

Bahmni Developed Modules Analysis

1.bahmni_atom_feed : Depends on base, product, sale, account

Class	Change done in bahmni module Bahmni Developed Modules Analysis
product.category	Added new column uuid, and set this value for row by overriding create method.
product.uom	-//-
product.uom.category	-//-
order.type.sale.shop	<p>This is a new class defined in module, but this class consists of relation with sale.shop object. But sale.shop class doesn't exist in any version after 7. We have to find some other way for this mapping. This class is used to map bahmni user location and openerp shop.</p>
sale.order	<p>New fields : external_id, group_id, group_description group_id: many2one field with visit object. Visit is a new object introduced in this module. But nowhere value for that field is set. group_description: just a character field Remove group_id and group_description field in new implementation and hence from method where it is referred.(in create_invoice), I don't see any significance of these 2 fields.</p>
sale.order.line	<p>New fields added are: external_id, external_order_id external_id field is used for storing encounter_id and external_order_id is used to store order id returned by bahmni .</p>
syncable.units	<p>New object introduced in module, but this is not getting used in bahmni for now. Skipping this class definition in new implementation, as it's not getting used anywhere</p>

Bahmni Developed Modules Analysis

reference.data.service	This is the class used to create service products which are listed below e.g. radiology test , lab test, lab panel. hence method for creating products is written in this class only.
radiology.test.service	New object introduced for mapping radiology tests. But there is no view defined for it, and also the method getting called when create.radiology.test category is passed to process_event method, that method is not defined in that class.but in the parent class i.e. reference.data.service Only methods for getting category and category hierarchy is defined in each child class.
lab.test.service	Same issue as that of above class, no method is defined, which is getting called. Method is defined in parent class i.e. reference.data.service Only methods for getting category and category hierarchy is defined in each child class.
lab.panel.service	Method is defined in parent class i.e. reference.data.service Only methods for getting category and category hierarchy is defined in each child class.
drug.service	This is new object introduced in this module to create products which are newly created in openmrs. And also to sync drug categories. In this class, there are methods defined like get_object_by_uuid and get_object_by_domain. As in this way there are 2 method calls are done, one within the class and other to orm method. So, it's better to directly call orm method.
Above 5 classes are for creating products only, just categories for products are different. and Drug service has sub-category, hence it has a separate class than other 4. So, combined functionality, by handling category using context or one additional parameter can be implemented.	
product.uom.service	New object defined for syncing newly created unit of measures in openmrs.

Bahmni Developed Modules Analysis

<p>processed.drug.order</p>	<p>New object introduced to track the status of orders, whether it is dispensed or not, but each time new record for this object is getting created whenever sale.order is created or updated. While updating sale_order, record related to that order should get updated instead of creating new one.</p> <p>QUERY : Once medicine is prescribed from bahmni, is that prescription editable on bahmni side again? If not then there is no need to write anything on sale_order updation and each time new order has to be generated. If yes, want to know the scenario for this event.</p> <p>=====</p> <p>When medicine is prescribed from bahmni, new drug order is created for pharmacy location and when delivery order is processed that means drugs are dispensed. When drugs are used in dispensary or hospital itself e.g. in I.C.U. some medicines are given to patient, then at that time, drug orders has to be created for that specific location only. And it should get completely processed at the time of creation.</p> <p>=====</p> <p>skipping processed.drug.order class definition, as this class only defined for storing status of drug orders, which we can track by just adding a field in order, why to define different class?</p>
<p>atom.event.worker</p>	<p>New object introduced, which contains the method for syncing data which comes from bahmni. process_event method from this class is only get called from bahmni.</p>
<p>atom.feed.marker</p>	<p>This object is designed to keep the track of events occurred on openerp side from bahmni side. In case of failure of event, reprocess failed events.</p>
<p>event.publisher</p>	<p>This is new object introduced in openerp, and this is used to publish events of any action related to product.product is taken place i.e. create, write, unlink of product. But for sale order, event is only published on creation of invoice, can't get it, why only on invoice creation.</p> <p>Right now, there is no reverse sync of openerp to any other system.</p> <p>Skipping event.publisher class definition, as this class is just an intermediate class to create event records, so we can directly create event records.</p>

Bahmni Developed Modules Analysis

event.records	New object introduced , which stores the events created by event publisher.
---------------	---

2. bahmni_customer_payment : base, account, account_voucher

account.voucher	<p>New fields added ;</p> <p>Balance_before_pay: this field is added to account_voucher to know the due amount for invoice or customer/supplier</p> <p>Total_balance: amount that would reflect the amount which would be pending after paying mentioned amount.</p> <p>Create_uid: why this field is added, when this field already exists in each and every table.</p> <p>Invoice_id: searched invoice from credit lines of voucher.</p> <p>Bill_amount: field to get the invoice total amount. But in invoice_id, only single invoice is getting considered and bill_amount also reflect the amount of single invoice.</p> <p>Def _def_journal() : This method is overridden to overwrite a condition. There is a condition for checking ttype value, and if that value matches to payment or receipt then journal type should be bank ; but removed this condition and made ttype to cash.</p>
account.invoice	<p>In method, def invoice_pay_customer is over-riden. In this method, instead of returning view_vendor_receipt_dialog_form view, view_vendor_receipt_form is returned. And instead of returning it with target new, it is returned as target current, which misdirect the invoice flow. So, if user makes a payment for an invoice</p> <p>In new implementation, target will be new only, as target current misleads the flow.</p>
res.partner	<p>def name_search method overridden to just modify the method. There no functional change, just a code logic change. Args are converted to query string and concatenated with query_string which is getting executed. This reduced a search method call, after query execution.</p> <p>-----</p> <p>Def name_get method is overridden to add Customer reference in string returned by name_get method.</p> <p>-----</p> <p>Def name_create method is overridden, but there is no change in name_create method which is overridden.</p> <p>-----</p> <p>Unique constraint added on ref field of partner.</p>

Bahmni Developed Modules Analysis

3. bahmni_dhis2_export : depends on product, stock, base

- Can't get why stock module's dependency is added in this module, nothing has been referred from stock module in this module.
- DHIS is a reporting system for visualization of district health information.
- In this module, new field dhis_code2 is added to res_company and product_product class.
- An action is added to company record to export the created csv.
- **In new implementation going to add this report to bahmni_atom_feed only.**

4. bahmni_internal_stock_move : depends on base, stock

- This module is dependant on base and stock module only, but sale_price field added in stock_production_lot class is getting referred in this module, in a sql report.
- And that field is added to stock_production_lot class in bahmni_stock_batch_sale_price module, hence dependency of bahmni_stock_batch_sale_price is must.
- Since, In bahmni_stock_batch_sale_price there is already dependency of base and stock, we don't need to add their dependency again.

stock.location	<p>Method name_get is overridden, but there is no change done in name display, as per my analysis name_get method was overridden to show short name for a location, but there doesn't exists field in stock.location, where short name can be entered.</p> <p>In new implementation skipping this method overriding.</p>
stock.move	<p>Method onchange_lot_id is overridden to add warning message when selected serial number/lot is already expired.</p> <p>-----</p> <p>New method defined _get_stock_for_location, for getting available product quantities at selection location.</p> <p>-----</p> <p>New field added : stock_availability Onchange_product_id method is overridden to set qty , which is obtained from _get_stock_for_location method call.</p> <p>-----</p> <p>New field added : stock_picking_time This is a functional field, whose value is set from picking's date value.</p> <p>-----</p> <p>Method overloading is done for onchange_quantity to set the available qty. and in method overloading move_lines has been also passed as argument, and returned to set value of move_lines; but returning value on onchange of child field, won't set value in parent field, So, I don't think this logic works.</p> <p>Skipped this method overriding in new implementation, as we can not write onchange method on field from one2many relation to change value in parent class.</p>

Bahmni Developed Modules Analysis

stock.move.split	New field added : stock_move Whose value is set as active_id, through overriding default_get method. Rest other values were set in default stock module only.
stock.picking	New onchange method has been added on location fields. So that the locations which are selected in the parent form, will get reflected in the child lines.
prod_last_moved.report	This is a sql report developed to get the view for getting list of products with their respective last moved date. Last moved date is date of movement of product from one location to another.
prodlots.report	Sql report of products to track the serial number by location. This report shows the serial number and it's current location and qty at that location.

5. bahmni_lab_seed_setup : depends on base, product, stock, bahmni_pharmacy_product

- This module consists of data files for creating Accounts, Product Category, Suppliers, Unit of measure, stock location. - **jss setup data**
- Also, deleted default service product, and to delete default product, uuid is added to service product record.
- Date format has been changed for english language. - **jss setup data**

6. Bahmni_logger :

- I have not understood the purpose of this module.
- This module gets installed by its own, as auto-install is set true for this module.
- **This was log handles specifically designed to jss.**

7. bahmni_module_install : depends on all modules.

- This is a module developed, to install all the bahmni modules in one go.

8. bahmni_pharmacy_product : depends on base, product, stock, bahmni_atom_feed, purchase

- In Access rights, access rights are overridden, but no changes done in their permissions. Only from stock_user's access rights, write permission has been removed by overriding it for product_product and product_template both. - **But as per openerp standard, if user has create access, then write access has to be given. - skipping this in new implementation.**
- In default product_template class, there is validation when purchase unit of measure category and unit of measure category is not same. This validation is removed by bahmni and allowed to change that.- **will keep the same validation, as user will get confused when default unit of measure is changed unknowingly, it's better user will select default UOM first and then purchase UOM.**
- When purchase unit of measure is changed, and if stock move for that product is not yet

Bahmni Developed Modules Analysis

present in the system, then changing is allowed though that unit of measure does not belong to same category. And it's default unit of measure is replaced with selected purchase unit of measure.

product.product	<p>get_product_available method is overridden to list products which are not expired.</p> <p>New fields added : uuid, drug, manufacturer all 3 are character fields only.</p> <p>mrp field for allowing user to set mrp of a product.</p> <p>low_stock: boolean field, but a functional field, and is set when available quantities for a product are below minimum qty level mentioned in reordering level.</p> <p>Actual_stock : functional field , for computing actual stock of product in warehouse, if location provided then for that specific location.</p> <p>-----</p> <p>Unlink method is overridden to raise event for event publisher.</p> <p>-----</p> <p>Create and write method is also overridden to publish event.</p> <p>-----</p> <p>New methods defined get_mrp and set_mrp, but those methods are used in bahmni_purchase_extension module. - moved to custom_product module.</p> <p>-----</p> <p>Procure_method is by default set to make_to_stock.</p>
-----------------	---

9. bahmni_print_bill : depends on base, stock, bahmni_sale_discount, bahmni_customer_payment

- New widget print-bill is introduced in this module, to directly give print command to printer. This widget is used on button which is shown on customer payment form, when user validate payment.
- One controller is written to fetch the bill details to qweb template, called in widget.
- Why created widget, can't it be just done through normal method call from jquery? As that widget is not useful anywhere else, since that method is specific to bill print only.
- One new field initials of character type added to partner.
- There is method written for converting number to marathi, but not used in report.

10. bahmni_purchase_extension : depends on base, purchase, bahmni_stock_batch_sale_price, product

- Purchase Order report template is overridden to add new column in the report. Purchase Order report is inherited to add manufacturer column in that.
- There is one css implemented, but I don't see that, it is working.

product.supplierinfo	<p>New character field added named manufacturer to save manufacturer name.</p> <p>Why do we need to add manufacturer ?</p>
----------------------	--

Bahmni Developed Modules Analysis

	<p>-----</p> <p>Unit_price field is added as functional field. It's value is computed based on pricelist.partnerinfo lines in supplierinfo. This field is visible only when user has access to purchase pricelists. But unit price is just taken from the line which has higher sequence, should that be based on quantities mentioned in supplierinfo? While that product is added to purchase quotation, at that time too price and quantity is taken from pricelist.partnerinfo line, and not from partner.supplierinfo.</p>
pricelist.partnerinfo	<p>New float type fields added : mrp and unit_price, From this unit_price only unit_price of product.supplierinfo is calculated. In this module, price field is renamed as mrp and new field has been created for unit_price. Since, this class does not exists anymore in Odoo, eliminating this customization. And adding mrp field to product.supplierinfo directly instead of pricelist.partnerinfo class.</p>
purchase.order	<p>Default method is defined for setting default warehouse_id, but that method is not at all getting used in bahmni, as till now bahmni has not done single warehouse implementation.</p>
purchase.order.line	<p>Onchange_product_id method is overridden in which code logic for setting up unit_price, mrp and manufacturer has been added.</p> <p>-----</p> <p>Onchange_qty and onchange_product_uom methods are newly added, which has suppressed the functionality of onchange_product_id method, which was called on product_uom and qty.</p> <p>-----</p> <p>When qty is changed, in onchange_product_id, there is warning raised, when user sets quantity less than min_qty set in the supplierinfo. - This warning no more exists in Odoo, so need to suppress.</p> <p>-----</p> <p>New fields added: Manufacturer : char type field, which stores the name of manufacturer, which is set in the supplierinfo</p> <p>-----</p> <p>Mrp: mrp price is set through product.supplierinfo's get_price method, in which price is set from pricelist.partnerinfo field, and if it is not set, then price set through pricelist set in the supplier. And if there is no record for supplierinfo, then mrp is set from product</p>

Bahmni Developed Modules Analysis

	<p>master. ----- Product_category : this is char field, which is functional field, and using this product_category name is set in the column. But this field is not stored in the database neither this field is getting used anywhere, not even displayed in the view ; So, didn't get the significance of this field.</p>
stock.picking.in	<p>New field added : Warned, boolean field , and it's default value is set to false.</p>
stock.partial.picking	<p>Def do_partial is inherited to add validation when picking_type is not internal on these two conditions : When duplicate serial number is entered or no serial number is entered. ----- In inherited do_partial method , picking_id.warned is used.. Warned field is just added to incoming_shipment, While do_partial method is same for all the shipments, hence this can lead to error while confirming shipment. In new implementation warned field will be added to stock.picking instead of stock.picking.in</p>
stock.move.split.lines	<ul style="list-style-type: none"> • Default method is set for sale_price field. • Onchange of cost_price new method is defined to set sale_price. • For calculating sale_price markup_percentage table is hard coded in this method, which can be made configurable. • Onchange_cost_price method is defined here, but in view this method is added to another module : bahmni_stock_batch_sale_price

11. bahmni_sale_discount : depends on base, sale, account

- New widget created for link-prescription, for adding a new button to redirect to bahmni app.
Try : instead of widget, if normal button can be used in js functionality, Or is there any other way to redirect to url.
- Sale Order report is overridden to add provider name column in the report.
- Account_print_invoice and account_report.xml files are not included in __init__.py and __openerp__.py resp.
- New class rounding.off is introduced in this module to round off the amount.
- There is many2one field with visit, and there is no dependency of bahmni_atom_feed, as visit class is defined in bahmni_atom_feed. - **skipping visit class definition in new implementation**

Account.move	Name_search and name_get methods are overridden, but there is no change in the methods, so no need to override it.
--------------	--

Bahmni Developed Modules Analysis

	<p>All the methods, fields of account.move are copy-pasted in this module, can't understand why it is. There is only single line change in validate method only, rest all things are identical to original. In new implementation only def validate has to be overridden.</p>
Account.invoice	<p>New fields added :</p> <ul style="list-style-type: none"> ● discount : float field for fixed percentage. ● discount_per : float field to define discount percentage. ● discount_method: field to select type of discount, fixed/percentage ● def compute_invoice_totals overridden to define logic for invoices of type out_refund. ● def action_move_create is overridden to debit invoice discount against respective account head. And for that new method is defined i.e. def _update_discount_head.
sale.config.settings	<p>New fields added :</p> <ul style="list-style-type: none"> ● group_final_so_charge : one new field added to sale order, which is changeable for user, this amount will be considered as final chargeable amount. ● group_default_quantity : added this field to allow setting product qty to -1. Field is defined in this module, but used in bahmni_atom_feed module. And only allowed to add -1 qty when sale order created through medication. ● round_off_by : this is the field used for rounding off amount to specified value. This method has to be generic as, this method is used.
Res.partner	<p>New character field defined for village. Villages can be configurable as this can be data which will get repeated.</p>
account.account	<p>Accounts for discount of type expense are created through xml here.</p>
rounding.off	<p>New class defined for rounding off value to the number defined in configuration. In new implementation, this class will get defined without table creation, as it does not require any record to save.</p>
sale.order	<ul style="list-style-type: none"> ● Create Invoice button's type is changed from action to workflow. ● directly workflow signal action_invoice_create is called on button. ● Other Information page is made invisible. - In new implementation this page will be visible. ● New fields added : <ul style="list-style-type: none"> ○ date_order : this field already exists in default module, no need to override this.

Bahmni Developed Modules Analysis

	<ul style="list-style-type: none"> ○ datetime_order : new datetime field, for default value set was string, which is giving error, so have solved this issue using datetime conversion. <ul style="list-style-type: none"> ■ Overridden create and write method to set value for date_order, as date_order has been made invisible in the form. ○ discount_percentage : float field for entering discount percentage depending on type of discount. ○ Chargeable_amount : if user is allowed to do so, this field will be editable to user for setting overall chargeable amount. ○ Default_quantity : field visible to user, who is allowed to set default quantity to -1. ○ discount_amount: if type of discount is fixed, then entering fixed discount amount in this field. ○ round_off: round_off amount calculated based on round off by value set in the configuration. ○ discount_acc_id: Discount Account head. ○ prev_outstanding_balance, total_outstanding : functional fields to get previous balances for partner. ○ partner_village, partner_uuid: functional fields of char type calculated through def partner_details. ○ Care_settings : selection type field to select type whether IPD/OPD. ○ provider_name: char field; can't we make this as many2one field? ○ All functional fields are overridden as newly added fields were supposed to be considered while calculation of total amount calculation of order, hence field and method both are overridden. ● def_prepare_invoice method is overridden to add values for new fields added in account.invoice.
sale.order.line.make_invoice	def make_invoices overridden to set the values for newly added fields in invoice.
sale.order.line	<ul style="list-style-type: none"> ● order_uuid: new char field for just generating a random unique id, using uuid. Field is added in this module, while getting used in bahmni_atom_feed, and it don't have any dependency of bahmni_sale_discount. ● All the methods which are overridden and newly defined are need in bahmni_atom_feed functionality only, no use of those methods in bahmni_sale_discount module.
account.receivables	New class defined , which is getting used nowhere. Skip this class definition in new implementation.

Bahmni Developed Modules Analysis

12. bahmni_stock_batch_sale_price : depends on base, stock, sale, sale_stock, product_expiry

account.invoice.line	New fields added : Batch_name and expiry_date
batch.stock.future.forecast	This a new object introduced in this module, but this a sql record, and hence records of this object are auto-created through sql view query.
product.product	Name_search method is overridden to find a drug, whose name starts with entered character/ string. ----- Name_get is overridden to add product category as suffix. ----- New method added to class, def get_stock_for_location to get the available quantities of product at selected location.

13. sale_stock : default sale_stock module is replaced by this module.

- Depends on : sale, stock, procurement
- There are few reports added and new fields added, which can also be added by inheriting classes or setting dependency on this module.
- New report has to be added for Average Turnaround time and sale turnaround time.
- Action for both the reports is added to stock_view.xml which need to be added in different file.
- **turnaround report requires sale_order, account_invoice and res_partner class, hence we can move this report to module which will be developed for custom sale related requirement.**

sale.config.settings	<ul style="list-style-type: none"> • New field has to be added : Convert_dispensed: which allows directly confirm sale order, if drug is dispensed from local shop. • New method has to be added: def set_convert_dispensed
sale.order	def _prepare_order_line_move has to be overridden to add prodlot_id to move_line values.

- stock.py is identical to that of stock.py file of default module, no change made to any method.

14. bahmni_web_extensions : depends on base, web

- accesskeyHighlight.js : Defining Accesskey in buttons. These Access keys are underlined and capitalized for buttons in which accesskey is defined.
- addContextOnExport.js : On click of export data, added domain, context and ids in returned action.

Bahmni Developed Modules Analysis

- addingAccessKeyToOne2ManyList.js: replaced string 'add an item' to Add an Item.
- addSerialNumberToListView.js: Added Sr.No. to each item in tree view. for each page, this sr.no. is generated.
- fixingErrorInCancelEdition.js: Error handling done in method cancel_edition. If attrs is not null, then only attrs.id condition has been checked.
- init.js : initialisation of all the above files.
- from the web controllers CSVExport controller is inherited to pass the context from action to the jquery method. and from jquery context is passed to the action, but can't identify the use of context.
- overriding index method has solved the issue of exporting csv, which was getting faced in default web module, but it still doesn't explains about use of context.
- **New class defined for creating warnings, but there is no need for creating new class just for that, as there are classes in openerp which are already available for that.**

15. print_receipt : depends on account_accountant

- **invoice_id field is referred in this module, while invoice_id field is defined in bahmni_customer_payment module, and there is no dependency set on that module.**
- Default report definition of account.voucher are commented and new report is defined, which is modified version of default report, as new fields has been added to account.voucher class, to include all those fields in report, new report has been developed. - **this could have been achieved by overriding report action, instead of commenting original report action in default module.**
- New button added to account.voucher form for printing report, which is an object type button, hence new method defined for that. - **As we already have report action for printing this report, we don't really need this button on form. Still keeping this button with string print receipt.**
- res.company class is inherited to change the header for reports in company.

16. scripts:

- There are shell scripts written for vagrant, not an openerp module.

17. search_customizations:

- onchange_partner_id method is overridden to directly populate amount to be paid in amount field in account.voucher.
- New reports are added for Accounts wise Collection and Expense report and Account wise invoice count report, which is grouped date-wise.
- In this module button for print and summarised bill are again added, which were already added in bahmni_print_bill module.
- When company logo is not set in the system, reports format are not getting properly printed. In this module, summarised bill is set to print in marathi - this thing can also be moved to configuration or we can provide 2 separate buttons for printing english and

Bahmni Developed Modules Analysis

marathi bill. Also, now there are two buttons on customer payment object saying print, but both buttons print different reports.

- There is one filter defined named : **Required Accounts Only** in this domain is hard coded for accounts, should it be configurable? Can't make filter configurable, user will need to create his own filter. Categories which are hard-coded are categories created through xml. But if that data is moved somewhere else/ or made optional to load, then this filter will raise error, as no such category exists.

18. search_seed_data : depends on base,product, stock, sale, sale_stock, bahmni_pharmacy_product

- In this module data for Accounts, Product Categories, Products, Unit Of measures, Stock Location, Warehouse and shop is created. - **search_setup_data**
- In this module again service product is deleted.
- data in bahmni_seed_setup and search_seed_setup is completely different.
- In settings.xml, main company data is changed, and it's partner data is also changed, partner's name, city, image, phone, email is changed. - **search_setup_data**
- Here, for changing image of company, i.e. logo binary code of image is given, can it be replaced with image? - **search_setup_data**

19. auth_crypt: default openerp module, whole module is overridden, just few changes to auth_crypt.py.

- password field is removed , which is a functional field, and hence it's functions are also removed.
- def check_credentials method is overridden.
- new method is defined : def get_salt_from.
- These things could have been accomplished by inheriting class, no need to replace the whole module, and **I consider, this customization is done to connect bahmni with openerp, moving this functionality to bahmni_atom_feed module.**