

ServiceNow web service for creating and updating Incidents

Goal:

We are going to need the ability to create and update tickets (or whatever they are called) in ServiceNow from other ESB services. Would you reach out to Sriram and Luc and start the process of documenting any web services that ServiceNow exposes for creating and updating tickets. What I'd like to have are:

1. Add any existing ServiceNow services related to tickets not in the Emory web service catalog to the Emory web service catalog.
2. Add/create a ServiceNow MOA — I think there may already be one that was created for the existing ServiceNow service or connector that was created for the user provisioning integration
3. Add a ticket object or whatever it is called in ServiceNow jargon and sample messages for create, update, delete, and create sync, update sync, and delete sync actions and regenerate the MOA
4. I believe we may already have a ServiceNow service that exposes requests to ServiceNow to the ESB. Determine the precise status of all of that and let me know. If there is an existing ServiceNow service or connector project in Subversion confirm what it does and point me to that otherwise create a new code project and point me to it.

Conclusion

- 1) There are two Moa objects: ServiceNowUser, Incident
- 2) There are two commands in the edu.emory.ServiceNowGateway2.0 Service

ServiceNowSyncCommand:

- FullPerson Sync Request, which in turns call ServiceNowRequestCommand
- paging.Event which in turns AmCom?

ServiceNowRequestCommand

- takes ServiceNowUser object from passed over from ServiceNowSyncCommand, and call ServiceNow web service to insert

- 3) Incident Moa object is currently NOT being used

Question

1) Should a different Moa object (ServiceRequest) needed in addition to Incident since There is only one object called Incident in ServiceNow? I would use ServiceRequest moa to replace Incident as current Incident moa definition is not sufficient.

2) What's the rationale for the current Incident Moa structure?

```
<!ELEMENT Incident (Number?, ConfigurationItem?, AssignmentGroup?, AssignedTo?, NocAlarmId?, ShortDescription?, WorkNotes*, Priority?, IncidentState?)>
```

```
<!ELEMENT ConfigurationItem (Name?, SupportGroup?, AmcomEventId?, ServiceOwner?, ServiceOwnerEventId?)>
```

```
<!ELEMENT AssignmentGroup (Name?)>
```

```
<!ELEMENT AssignedTo (PublicId?, PersonalName?)>
```

3) Here is the proposed Moa definition of ServiceRequest (newly added fields are underlined). There are 147 fields in the ServiceNow WSDL Incident Object. How many should we use?

```
<!ELEMENT ServiceRequest (Number?, SysId?, UserNetId?, ContactNetId?, ConfigurationItem?, AssignmentGroup?, AssignedTo?, NocAlarmId?, ShortDescription?, WorkNotes*, Priority?, IncidentState?)>
```

```
<!ELEMENT ServiceRequestQuerySpecification (Number?, SysId?, UserNetId?)>
```

4) Here is the proposed Moa definition of Incident version 2 for use as both Incident and ServiceRequest (newly added fields are underlined). There are 147 fields in the ServiceNow WSDL Incident Object. How many should we use?

```
<!ELEMENT Incident (Number?, SysId?, UserNetId?, ContactNetId?, ConfigurationItem?, AssignmentGroup?, AssignedTo?, NocAlarmId?, ShortDescription?, WorkNotes*, Priority?, IncidentState?, RecordType?)>
```

```
<!ELEMENT IncidentRequestQuerySpecification (Number?, SysId?, UserNetId?)>
```

Note: The value for RecordType are Incident or Service Request

Existing Artifacts:

Moa and Sample message definition:

- message/releases/com/service_now/ServiceDesk/Incident

build and packaging on the emory package server:

- service_now-moas
- emory-servicenow-gateway-source-2.0

project source on the subversion:

- emoryoit/project/emory-servicenow-gateway/2.0

ServiceNowSyncCommand:

- FullPerson(Create,Update,Delete)=>FullPersonSyncHandler(p2p)=>ServiceNowUserRequestor(Create,Update)=>ServiceNowRequestCommand
- paging.Event (Create)=> EventSyncHandler=>IncidentClient,ServiceNow_incidentStub=>AmCom? (Can Kelly elaborate here???)

ServiceNowRequestCommand:

- ServiceNowUser(Create,Update)=>ServiceNowUserRequestHandler.createRequest=>ServiceNow_u_emory_imp_userStub.insert
- Incident=>ServiceNowIncidentRequestHandler: not doing anything (Kelly confirm?): dev instance only accepting ServiceNowUser

[ServiceNow+Analysis+Template](#)

[Service-Now integration schematic](#)

To access these wsdl, use password from the properties in the P2P consumer of edu.emory.ServiceNowGateway from dev openeai console

- <https://emory.service-now.com/incident.do?WSDL>
- <https://emorytest.service-now.com/incident.do?WSDL>
- [//https://emorymud.service-now.com/incident.do?WSDL](https://emorymud.service-now.com/incident.do?WSDL)

Deployed Service (In production)

- edu.emory.ServiceNowGateway: it has a pub-sub consumer and a p2p consumer.

Terminology in ServiceNow per Kevin Chen

1. Incident: when using as a generic term in ITIL context (nothing to do with ServiceNow), an incident means something is broken. In our implementation, an incident really means something is broken in production.
2. Request: when using as a generic term in ITIL context, a request means a request to get something done such as have a new software installed on our laptop or create a new VIP. In the AWS workflow case, we need to submit a request to the security team for VPN tunnel creation.
3. In ServiceNow implementation, request is implemented as a special type of incident. Both request and incident are handled by the ServiceNow Incident module which uses the same data structure for both incident and request. Incident has an incident type as "incident", and request has an incident type as "request".
4. The ServiceNow web service to handle both incident and request are the Incident web service. The wsdl should be the same for both incident and request. Only difference is the incident type data value is different.

Ref

At Emory:

[Service-Now Integration - Foundation Phase](#)

[ServiceNow+Analysis+Template](#)

Outside of Emory:

[ServiceNow Integration Documentation](#)

[ServiceNow WS Doc](#)