

Arquitectura de Integração de Sistemas (AIS) - 2º Semestre 2020/2021

Prof. Responsável: Prof. Ricardo Gonçalves

Regente: Prof. Carlos Agostinho

Professor: Prof. José Ferreira

Trabalho Prático N°3: Desenvolvimento de Processos de Negócio

#Lab 3: Utilização do ACTIVITI para modelar e executar processo de negócio (continuação)

- **Objetivo:** Pretende-se continuar o modelo da aula anterior para representar o processo de um sistema inteligente que seja capaz de correlacionar a temperatura de uma casa com a temperatura exterior, tomando decisões para ligar ou desligar equipamentos HVAC em casa, de acordo com a temperatura de conforto escolhida por um utilizador. Nesta aula vamos ainda estudar adaptações ao processo e sua implementação para estimular a problemática da interoperabilidade.

#2.3: Continuando a aula anterior: Correlacionar a Temperatura

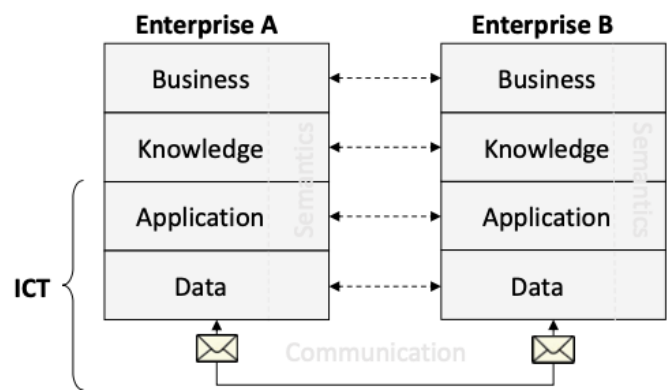
- **Exercício 2.3.0:** Agora que já conseguimos ler a temperatura exterior de forma automática e que, ainda se forma manual, também temos acesso à temperatura interior, utilizem o Activiti para criar um novo processo, introduzindo Gateways que tragam alguma inteligência.
 - Vamos a uma proposta simples: Se a temperatura interior for diferente da exterior, liga HVAC (ainda não sabemos qual equipamento) e mostra temperaturas. Tomem nota que as condições de saída do gateway têm que ser configuradas no ACTIVI fazendo uso das variáveis em definidas.
 - Incluir print-screen do novo processo
- **Exercício 2.3.1:** Tendo em conta o objetivo do lab, descrevam textualmente como poderíamos tornar o processo mais realístico, i.e., o que fazer com as temperaturas lidas? De que forma a temperatura da casa pode ser gerida para tirar melhor proveito da informação da temperatura interior e exterior?

#3.0: Um exemplo de um sistema um pouco mais inteligente

Podem encontrar no Moodle uma versão um pouco mais inteligente do nosso processo.

1. Importam o zip 'Exercicio30App.zip' cedido neste trabalho. Após importarem o ficheiro, podem verificar que aparece a App 'Exercicio30App' juntamente com o seu processo. Este processo faz uso das seguintes tarefas BPMN:
 - "User Tasks": onde a interação com o utilizador é necessária, seja para pedir informações ou mostrar informação

- “Service Tasks”: onde a interoperabilidade com o sistema é necessária. Como visto na aula anterior a service task faz uso de bibliotecas pré-programadas em JAVA. No caso apresentado temos um exemplo de interoperabilidade ao nível dos dados e aplicações (ver figura), uma vez que usamos uma API fornecida por uma empresa externa.



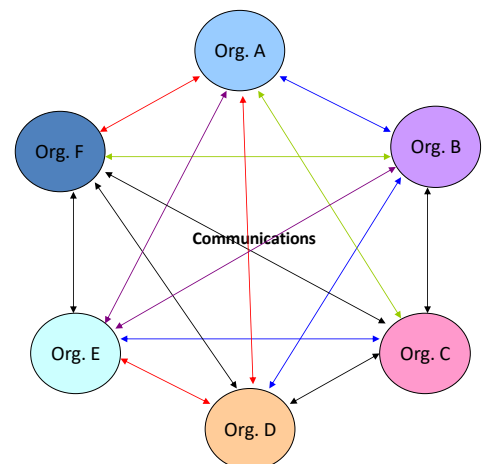
- “Script Tasks”: onde o automatismo do sistema é executado diretamente no ACTIVITI, no nosso caso apenas para efeitos de simulação
2. Verifiquem e modifiquem todas as configurações do processo importado, nomeadamente a ligação à vossa biblioteca para ler as temperaturas do OpenWeather, e o nome das variáveis usadas.
 3. Executem o processo e testem os valores

- **Exercício 3.0.0:** Descrevam o comportamento do processo. Qual a logica implementada nas condições?
- **Exercício 3.0.1:** Identifiquem o que está mal neste processo e modelem uma versão que previna a possibilidade de uma utilização errada do sistema. Incluir print-screen do novo processo. (Para não perderem muito tempo, o processo não precisa de ser executado. Apenas estamos interessados na modelação)

#3.1: Situações de interoperabilidade

Agora que o ACTIVITI e a modelação de processos já não tem segredos vamos testar casos em que seja necessário estar atento a diferentes problemas de interoperabilidade.

1. Já verificamos que a “service task” com a ligação à API do OpenWeather é um exemplo clássico de interoperabilidade de dados e aplicações. A forma como foi implementada pode ser ainda categorizada como “unregulated interoperability” (ver figura) uma vez que foi desenvolvido uma classe (“conector”) que invoca o serviço daquela empresa. No caso de querermos aumentar a robustez do nosso sistema e estarmos preparados para nos ligar a uma API de outra empresa, teríamos de desenvolver outro conector seguindo a especificação de outra API.



- **Exercício 3.1.0:** Implementem a funcionalidade para poderem usar um conector alternativo ao OpenWeather e desenvolvam a temperatura exterior usando a média das temperaturas recebidas. Podem usar o Accuweather (<https://developer.accuweather.com/packages>), subscrevam a versão free.
 - Façam registo no site do accuweather. Como na API do OpenWeather é necessário uma key para invocar o serviço. Para conseguir a key, carregue na tab ‘My Apps’,

preenchem os dados pedidos e criem a App. No final vão ter a App aprovada, podendo retirar a key.

- Comecem por testar a API “current conditions” e o método com o mesmo nome usando o Postman, com os seguintes comandos:

- URL: `http://dataservice.accuweather.com/currentconditions/v1/{location key}`
- A ‘location key’ para Lisboa é 274087 e Almada 275589. Para obterem keys para outras cidades podem usar o método “city search” da API “Locations”: <https://developer.accuweather.com/accuweather-locations-api/apis/get/locations/v1/cities/search>
- Nos parâmetros adicionem:
 - ‘apikey’, com o valor que retiram do site, explicado em cima.
 - ‘language’ com o valor pt-pt,
 - ‘details’ com o valor false (podem testar a true para verificarem a diferença)
- Nota: também poder testar o serviço diretamente no site to AccuWeather: <https://developer.accuweather.com/accuweather-current-conditions-api/apis/get/currentconditions/v1/%7BlocationKey%7D>

- Preparem o jar para poderem invocar este serviço e o serviço anteriormente implementado, obtendo a média da temperatura da cidade através de dois conectores

- Atualizem a biblioteca JAR no ACTIVITI e executem. Incluir um print-screen do código desenvolvido

- **Exercício 3.1.1:** Se quisermos classificar este tipo de interoperabilidade em Técnica, Sintática ou Semântica, qual escolham e porquê?

2. (opcional) Como podem verificar, a “User Task” “Entrar em Casa”, está a servir como configuração de um utilizador. Vamos agora imaginar que poderíamos usar esta configuração para automaticamente regular a temperatura em outros espaços por nós frequentados.

- **Exercício 3.2.0:** Adicionem uma “Service task” cuja função é exportar a configuração de utilizador e temperatura de conforto para um ficheiro JSON para que este possa ser lido por outro sistema de controlo de temperaturas (e.g. num quarto de hotel)