Introduction to Drupal Commerce

DrupalCamp China 2013

David Suissa: DYdave

March 23, 2013

Overall plan

- History: where does DC comes from?
- Getting started with DC:
  - Commerce Kickstart: Demo time!
  - Objects
  - Data models
- Working with Rules and Commerce:
  - Pricing rules
  - Discount/coupons
  - Shipping
  - Payment & Checkout
- Advanced Topics
- Drupal Commerce in China
- Roadmap and what’s coming up
- Questions

Ubercart History (D6)

- Ubercart is a set of modules created by the Commerce Guys for Drupal 6
- Ubercart is the most commonly used contributed commerce package for building online stores in Drupal 6
- Problem:
  Ubercart was built around a simple store model and expanded via contributed modules to do many things it was never designed to do

From Ubercart to Drupal Commerce

- Leveraging Drupal 7’s increased flexibility with Fieldable Entities and more Views and Rules integration.
- Drupal Commerce is a very flexible E-commerce system, by design, much more than Ubercart could ever be.
- Commerce customizations are more in the database than in code
  - Tasks that used to require heavy handed overrides, can now be configured through a U
  - These database items are usually exportable, so you can move them to code if you desire (views, rules)
- As a results:
  Commerce satisfies more use-cases with less work

Ubercart to Commerce

What can you *do with it?!*

Ubercart offers a wide **feature set, but!** operating outside of the core feature! set is difficult at best."

“Application” mindset focusing on the! modules’ **performance out-of-the-box."**

Sacrificed API functionality and i18n! for UI based customizations."

What can you *build with it?!*

Commerce modules offer **core e-commerce"**
systems and components, providing basic!
functionality with amazing flexibility."

“Framework” mindset focusing on loosely!
coupled modules and **adaptability."**

Privileges developers and site builders at!
the core level, administrators and reusable!
feature development at the profile level."

What types of Commerce

Anything is possible...

- Donations
- Physical Products (clothes, accessories, shoes, etc…)
- Digital Products (PDF downloads, e-books, images, etc…)
- Subscriptions (Service based platforms with recurring payments, etc…)
- Event registration (Concert tickets booking, events booking, etc…)

DC in Numbers

Commerce adoption

increased after 1.0

- More than 25K sites using Drupal Commerce.
- 180K+ downloads

First distribution

by Commerce Guys

First Release on June 2011

- Top used and downloaded distribution on drupal.org
- 145K+ downloads

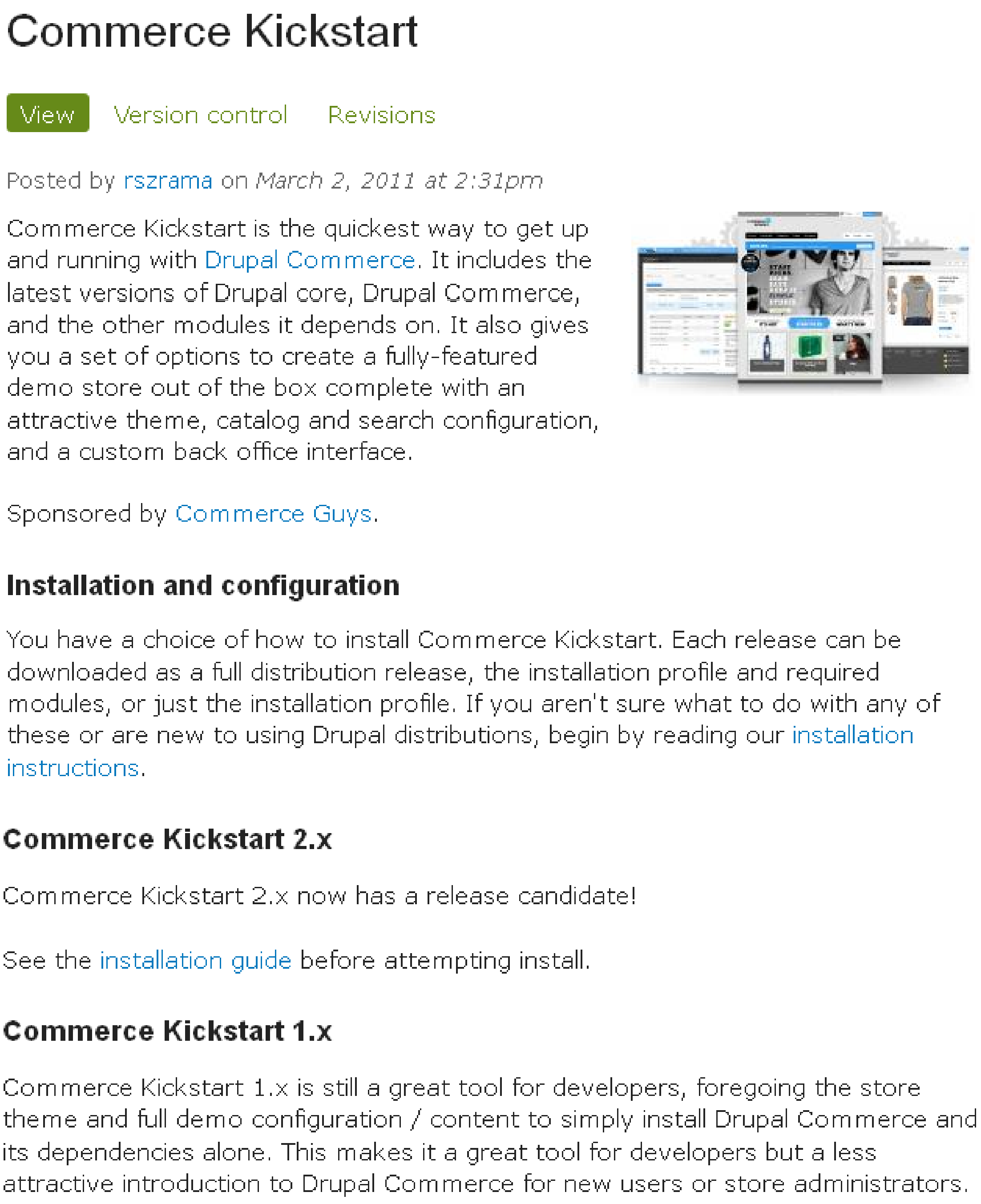
Getting Started

Drupal Commerce is an eCommerce framework, focusing on what you can **build with it, not simply what you can** do with it out of the box.

Commerce Kickstart

- Commerce Kickstart (http://drupal.org/project/commerce_kickstart) is an installation profile that includes the following:
  - Basic Drupal Commerce with all its dependencies
  - Drupal Commerce configuration with example store content
- Details on installing Commerce Kickstart: http://drupal.org/node/1291122
- For more information on Drupal installation profiles: http://mostrey.be/howto-install-drupal-installation-profile
- Installing Commerce Kickstart on share hosting:

  http://browarddrupal.org/PastMeetings/april-2012



Commerce Kickstart

Posted by rszrama on *March 2, 2011 at 2:31pm*

Commerce Kickstart is the quickest way to get up and running with Drupal Commerce. It includes the latest versions of Drupal core, Drupal Commerce, and the other modules it depends on. It also gives you a set of options to create a fully-featured demo store out of the box complete with an attractive theme, catalog and search configuration, and a custom back office interface.

Sponsored by Commerce Guys.

Installation and configuration

You have a choice of how to install Commerce Kickstart. Each release can be downloaded as a full distribution release, the installation profile and required modules, or just the installation profile. If you aren't sure what to do with any of these or are new to using Drupal distributions, begin by reading our installation instructions.

Commerce Kickstart 2.x

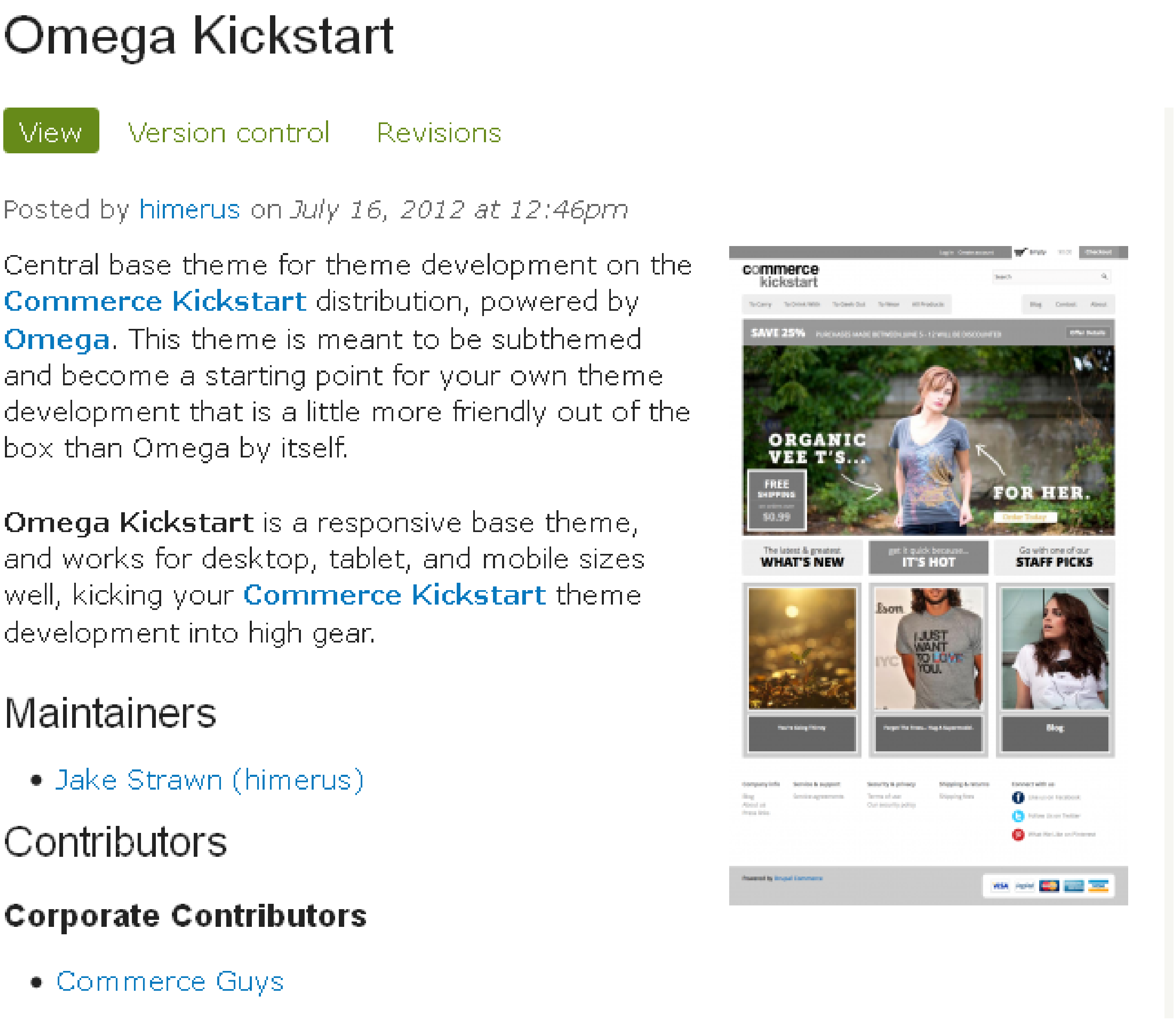
Commerce Kickstart 2.x now has a release candidate!

See the installation guide before attempting install.

Commerce Kickstart 1.x

Commerce Kickstart 1.x is still a great tool for developers, foregoing the store theme and full demo configuration / content to simply install Drupal Commerce and its dependencies alone. This makes it a great tool for developers but a less attractive introduction to Drupal Commerce for new users or store administrators.

http://drupal.org/project/commerce_kickstart



Omega Kickstart

Posted by himerus on *July 16, 2012 at 12:46pm*

Central base theme for theme development on the **Commerce Kickstart** distribution, powered by **Omega**. This theme is meant to be subthemed and become a starting point for your own theme development that is a little more friendly out of the box than Omega by itself.

Omega Kickstart is a responsive base theme, and works for desktop, tablet, and mobile sizes well, kicking your **Commerce Kickstart** theme development into high gear.

Maintainers

- Jake Strawn (himerus)

Contributors

Corporate Contributors

- Commerce Guys

http://drupal.org/project/omega_kickstart

Demo Time!
Hands on!

Kickstart Installation on simplytest.me in 5 minutes

Custom demo setup by Vincent Zhang and DYdave.
Thanks very much for your help at previous Workshop.
(March 16)

A Few Starters

- Get familiar with Views 3. Here are some excellent videos (~30) from NodeOne: http://dev.nodeone.se/en/taming-the-beast-learn-views-with-nodeone

- Get familiar with Rules. Here are some more great videos (~17) from NodeOne: http://dev.nodeone.se/en/learn-rules-with-nodeone-part-1-overview

Basic Concepts

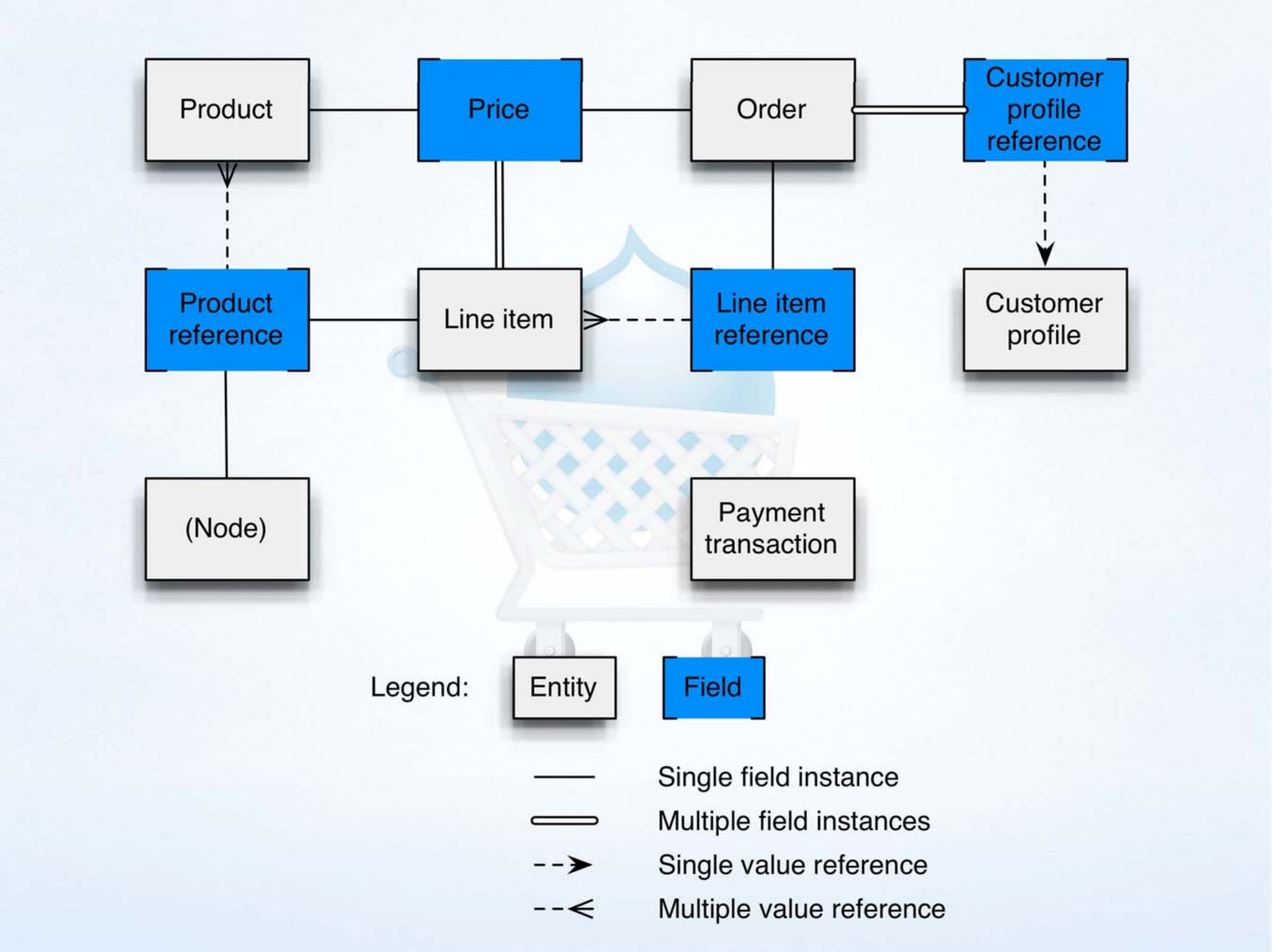
- Products aren’t nodes, they’re entities
  - You must setup a display node which references products
- Most configurable action run through rules
- The checkout process is configurable
  - There are 4 steps in the process that panes can be placed into
- Extensive views integration makes even the shopping cart easily configurable

Drupal Commerce Conventions

- Product
  - Represents the item being sold
  - Entity not directly visible to customer
- Product Display
  - Presentation of items for sale

- Entities
  - Product
  - Line Item
  - Order
  - Customer
  - Payment

Commerce Object Relations



DC - Entities and Fields

- Drupal Commerce creates five entity types:
  - Product
  - Line item
  - Order
  - Customer profile
  - Payment transaction

- Drupal Commerce creates four field types (fields attach to entities):
  - Price
  - Product reference
  - Line item reference
  - Customer profile reference

- For more info on fields : http://drupal.org/documentation/modules/field

DC - Products and Product Display

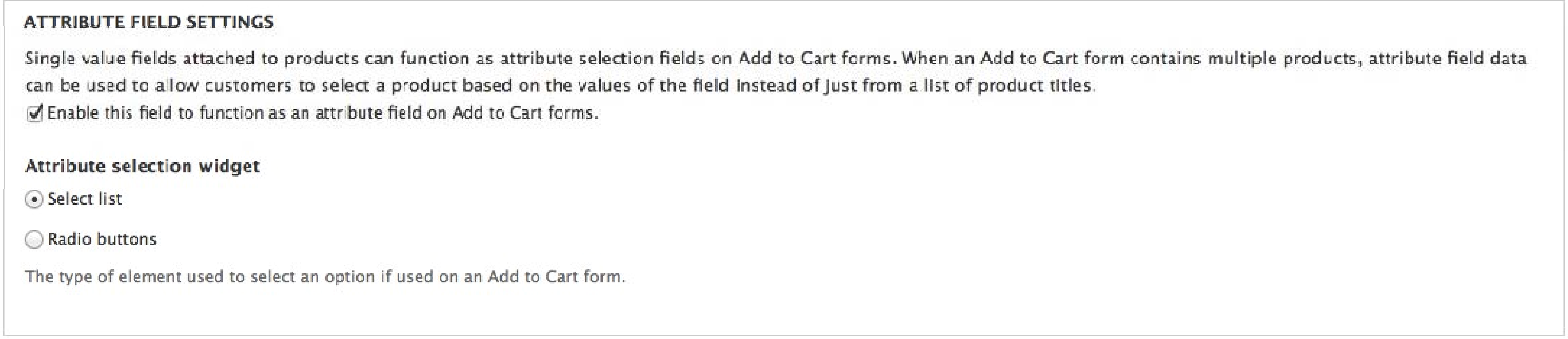
- A product can be one of the following:
  - A single item for sale on the site
  - A variation in a group of items for sale on the site (e.g. t-shirt size)
  - A non-tangible product reserved through the site (e.g. event registration)
  - An item that is not purchased, but represents a payment (e.g. donation)

- A product can be displayed on a single page, or on pages grouping multiple products together, or on multiple different pages or Views, or not at all.

- Drupal Commerce enforces a separation between the definition of a product, and the product display.

Attributes

- Product references can be multi-value fields
- Attributes are simple fields with one additional option
  - Only single value list items can become attributes



ATTRIBUTE FIELD SETTINGS

Single value fields attached to products can function as attribute selection fields on Add to Cart forms. When an Add to Cart form contains multiple products, attribute field data can be used to allow customers to select a product based on the values of the field instead of just from a list of product titles.

☑ Enable this field to function as an attribute field on Add to Cart forms.

Attribute selection widget

● Select list

○ Radio buttons

The type of element used to select an option if used on an Add to Cart form.

DC - Shopping Cart

- Shopping carts in Drupal Commerce are just orders with particular statuses (e.g. checkout, review, payment, completed, canceled, etc.).
- As soon as a product is added to the cart, an order is created and associated with the user (if logged in) or in the session (if not logged in).
- Each change to a shopping cart order is saved as a revision.
- Shopping cart orders do not automatically time out (there is work to make this configurable via rules), so they remain until checked out or manually removed by the admin. Old cart orders may contain valuable information that can be retrieved before they are deleted.

DC - Price

- The unit price field includes the following:
  - Amount
  - Currency code (three character ISO-4217 currency code of the price)
  - Data (additional data stored in the price, for example, components)

- Core price component types:
  - Base price – price prior to calculation
  - Discount – price component type for price reductions
  - Fee – price component type for price increases
  - Tax rates – each tax rate gets its own component type so the total tax collected for an order can be found in its order total price field

Pricing Rules

- Price information of products is probably one of the most dynamic data and highly variable:

→ On the fly calculation of pricing for:

  - Taxes
  - Discounts
  - Any price variations

Everything is controlled through Rules (payment, shipping, pricing, tax rates) so let’s pause a bit and take a closer look at Rules.

Let’s talk Rules for a moment

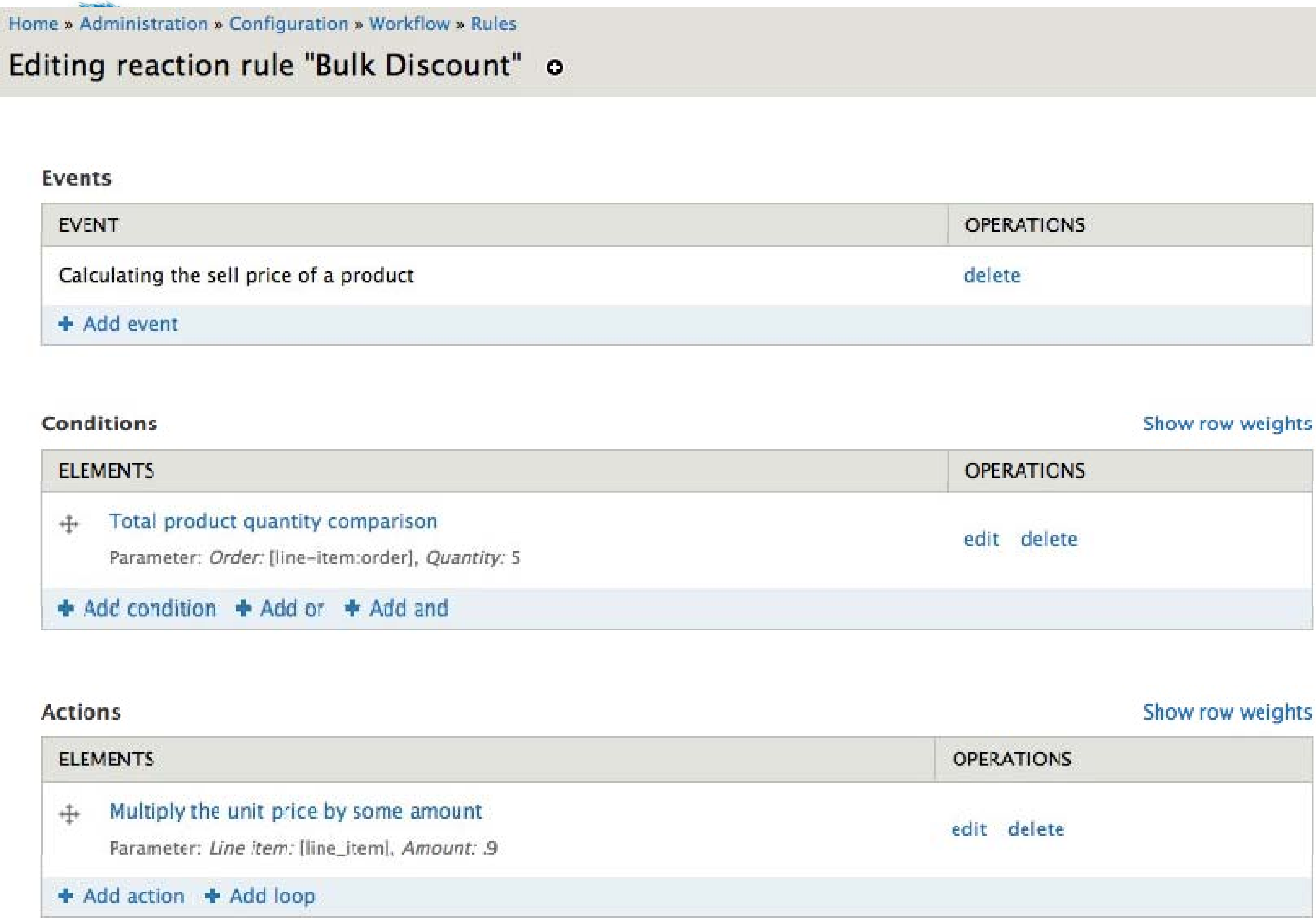
- Rules is a great module/API written by Wolfgang Ziegler (@the_real_fago) that lets you define conditionally executed actions based on occurring events. In other words, Rules lets you take specific actions based on events that occur on your site. For Rules paid development and consulting services: http://wolfgangziegler.net/contact

- More videos on Rules can be found on the NodeOne site: http://dev.nodeone.se/en/learn-the-rules-framework

- This presentation will focus on Rules used in Drupal Commerce, so if you want to learn more about this subject, contact the @CommerceGuys or checkout their channel: http://vimeo.com/channels/commerceguys

What are Drupal Rules?

- Drupal Rules are a mechanism for adding business logic to your site.
- Drupal Rules lets you define actions that your site can take when a variety of events occur.
- Drupal Rules lets you add one or more conditions when an event occurs and only perform an action when an event occurs and certain conditions apply.
- For example, an email can be sent (**action**) when a specific user role (**condition**) posts content (**event**)
- A basic rule is made up of an event and an action, but a condition is optional



Editing reaction rule "Bulk Discount"

Events

EVENT	OPERATIONS
Calling the sell price of a product	delete
✚ Add event	

Conditions

ELEMENTS	OPERATIONS
Total product quantity comparison Parameter: *Order:* [line-item:order], *Quantity:* 5	edit delete
✚ Add condition ✚ Add or ✚ Add and	

Actions

ELEMENTS	OPERATIONS
Multiply the unit price by some amount Parameter: *Line item:* [line_item], *Amount:* .9	edit delete
✚ Add action ✚ Add loop	

Drupal Rules Components

In Short:

(if you only need to remember something about it)

- A Rule has three parts:

- Event – thing listening for/subscribed to	**When something happens**
- Condition – should I keep going?	**If a certain condition is met**
- Action – what we will do	**Do something**

Product Sell Price Rules

- Product sell prices are determined via a Rules based calculation process.
  - A new line item is created representing the product
  - The unit price of the line item is initialized to the base price
  - The line item is then passed through the Rules via the *Calculating the sell price of a product* event where its unit price may be manipulated as necessary
  - The final unit price of the line item becomes the sell price of the product displayed

- Sell price calculations can include discounts, taxes, currency conversion, etc.

Coupons and Discounts

Commerce Coupon:

http://drupal.org/project/commerce_coupon

Fixed price, Percentage price, Various conditions or limits, etc…

Commerce Discounts:

(This is greatly related with the Pricing and pricing rules.)

http://drupal.org/project/commerce_discount

This is really only a user interface to create Rules. But in fact everything could already be controlled through custom Rules. For example, role based pricing discounts can be setup by adding a custom pricing rule.

Commerce Shipping API and Rules

Commerce Shipping

http://drupal.org/project/commerce_shipping

Integrated with many shipping extensions:

http://drupal.org/project/commerce_ups

http://drupal.org/project/commerce_flat_rate

Commerce Shipping provides a global API framework allowing any developer to extend the shipping logic with any specific Requirements, for example in China:

- Shunfeng price calculation
- Yunda price calculation
- Etc…

More extensions at:

http://www.drupalcommerce.org/extensions/module/shipping

Checkout and Payment

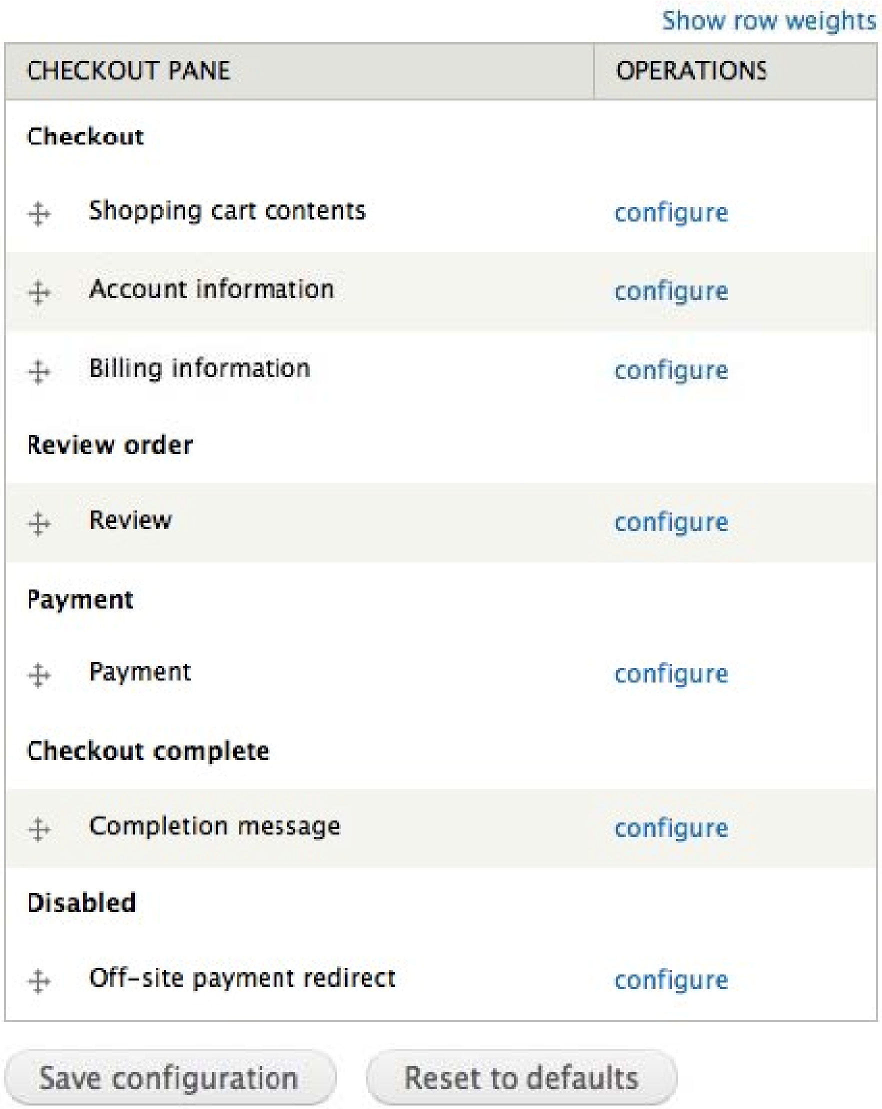
- The checkout form is made up of checkout pages called checkout panes (i.e. container divs).

- The default progression through the checkout form goes from one non-empty checkout page to the next until the completion page is reached. The default behavior can be modified via the drag and drop checkout form builder.

- As the customer progresses through checkout, their cart order is continually being updated (the checkout form acts as a multi-step order edit form with the end result being a fully formed order ready for fulfillment.

- The payment page is meant to be used for the off-site payment redirect checkout pane. It is purely a point of departure and return for off-site payment methods (e.g. PayPal) and should not be used to hold any other checkout panes.

Checkout Process

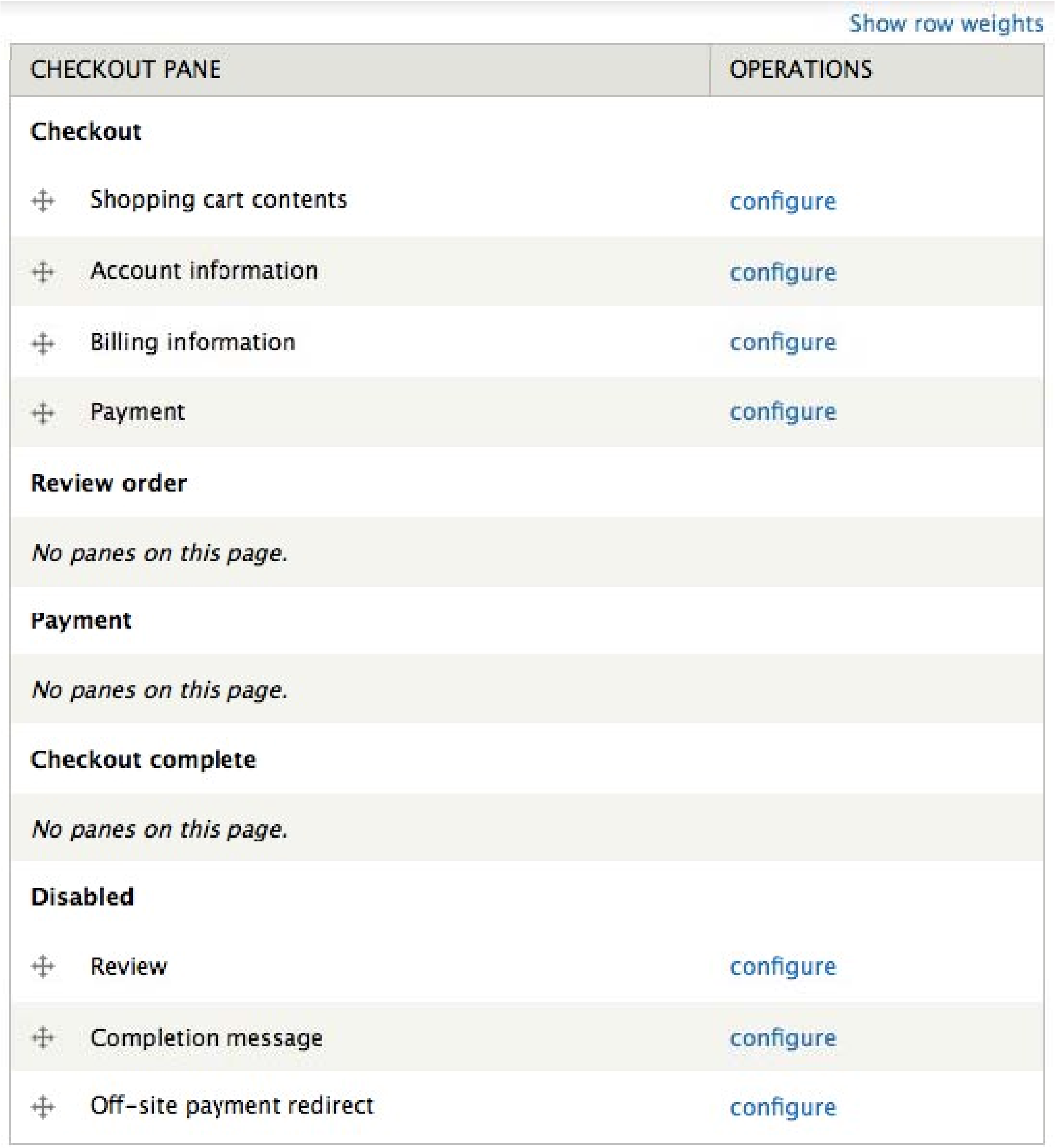


CHECKOUT PANE	OPERATIONS
Checkout	
Shopping cart contents	configure
Account information	configure
Billing information	configure
Review order	
Review	configure
Payment	
Payment	configure
Checkout complete	
Completion message	configure
Disabled	
Off-site payment redirect	configure

- Four basic steps in the checkout process
- Steps are filled with panes
- Panes can can be drag and drop sorted and moved from one step to another
- Carefully consider the order that you collect information
- Payment should usually be the last functional pane

One Page Checkout?

- A single page checkout model is possible with Commerce
- You still might want a review pane or thank you message



Show row weights

CHECKOUT PANE	OPERATIONS
Checkout	
✣ Shopping cart contents	configure
✣ Account information	configure
✣ Billing information	configure
✣ Payment	configure
Review order	
No panes on this page.	
Payment	
No panes on this page.	
Checkout complete	
No panes on this page.	
Disabled	
✣ Review	configure
✣ Completion message	configure
✣ Off-site payment redirect	configure

Commerce Payment API and Rules

Commerce Payment API framework allows any developer to extend the Payment logic with any specific Third Party API requirements, for example:

- Alipay payments: http://drupal.org/project/commerce_alipay
- Paypal: http://drupal.org/project/commerce_paypal
- Etc…

More extensions at:

http://www.drupalcommerce.org/extensions/module/payment

Advanced Topics

Advanced API coding and integration.

ERP: Some work is undergoing with Erpal integration

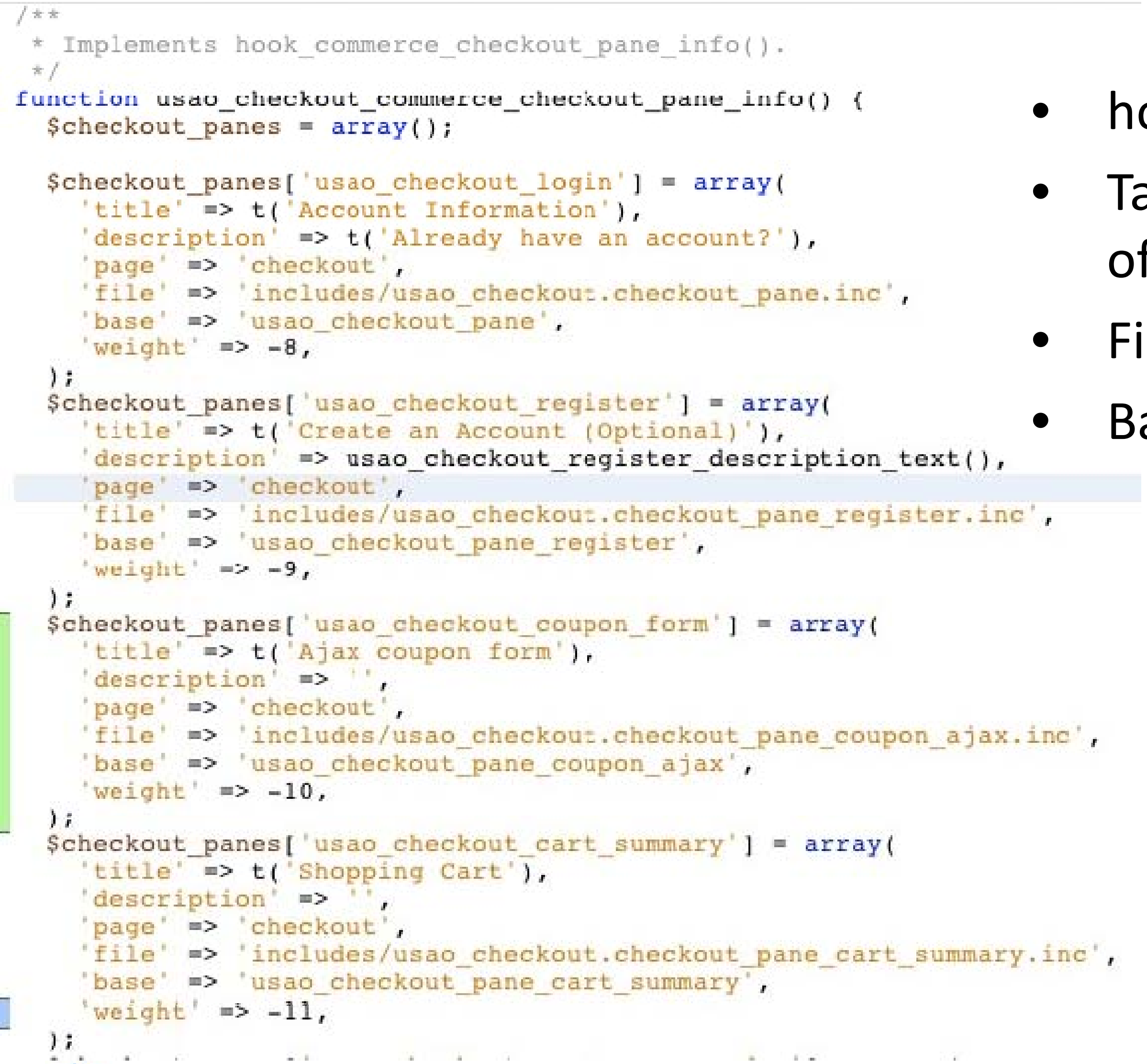
Social Commerce: Give commissions to sellers

Merchants/Buyers platform

Commerce API

- Working with entities and metadata wrappers
- Commerce panes
- Fields and formatters
- Payment API
- Shipping API
- The rest is provided by Rules, Views and Entity APIs

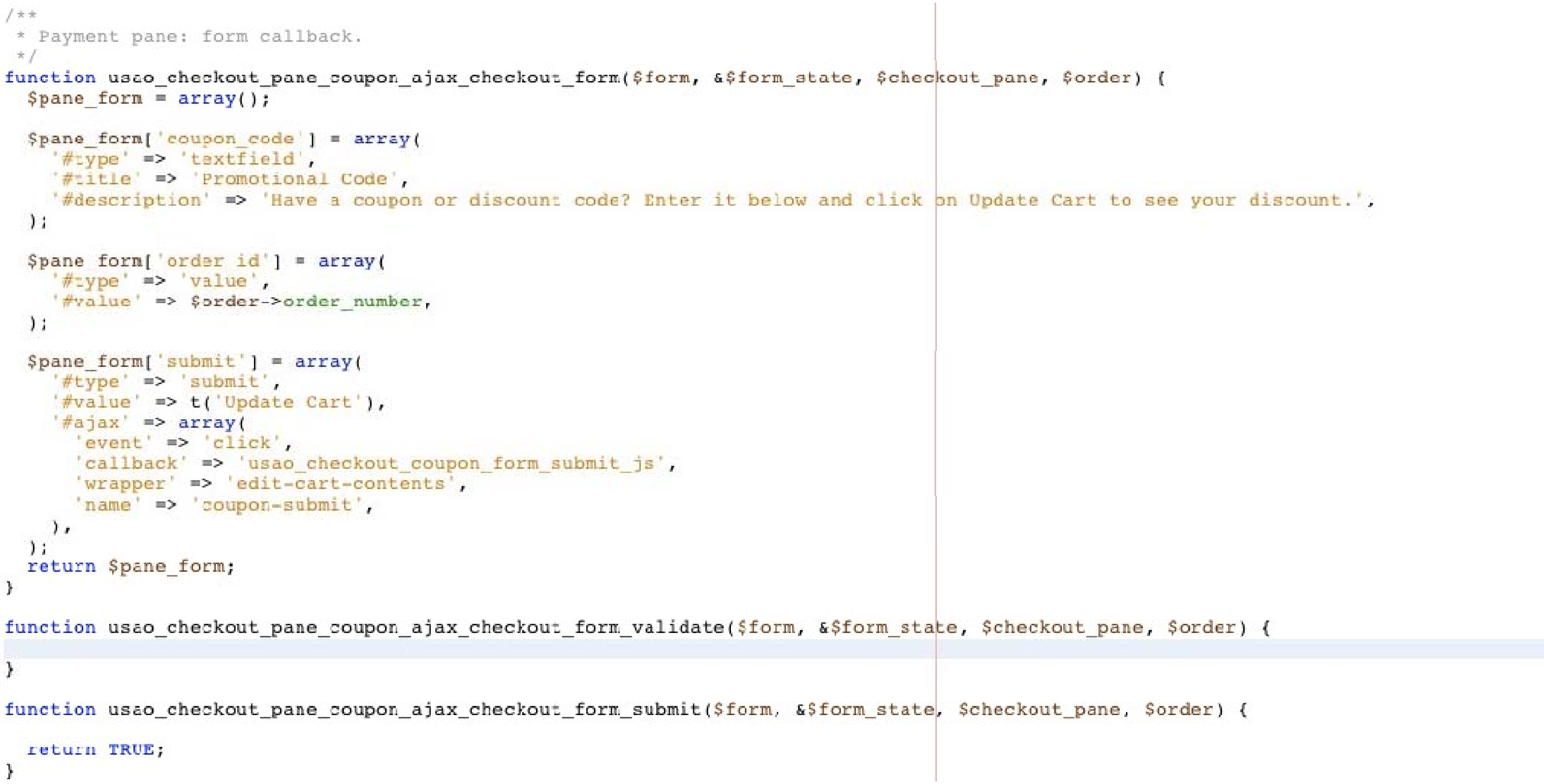
Commerce API Coding Example



```
/**
 * Implements hook_commerce_checkout_pane_info().
 */
function usao_checkout_commerce_checkout_pane_info() {
  $checkout_panes = array();

  $checkout_panes['usao_checkout_login'] = array(
    'title' => t('Account Information'),
    'description' => t('Already have an account?'),
    'page' => 'checkout',
    'file' => 'includes/usao_checkout.checkout_pane.inc',
    'base' => 'usao_checkout_pane',
    'weight' => -8,
  );
  $checkout_panes['usao_checkout_register'] = array(
    'title' => t('Create an Account (Optional)'),
    'description' => usao_checkout_register_description_text(),
    'page' => 'checkout',
    'file' => 'includes/usao_checkout.checkout_pane_register.inc',
    'base' => 'usao_checkout_pane_register',
    'weight' => -9,
  );
  $checkout_panes['usao_checkout_coupon_form'] = array(
    'title' => t('Ajax coupon form'),
    'description' => '',
    'page' => 'checkout',
    'file' => 'includes/usao_checkout.checkout_pane_coupon_ajax.inc',
    'base' => 'usao_checkout_pane_coupon_ajax',
    'weight' => -10,
  );
  $checkout_panes['usao_checkout_cart_summary'] = array(
    'title' => t('Shopping Cart'),
    'description' => '',
    'page' => 'checkout',
    'file' => 'includes/usao_checkout.checkout_pane_cart_summary.inc',
    'base' => 'usao_checkout_pane_cart_summary',
    'weight' => -11,
  );
```

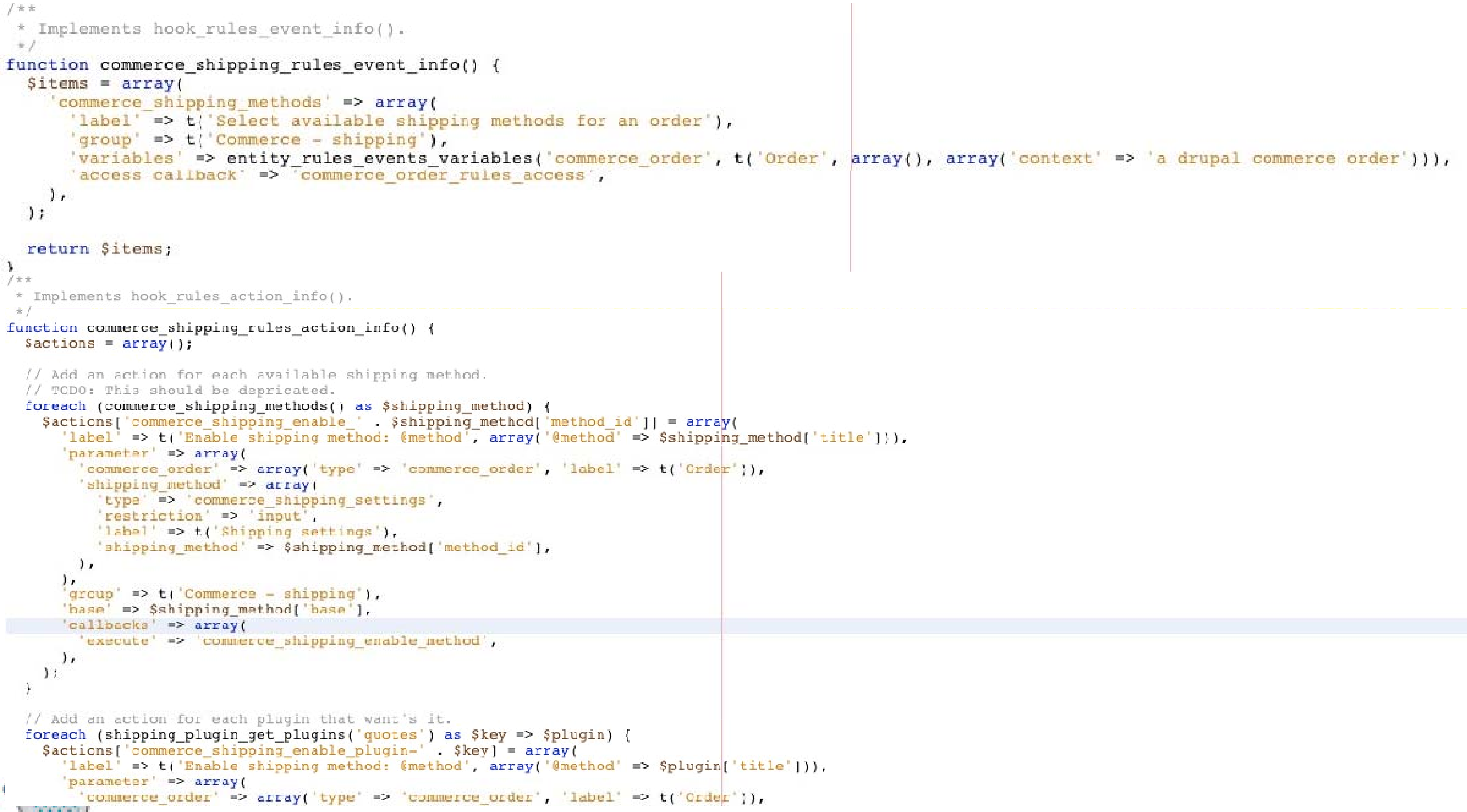
- hook_commerce_checkout_pane_info
- Takes no parameters, returns and array of checkout panes
- File points to an include file
- Base is the base function name

DC API Coding Example: Coupon pane



```php
/**
 * Payment pane: form callback.
 */
function usao_checkout_pane_coupon_ajax_checkout_form($form, &$form_state, $checkout_pane, $order) {
  $pane_form = array();

  $pane_form['coupon_code'] = array(
    '#type' => 'textfield',
    '#title' => 'Promotional Code',
    '#description' => 'Have a coupon or discount code? Enter it below and click on Update Cart to see your discount.',
  );

  $pane_form['order_id'] = array(
    '#type' => 'value',
    '#value' => $order->order_number,
  );

  $pane_form['submit'] = array(
    '#type' => 'submit',
    '#value' => t('Update Cart'),
    '#ajax' => array(
      'event' => 'click',
      'callback' => 'usao_checkout_coupon_form_submit_js',
      'wrapper' => 'edit-cart-contents',
      'name' => 'coupon-submit',
    ),
  );
  return $pane_form;
}

function usao_checkout_pane_coupon_ajax_checkout_form_validate($form, &$form_state, $checkout_pane, $order) {

}

function usao_checkout_pane_coupon_ajax_checkout_form_submit($form, &$form_state, $checkout_pane, $order) {

  return TRUE;
}
```

DC API Coding Example: shipping



```php
/**
 * Implements hook_rules_event_info().
 */
function commerce_shipping_rules_event_info() {
  $items = array(
    'commerce_shipping_methods' => array(
      'label' => t('Select available shipping methods for an order'),
      'group' => t('Commerce - shipping'),
      'variables' => entity_rules_events_variables('commerce_order', t('Order', array(), array('context' => 'a drupal commerce order'))),
      'access callback' => 'commerce_order_rules_access',
    ),
  );

  return $items;
}
/**
 * Implements hook_rules_action_info().
 */
function commerce_shipping_rules_action_info() {
  $actions = array();

  // Add an action for each available shipping method.
  // TODO: This should be depricated.
  foreach (commerce_shipping_methods() as $shipping_method) {
    $actions['commerce_shipping_enable_' . $shipping_method['method_id']] = array(
      'label' => t('Enable shipping method: @method', array('@method' => $shipping_method['title'])),
      'parameter' => array(
        'commerce_order' => array('type' => 'commerce_order', 'label' => t('Order')),
        'shipping_method' => array(
          'type' => 'commerce_shipping_settings',
          'restriction' => 'input',
          'label' => t('Shipping settings'),
          'shipping_method' => $shipping_method['method_id'],
        ),
      ),
      'group' => t('Commerce - shipping'),
      'base' => $shipping_method['base'],
      'callbacks' => array(
        'execute' => 'commerce_shipping_enable_method',
      ),
    );
  }

  // Add an action for each plugin that want's it.
  foreach (shipping_plugin_get_plugins('quotes') as $key => $plugin) {
    $actions['commerce_shipping_enable_plugin-' . $key] = array(
      'label' => t('Enable shipping method: @method', array('@method' => $plugin['title'])),
      'parameter' => array(
        'commerce_order' => array('type' => 'commerce_order', 'label' => t('Order')),
```

Drupal Commerce in China

- How does it compare with local solution
  - ShopEx
  - EcShop

- Actually most of people still work with Third party hosting e-com platform, such as:
  - Taobao
  - T-Mall

Strong need for integration in China: needs work.

Roadmap and what’s coming up

Let’s conquer the world!

Extensions of the Commerce module base for:

- Shipping
- Payment
- More sales extensions, CRMs

Resources

- http://www.drupalcommerce.org/extensions
- http://www.drupalcommerce.org
- http://www.drupal.org/project/commerce
- http://www.drupal.org/project/commerce_kickstart
- http://www.commerceguys.com/resources

Want More?

Join our community developments:

Pin down in your Calendar!

(Don't miss out our next dates, we have lots to do for Drupal Commerce in China!)

Trainings, **Custom Developments**, **Bug fixing**, **Patching**, **Maintenance**, **Testing**, and much more at:

- 上海Drupal协同贡献聚会 - 2013年4月20日 / Drupal contribute workshop SH - April 20, 2013: http://groups.drupal.org/node/289713
- 上海Drupal协同贡献聚会 - 2013年5月18日 / Drupal contribute workshop SH - May 18, 2013: http://groups.drupal.org/node/289733

Questions

Thank you for your attention.

More questions or comments, feel free to get in touch:

- DYdave on drupal.org or IRC #drupal-china, #drupal-commerce
- david.suissa@davyin.com