1000,4)+ \n    12*numpy.random.binomial(1,0.3,(1000,4))+40
    return grades
```
@view()
def plot_grades(fs, query, course):
    grades = query.get_grades(course)
    filename = course+'_'+str(time.time())+'.png'
    title("Histogram of course grades")
    hist(grades)
    f = fs.open(filename, "w")
    savefig(f)
    f.close()
    fs.expire(filename, 5*60)
    return "<img src="+fs.get_url(filename)+">"
Histogram of course grades
@event_handler()
def djt_event_count_event(mongodb, events):
    ''' Count events per actor. '''
    for evt in events:
        if evt.actor:
            collection = mongodb['user_event_count']
            collection.update({'actor': evt.actor},
                               {'$inc':{'event_count':1}},
                               upsert = True)
@query()
def user_event_count(mongodb, user):
    ''' Number of events seen by a specific user
    '''
    collection = mongodb['user_event_count']
t = list(collection.find({'actor': user}))
if len(t):
    return t[0]['event_count']
return 0

@view()
def user_event_count(query, user):
    ''' Number of events seen by a specific user
    '''
    return user + " saw " + str(query.user_event_count(user)) + " events."
Status

• Early prototype – may evolve a lot – but a few early adopters
• Being productized within edX
• Still need a schema/data model
• We want your help! Does it meet your needs?
Questions?

• Prototype at:
  https://github.com/edx/insights

• Mailing list at:
  https://groups.google.com/forum/#!forum/insights-dev

• edX prototype use-cases at:
  https://github.com/edx/edxanalytics