ERP5 Calendar

- Defining people calendar
- Defining how task fill the calendar
- Managing capacity

Support ERP5 !
www.erp5.org/howtohelp

© Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Defining Person availability

Personal Calendar

Source: person

Personal Calendar Period

Start Date: 2006/12/18-08:00
Stop Date: 2006/12/18-12:00
Resource: calendar_period_type/labour R
Quantity: Q

Stock

uid	node_uid	resource_uid	Quantity	Date	mirror_date
Calendar period	Person	R	Q	start_date	stop_date

Q > 0
Q in second

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Defining Person unavailability

Personal Calendar

Source: person

Personal Leave Period

Start Date: 2006/12/18-08:00
Stop Date: 2006/12/18-12:00
Resource: calendar_period_type/holiday R'
quantity: K

Stock

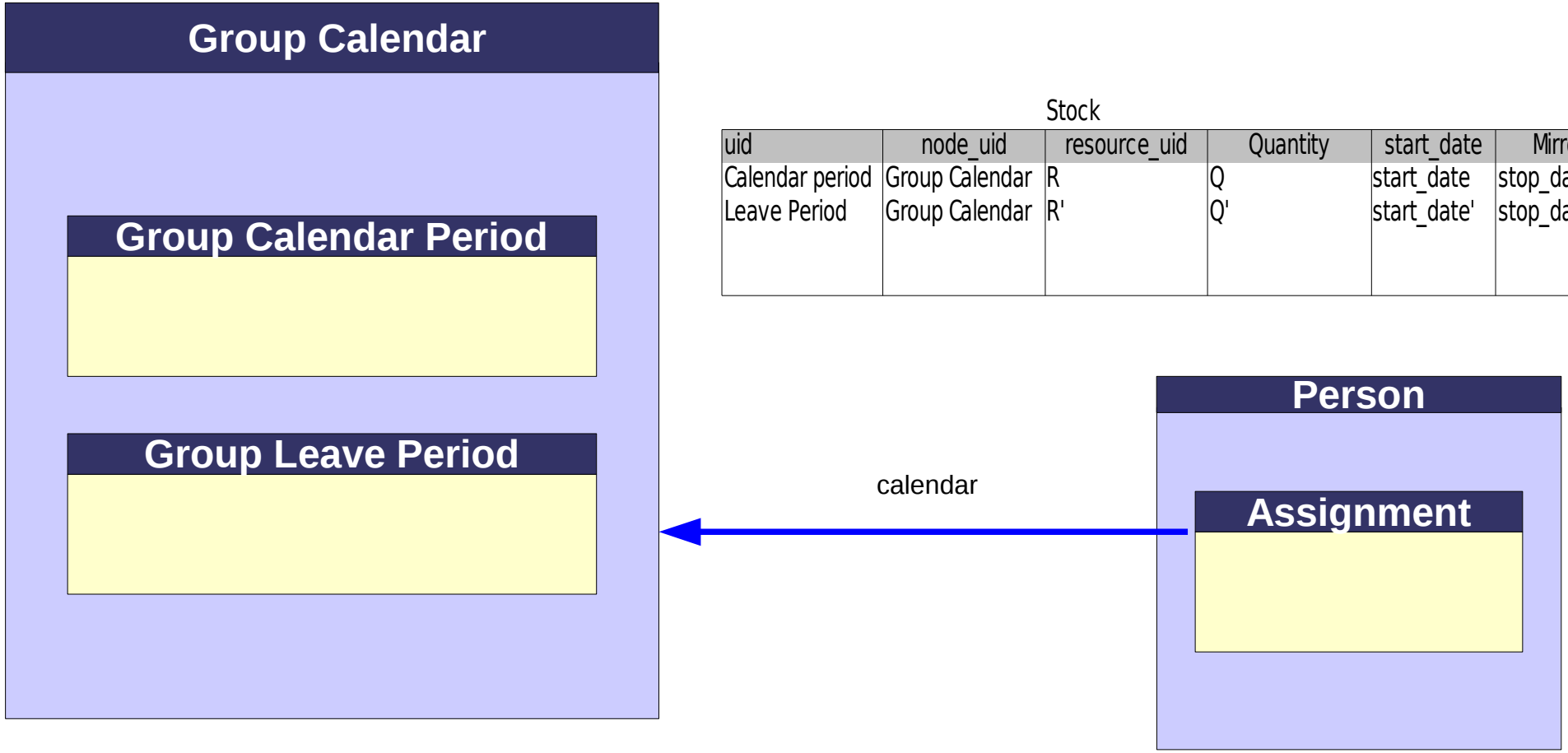
uid	node_uid	resource_uid	Quantity	Date	mirror_date
Leave period	Person	R'	K	start_date	stop_date

K < 0
K in second

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Multiple person (un)availability



Stock

uid	node_uid	resource_uid	Quantity	start_date	Mirror
Calendar period	Group Calendar	R	Q	start_date	stop_date
Leave Period	Group Calendar	R'	Q'	start_date'	stop_date'

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Recurring period

Group Calendar

Group Calendar Period

Start Date: 2006/12/18-08:00
Stop Date: 2006/12/18-12:00
Resource: calendar_period_type/labour R
Quantity: Q
Periodicity Stop Date: 2006/12/23
Recurring parameter: For each day

Calendar Exception

Exception Date: 2006/12/20

uid	...	Date	Mirror date
Calendar period		18/12/06 08:00	18/12/06 12:00
Calendar period		19/12/06 08:00	19/12/06 12:00
Calendar period		21/12/06 08:00	21/12/06 12:00
Calendar period		22/12/06 08:00	22/12/06 12:00

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Time API

```
def getAvailableTime(self, from_date=None, to_date=None,
                     portal_type=[], node=[], resource=[], **kw):
```

Calculate available time in second for a node
Returns an inventory of a single or multiple resources on a single node as a single float value

from_date (>=) - only take rows which mirror_date is >= from_date
to_date (<) - only take rows which date is < to_date
node - only take rows in stock table which node_uid is equivalent to node
resource - only take rows in stock table which resource_uid is equivalent to resource
portal_type - only take rows in stock table which portal_type is in portal_type parameter

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Time Sequence API

def **getAvailableTimeSequence**(self, from_date, to_date, portal_type=[], node=[], resource=[], **kw):

Calculate available time for a node in multiple period of time.
Each row is the available time for a specific period
second, minute, hour, day, month, year - duration of each time period (cumulative)

[duration_1, duration_2, ...]

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Capacity of a Person

Person

New kind of object ? Career ?

service_type / capacity_unit	second	kg	m3
development	100%	0%	0%
accounting	70%	0%	0%
cleaning	130%	0%	0%

getCapacityRatio(capacity_unit, service_type=None)

Return the ratio (no unit) for the specific capacity_unit.

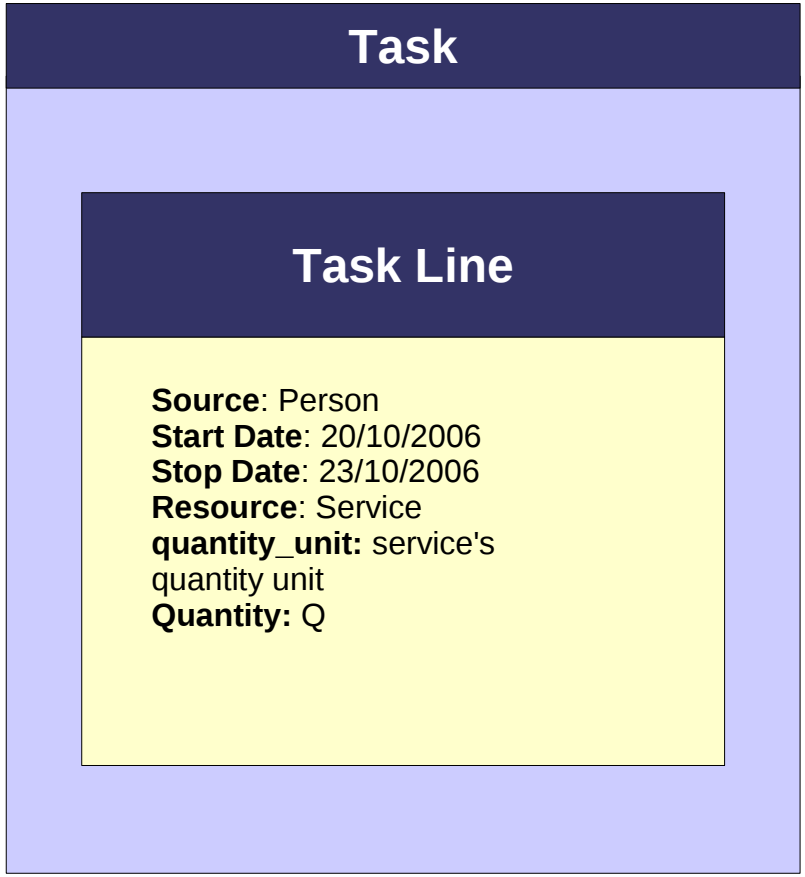
If service type is not specified, returns the average of available ratios.

- **capacity_unit**: returns the ratio for a specific capacity unit
- **service_type**: returns the ratio for a specific service type. If not specified, returns the average.

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

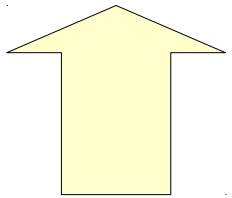
ERP5

Task consumes quantity of Service



Stock

uid	node_uid	resource_uid	Quantity	start_date
Task line	Person	Service	Q'	start_date



Q' in default service's quantity unit

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Service in capacity unit

Service

quantity_unit / capacity_unit	second	kg	m3
Line of code	1023	0	0
training	1200	0	0
wrapping	120	0	0

1 line of code = 1023 seconds

Convertion table

resource_uid	quantity_unit_uid	Capacity_uid	Conversion
Service	Line of code	Second	1023
Service	Training	Kg	0

Yes. What is the API ? data model for conversion ?

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Task consumes capacity

getInventory(quantity_unit=capacity_unit, **kw)

- **quantity_unit**: return the inventory in the converted unit

Stock

uid	node_uid	resource_uid	Quantity	start_date
Task line	Person	Service	Q'	start_date

Convertion table

resource_uid	quantity_unit_uid	Capacity_uid	Conversion
Service	Line of code	Second	1023
Service	Training	Kg	0

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Capacity API

```
def getAvailableCapacity(self, from_date, to_date,
                         portal_type=[], node=[], resource=[],
                         capacity_unit=None, **kw):
```

Calculate available capacity for a node.
Each row is the available time for a specific period

node - only take rows in stock table which node_uid is equivalent to node

portal_type - only take rows in stock table which portal_type is in portal_type parameter

resource - only take rows in stock table which resource_uid is equivalent to resource

from_date (>=) - return period which start >= from_date

to_date (<) - return period which start < to_date

capacity_unit – unit in which the capacity is returned (REQUIRED)

Person:

getAvailableCapacity = getAvailableTime() - getInventoryList(node=person_uid, quantity_unit=second, group_by_resource_type=1) * getCapacityRatio(second, service_type=resource_type)

Organisation:

getAvailableCapacity = SUM(person.getAvailableCapacity()) [person in organisation]

© Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Issues

- GetAvailableCapacity does not manage Task as flow of capacity

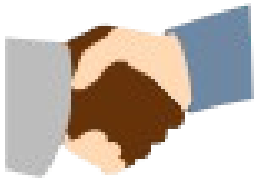
- getAvailableCapacity on Person: need hardcoded capacity unit value (second)

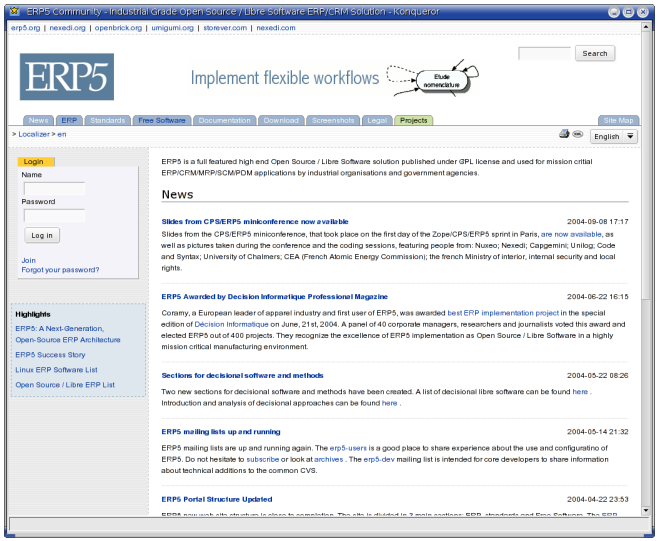
Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

Commercial Support

Enterprise Services

- Consulting
- Professional Training
- Custom Development



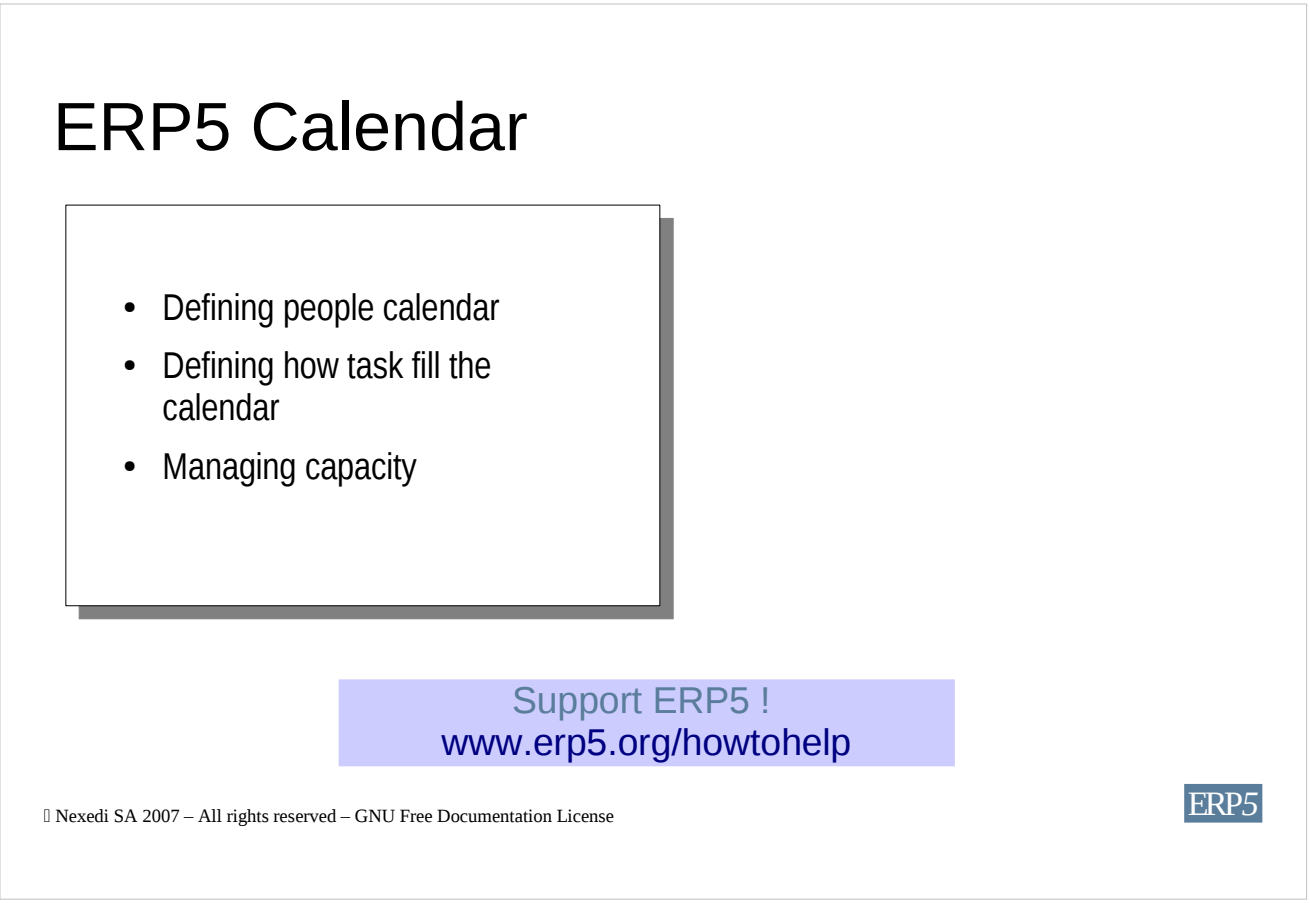


www.erp5.org
Open Portal

Join ERP5 Service Network !
www.erp5.org/support

© Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5



This visual guide has been created for learning and for teaching ERP5. You are free to copy it, modify it, represent it or distribute it as for any Free Software and Documentation created by Nexedi SARL

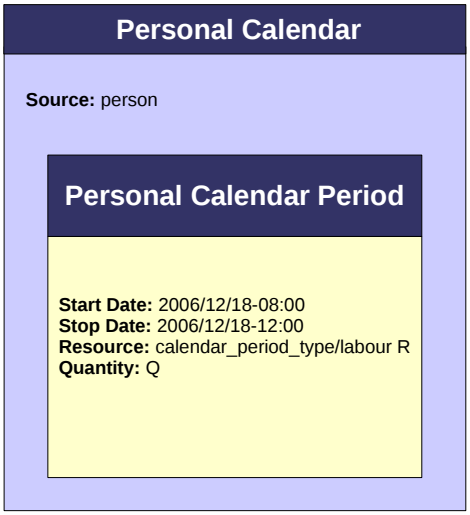
For learning ERP5 alone, readers should first have a quick look at the illustration on the upper part of the page then read the text bellow carefully and associate each word written in bold to the corresponding item(s) in the illustration. For example, the term **GNU Free Documentation License** is written in bold because it defines the license of the above illustration.

This visual guide can also be used as a presentation for teaching ERP5 to a public audience. In this case, teachers are required to read audibly to the audience the Back Cover Text.

Copyright

Permission is granted to copy, distribute and/or modify this document according to the terms of the GNU Free Documentation Licence, Version 1.1 (http://www.gnu.org/copyleft/fdl.html) with Front Cover Text "Support ERP5! www.erp5.org/howtophelp", with Back Cover Text "This visual guide has been created for learning and for teaching ERP5. You are free to copy it, modify it, represent it or distribute it as for any Free Software and Documentation created by Nexedi SARL" and with Invariant Section "Commercial Support".

Defining Person availability



Stock

uid	node_uid	resource_uid	Quantity	Date	mirror_date
Calendar period	Person	R	Q	start_date	stop_date

Q > 0
Q in second

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

Calendar Period are used to defined the availability of a **Person**, which means his working time period.

It does not defined what does the person during this time (this is represented by Task/Task Report).

Time availability is catalogued in the **stock** table.

The **quantity** is catalogued in **second,** which is the standard time unit.

Calendar Period's quantity is a **positive** value.

This Period can be **recurring** in order to simplify how the user enters his calendar.

Defining Person unavailability

Stock

uid	node_uid	resource_uid	Quantity	Date	mirror_date
Leave period	Person	R'	K	start_date	stop_date

K < 0
K in second

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

Calendar Event are used to defined the **unavailability of a Person (**holiday, strike, disease, ...).

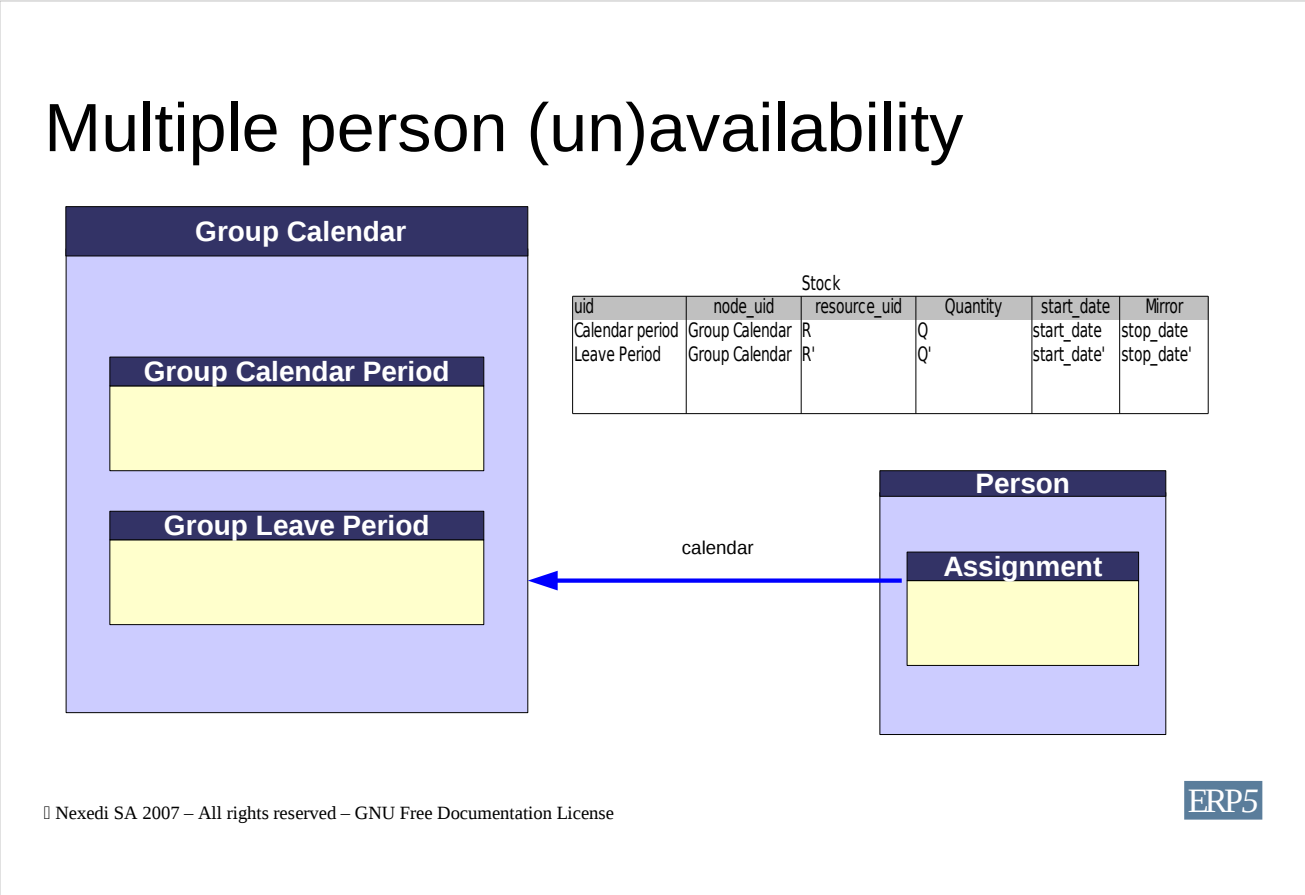
It defined why the person didn't work as he was supposed to ?

Time availability is catalogued in the **stock** table.

The **quantity** is catalogued in **second,** which is the standard time unit.

Calendar Period's quantity is a **negative** value.

This Event can be **recurring** in order to simplify how the user enters his calendar.



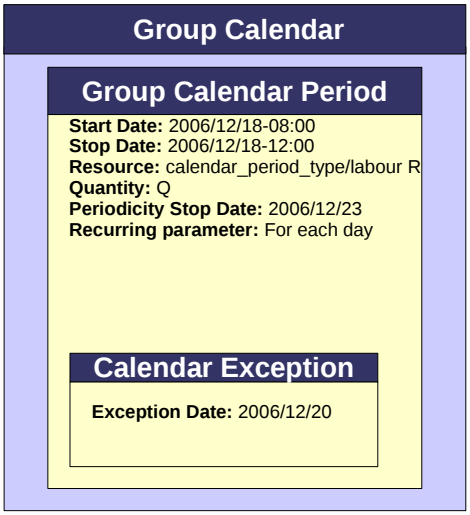
Group Calendar Period/Event are used to defined the (un)availability of a **Group of person**, which means there working time period.

The group of person is defined by associating the person's assignment to the Group Calendar through the calendar base calendar.

Except this, it works as personal calendar.

2 Period portal types are required, because the user interface may change (ex: to restrict the calendar_period_type categories). Another reason is that quantity may be calculated automatically (from start and stop dates values), and in one case, it has to be a positive value, and in the other case, a negative value.

Recurring period



uid	...	Date	Mirror date
Calendar period		18/12/06 08:00	18/12/06 12:00
Calendar period		19/12/06 08:00	19/12/06 12:00
Calendar period		21/12/06 08:00	21/12/06 12:00
Calendar period		22/12/06 08:00	22/12/06 12:00

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

All period portal types (Personal Calendar Period, Personal Leave Period, Group Calendar Period and Group Leave Period) may be defined as recurring period.

After defining the **periodicity stop date** and the **recurring parameters**, one object from the ZODB may be **catalogued multiple time** with different date values.

Calendar Exception defines a non catalogued period for a specific date.

Time API

```
def getAvailableTime(self, from_date=None, to_date=None,
                     portal_type=[], node=[], resource=[], **kw):
  Calculate available time in second for a node
  Returns an inventory of a single or multiple resources on a single
  node as a single float value
  from_date (>=)  - only take rows which mirror_date is >= from_date
  to_date   (<)   - only take rows which date is < to_date
  node            - only take rows in stock table which node_uid is
                    equivalent to node
  resource        - only take rows in stock table which resource_uid is
                    equivalent to resource
  portal_type     - only take rows in stock table which portal_type
                    is in portal_type parameter
```

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Time API is defined and implemented on Selection Tool.

Aliases can be added on others classes to automatically defined some parameter.

Ex: on the Person class, node parameter is automatically calculed as the combination of the person uid and group calendar uid related to person assignment.

Time Sequence API

def **getAvailableTimeSequence**(self, from_date, to_date, portal_type=[], node=[], resource=[], **kw):

Calculate available time for a node in multiple period of time.
Each row is the available time for a specific period
second, minute, hour, day, month, year - duration of each time period (cumulative)

[duration_1, duration_2, ...]

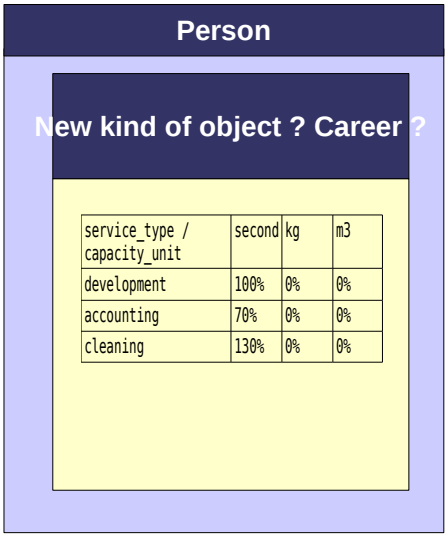
Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

Time Sequence API is defined and implemented on Selection Tool.

Aliases can be added on others classes to automatically defined some parameter.

Ex: on the Person class, node parameter is automatically calculed as the combination of the person uid and group calendar uid related to person assignment.

Capacity of a Person



service_type / capacity_unit	second	kg	m3
development	100%	0%	0%
accounting	70%	0%	0%
cleaning	130%	0%	0%

getCapacityRatio(capacity_unit, service_type=None)

Return the ratio (no unit) for the specific capacity_unit.

If service type is not specified, returns the average of available ratios.

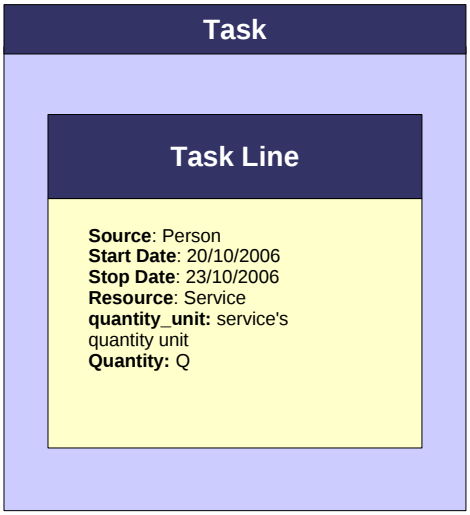
- **capacity_unit**: returns the ratio for a specific capacity unit
- **service_type**: returns the ratio for a specific service type. If not specified, returns the average.

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

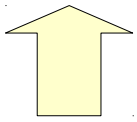
Capacity of a Person represents the potential output that a person can produce.

Task consumes quantity of Service



Stock

uid	node_uid	resource_uid	Quantity	start_date
Task line	Person	Service	Q'	start_date



Q' in default service's quantity unit

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

Task Line **consumes a quantity** of a service unit.

It's the answer of: what did the person do during his working time ?

The **quantity_unit** is a valid unit from the Service.

Service in capacity unit

Service

quantity_unit / capacity_unit	second	kg	m3
Line of code	1023	0	0
training	1200	0	0
wrapping	120	0	0

1 line of code = 1023 seconds

Convertion table

resource_uid	quantity_unit_uid	Capacity_uid	Conversion
Service	Line of code	Second	1023
Service	Training	Kg	0

Yes. What is the API ? data model for conversion ?

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

The **quantity_unit** define the list of valid quantity unit for the Service.

It will be used later by **Task, Capacity, ...**

Task consumes capacity

getInventory(quantity_unit=capacity_unit, **kw)

- **quantity_unit**: return the inventory in the converted unit

Stock

uid	node_uid	resource_uid	Quantity	start_date
Task line	Person	Service	Q'	start_date

Convertion table

resource_uid	quantity_unit_uid	Capacity_uid	Conversion
Service	Line of code	Second	1023
Service	Training	Kg	0

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

Task Line **consumes a quantity** of a service unit.

It's the answer of: what did the person do during his working time ?

The **quantity_unit** is a valid unit from the Service.

Capacity API

def **getAvailableCapacity**(self, from_date, to_date,
portal_type=[], node=[], resource=[],
capacity_unit=None, **kw):

Calculate available capacity for a node.
Each row is the available time for a specific period

node - only take rows in stock table which node_uid is equivalent to node
portal_type - only take rows in stock table which portal_type is in portal_type parameter
resource - only take rows in stock table which resource_uid is equivalent to resource
from_date (>=) - return period which start >= from_date
to_date (<) - return period which start < to_date
capacity_unit – unit in which the capacity is returned (REQUIRED)

Person:

getAvailableCapacity =
getAvailableTime() - getInventoryList(node=person_uid, quantity_unit=second, group_by_resource_type=1) * getCapacityRatio(second, service_type=resource_type)

Organisation:

getAvailableCapacity =
SUM(person.getAvailableCapacity()) [person in organisation]

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

ERP5

Time API is defined and implemented on Selection Tool.

Aliases can be added on others classes to automatically defined some parameter.

Ex: on the Person class, node parameter is automatically calculed as the combination of the person uid and group calendar uid related to person assignment.

Issues

- GetAvailableCapacity does not manage Task as flow of capacity

- getAvailableCapacity on Person: need hardcoded capacity unit value (second)

Nexedi SA 2007 – All rights reserved – GNU Free Documentation License

The ERP5 project is coordinated by Nexedi. Certified Consulting and Development services are provided worldwide by a network of partner companies.

Just like any ERP, ERP5 is complex system which requires experience in order to provide complete satisfaction to clients. Configuring ERP5 requires much more effort and experience than simply installing ERP5 software components alone.

Companies interested in joining ERP5 professional services network are encouraged to contact Nexedi. Nexedi provides complete training and certification services. Nexedi can assist ERP5 service providers in providing optimal presales, implementation and aftersales service to enterprise clients. Nexedi is looking forward to assist consulting or software companies to build a sustainable, profitable and independent business.

For more information, please refer to www.erp5.org/support.