Pag 121

Interoperability atributo de calidad

Interoperability se refiere al grado en que 2 o más sistemas puede intercambiar información vía interface en un contexto particular.

Si tiene un grupo de sistema se conoce como Sistema de Sistemas y tiene los tipos:

Tipos de Sistemas de Sistemas (SoS)

1 SoS Dirigido

Tiene objetivos comunes y una administración centralizada, toda la autoridad es oficial y el resto de los sistemas se subordina al sistema principal.

2 SoS de conocimiento

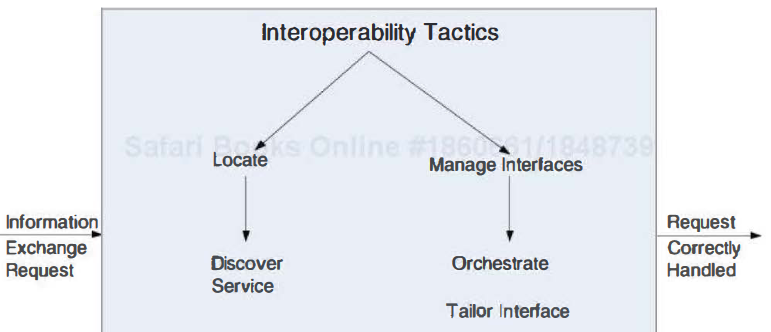
Tiene objetivos comunes y una administración centralizada, pero los subsistemas tienen su propia administración.

3 SoS colaborativo

No hay objetivos comunes, una administración centralizada, los sistemas voluntariamente trabajan para lograr objetivos comunes.

4 SoS virtual

Trabaja igual que SoS colaborativo, pero los sistemas no se conocen entre sí.



Hay estándares, pero no son suficientes para garantizar la interoperabilidad porque las necesidades de cada sistema son muy particulares.

El escenario

|  |  |
| --- | --- |
| Fuente de estimulo | Un sistema inicia una solicitud para interoperar con otro sistema. |
| Estimulo | Una solicitud de para intercambiar información con otros sistemas |
| Artefactos | El sistema que desea interoperar |
| Entorno | El sistema que desea interoperar es descubierto en tiempo de ejecución o es conocido desde antes |
| Respuesta | La solicitud para interoperar, la información es entendida por el sistema receptor  de manera sintáctica y semántica.  La solicitud es (apropiadamente) rechazada o aceptada, y las entidades (sistema o  personas) son notificadas.  La solicitud es (apropiadamente) aceptada y la información es intercambiada  exitosamente  la solicitud es grabada en bitácora por uno o más sistemas involucradas |
| Medición de la respuesta | Es el porcentaje información intercambiada exitosamente  Es el porcentaje información rechazada |

Tácticas

1 Locate

1.2 Discover service,

Localiza un servicio a través de un barrido de alguna carpeta, el servicio puede localizarse por tipo de servicio, por nombre, por localidad, u otro atributo.

2 Administrar interfaces

2.1 Orchestrate

Es una táctica que administra la secuencia de invocación de servicios específicos, la orchestracion se usa porque la interacción es complicada. La orchestration es un tipo de “scripts” que facilita la interacción.

2.2 Tailor interface

Es una táctica que agrega o elimina capacidades a una interface.

Las capacidades pueden ser traducción, agregar capacidad de buffer o normalizar los datos que se van a agregar, el caso en que eliminan capacidades es para usuarios no confiable. Un ejemplo de esta táctica es el pattern decorator.

De la lista de tácticas seleccionadas contrastar contra ……

**Checklist de diseño (contrastar con Categorías de decisiones de diseño arquitecturales)**

**Son para contrastar si las decisiones tomadas cumplen esas preguntas o sentencias (asignaciones)**

|  |  |
| --- | --- |
| Asignación de responsabilidades | Determinar cuál de las responsabilidades del sistema necesitan operar con otros sistemas.  Asegurar:  -aceptar el request  -intercambiar la información  -rechazar el request  -notificar a las entidades involucradas (sistemas, personas)  -corregir o enmascarar la fault o failure  -Loggear el request (almacenar en bitácoras Log) |

|  |  |
| --- | --- |
| Modelo de coordinación | Asegurar que el mecanismo de coordinación satisfaga los requerimientos  Considerar:  -el volumen de tráfico generado por tu sistema y generado por otros sistemas que no controlas.  -tiempo de envió de mensajes entre sistemas  -registro de mensajes enviados por otros sistemas  -inestabilidad de los mensajes enviados por “tus” sistemas |

|  |  |
| --- | --- |
| Modelo de datos | Determinar la sintaxis y semántica de las abstracciones de datos mayores y que pueda ser utilizado para intercambiar información.  -asegurar que las abstracciones de datos mayores sean consistente con los datos de los sistemas con los que interactúa, es posible que se hagan transformaciones desde o para las abstracciones de datos de otros sistemas con los que interactúa. |

|  |  |
| --- | --- |
| Mapeo entre elementos arquitecturales | Para la interoperabilidad el mapeo o correspondencia más crítica o complicada es la de componente a procesadores.  Aunque es necesario que los componentes estén soportados por procesadores que accedan a las redes, lo más importante es la seguridad y disponibilidad. |

|  |  |
| --- | --- |
| Administración de recursos | Asegurar que la interoperación con otros sistemas (aceptar o rechazar solicitudes) nunca llegue a un estado critico como “cuello de botella”.  Asegurar un nivel aceptable de recurso para soportar la comunicación de las solicitudes.  Asegurar que haya un buen protocolo para la asignación de recursos. |

|  |  |
| --- | --- |
| Binding time | Determinar cómo los sistemas pueden alcanzar una interoperación y como se reconocen entre sí mismos.  Por cada sistema asegurar:  -asegurar la existencia de una política para el manejo de conexión con sistemas conocidos y desconocidos.  -asegurar que existan y funcionen los mecanismos para rechazar solicitudes y que queden registrados en bitácora (loggeados).  -en caso de conexión tardía asegurar los mecanismos que soporten el descubrimiento (rastreo) de nuevos servicios o protocolos, o el envío de información usando protocolos establecidos. |

|  |  |
| --- | --- |
| Selección de tecnología | De las tecnología seleccionadas, son visibles en el espacio de interfaces del sistema ? y si es visible que efectos de interoperabilidad va a haber ?  Los efectos son limitados, parciales o no hay efectos ?  Asegurar un nivel de efectos aceptables  Considere tecnologías que fueron diseñados para soportar interoperabilidad como servicios web, estos recursos pueden ser usados para satisfacer la interoperabilidad de “su” sistema ? |