SACM Routing

Routing

- CI Routing identyfikujemy na podstawie id

Atrybuty CI Routing

Atrybut	Opis
Routing_ID	Unikalny identyfikator routingu, tworzony podczas jego wdraania

Typy Relacji

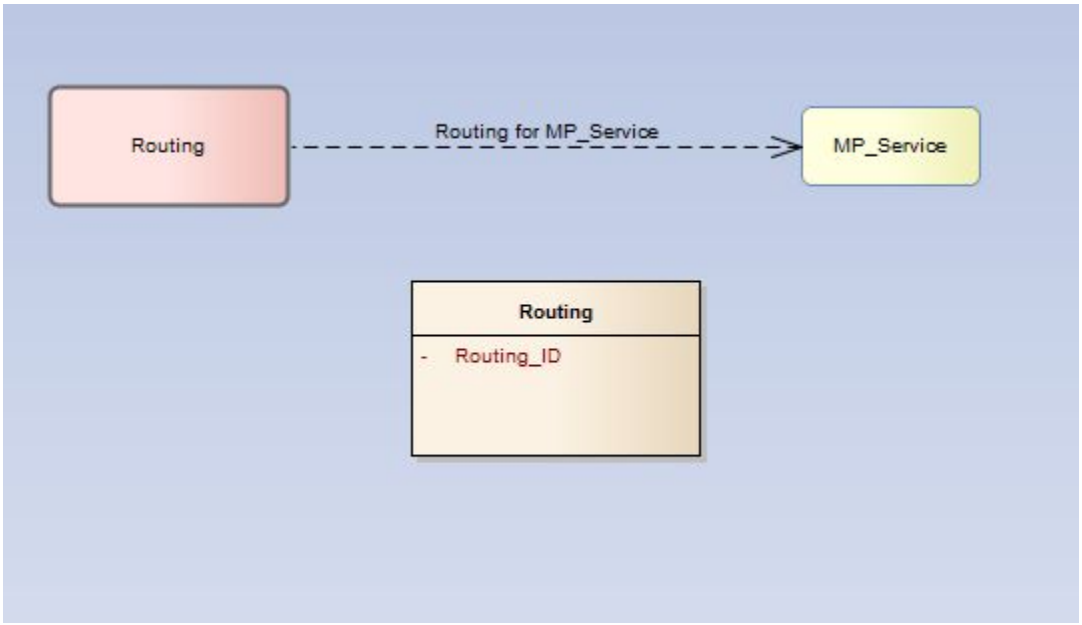
Routing jest w relacji z MP_Service poprzez unikalne ID jakie posiada kada mp usuga.

Cykl ycia Routingu

Aktywnoci powodujce zmian:

- *New LA SMS*
- *SR_Konfiguracja_SimpleResponseServie*
- *SR_Routing*

Dokumentacja



Potoczne typy:

- IN (Wiele mp usug powizanych z danym routingiem zawierajcym globaln warto)
- OUT (MP_SERVICE_ID=ROUTING_ID)

Na potrzeby SACM na bazie zosta utworzony widok ***routing_to_consequence_id***, obrazujcy relacje midzy *routing ID*, a serwisami/konsekwencjami na jakie kieruje jaki ruch

```
create view routing_to_consequence_id as
select routing_id, extractValue(value, '/routing-element/consequence/service-id') as
consequence_id, extractValue(value, '/routing-element/@priority') as priority
from routing, table(xmlsequence(extract(xmltype(LINEAR_ROUTING),
'/routing/routing-element'))) x
```

Routingi s to pliki konfiguracyjne xml suce do zarzdzania transmisj wiadomoci pomidzy usugami (rozumianymi tu równie jako bramki, serwisy oraz samo MP) **Prefix** jest to charakterystyczny cig znaków, znajdujacy si na pocztku treci sms,a definiowany w celu kierowania obsugi do

okrelonej usugi **LA** jest to numer bramki np. 7222 7333 92544 na jaki lub z jakiego kierowana jest wiadomos **IN** <- smsm przychodzcy do Avantis **OUT** <- sms wychodzcy z Avantis

Dziaanie routingu: na kocu routingu znajduje si lista "zbinodowanych" serwisów. Kady podbinodwany serwis wysya wiadomoci do routingów ,a uda mu si, trafi na taki który go przejmie:

Sów kilka o tym jak dziaa routing:

```
<routing-element priority="100">

<= tu si zaczyna pojedynczy element routingu
  <condition operator = "and">

    <= to oznacza, ze spenione musz by wszystkie kolejne warunki w klauzuli AND
      <condition operator = "equals" expression = "operator" value = "26003" />

        <= tu okrelamy operatora
          <condition operator = "or">

            <= tu otwieramy klauzul OR która mówi e wystarczy aby by sepniony jedenz kolejnych warunków
              <condition operator = "equals" expression = "sender" value = "7136" />

                <= warunek sender musi mie okrelon warto
          </condition>

            <= koniec warunków w OR
  </condition>

    <= koniec warunków w AND
  <consequence>

    <= w tej czesci opisujemy co MP ma zrobi z wiadomoci
      <service-id>1248</service-id>

        <= mpid usugi kótra ma obsuy waidomo
  </consequence>

    <= koniec obsugi
</routing-element>

  <= koniec routingu
```

Mierniki

Prawidowa zmiana routingu to taka która:

- pozostawia po sobie informacje o username i tasku który da si zmapowa na aktywno procesu (np. po nazwie projektu w mantisie)
- zostaa odpowiednio przetestowana
- jeli bya czyniona w ramach Request Fulfilment to task musi by odpowiednio zamknity

Audyt ma suy zdobywaniu informacji o rzeczywistych zmianach. Moe te suy jako wyzwalacz potrzeby modyfikacji procesu, chyba e zostao to jawnie okrelone gdzie indziej w procesie.
Audyt na wejciu dostaje fakt samej zmiany, na wyjciu ma udostpnia nastpujce informacje:

- id zmiany
- id zmienionego routingu
- data i czas zmiany
- kto dokona zmiany
- z jakiego taska wynikaa zmiana
- w której aktywnoci procesu doszo do zmiany

CM_FAIL

- Monitoruje wszelkie ze zmiany na routingach, obecnie dane uwaane za z zmian jest brak taska na podstawie, której ta zmiana zasza.

```
select ar.user_name, ar.archive_time "TIME", ar.archived_routing_id Routing, ar.task_id,
ar.sequence_id
from mp.arch_routing ar
where ar.archive_time > current_timestamp - interval '1' day
and (ar.task_id is null
or upper(ar.task_id) like '%N/A%'
or ar.task_id like '0'
or ar.user_name is null)
order by ar.archived_routing_id,ar.user_name,ar.archive_time;
```

CM_COUNT

- Podlicza ilo nieprawidowych zmian wykrytych w CM_FAIL, zmian, które zostay poprawione.

CM_ERROR

- Pokazuje wszystkie róda za pomoc, których nastpuj wszelkie zmiany. Okrela ich ródo (Projekt, Kategori, Task) oraz alarmuje w razie kiedy zmiana nastpia z nieznanego wczeniej nie okrelonego róda, bd z nieprawidowego róda.

Pokazuje ródo zmiany w JIRA

```
select jd.pkey, jp.pname as projekt, it.pname as issuetype, cfd.customvalue as kategoria, jd.reporter as
zglaszajacy, jd.assignee as przypisany_do
from JIRA_DNIU_RO.JIRAISSUE_DNIU jd
join jira.project jp on jp.id = jd.project
join jira.issuetype it on it.id = jd.issuetype
left join jira.customfieldvalue cfv on cfv.issue = jd.id and cfv.customfield = 10401
left join jira.customfieldoption cfd on cfd.id = cfv.stringvalue
where jd.pkey like '%TASK%';
```

Pokazuje ródo zmiany w Mantis

```
SELECT mbt.id as ID, mbt.summary AS Tytul,to_char(mbt.date_submitted::abstime::date,'yyyy-mm-dd') as
data_zgloszenia,mutr.username as zglaszajacy,
mpt.name as projekt,mct.name as kategoria
FROM mantis_bug_table mbt
JOIN mantis_project_table mpt ON mbt.project_id = mpt.id
JOIN mantis_user_table mutr ON mbt.reporter_id = mutr.id
JOIN mantis_category_table mct ON mbt.category_id = mct.idwhere
CAST(mbt.id AS TEXT) like '%TASK%'
ORDER BY mbt.date_submitted,id;
```