

# Apéndice A - Servicios del OMS y del SF

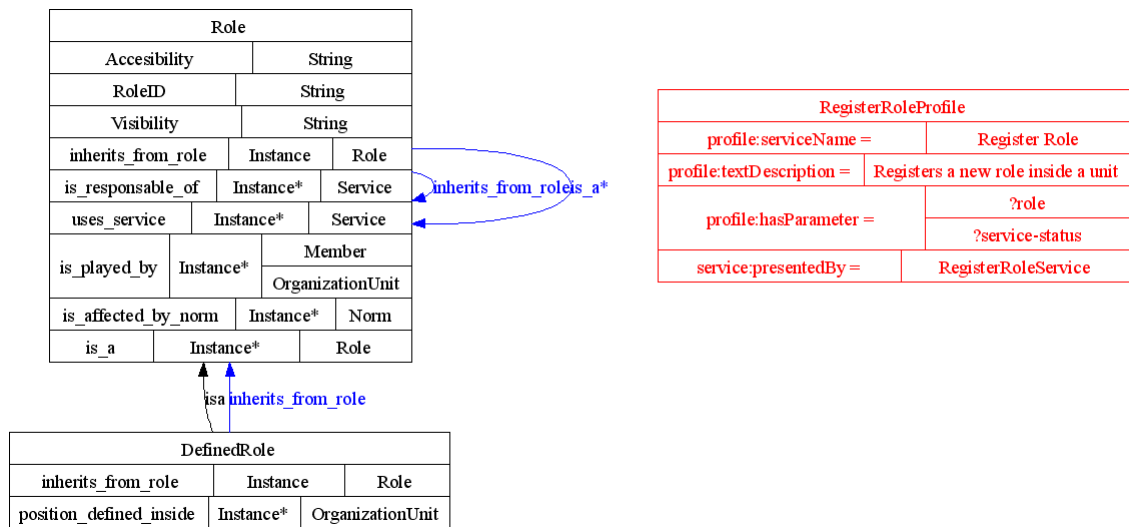
## Servicios OMS

### RegisterServices

A continuación se describen los servicios de registro de componentes estructurales:

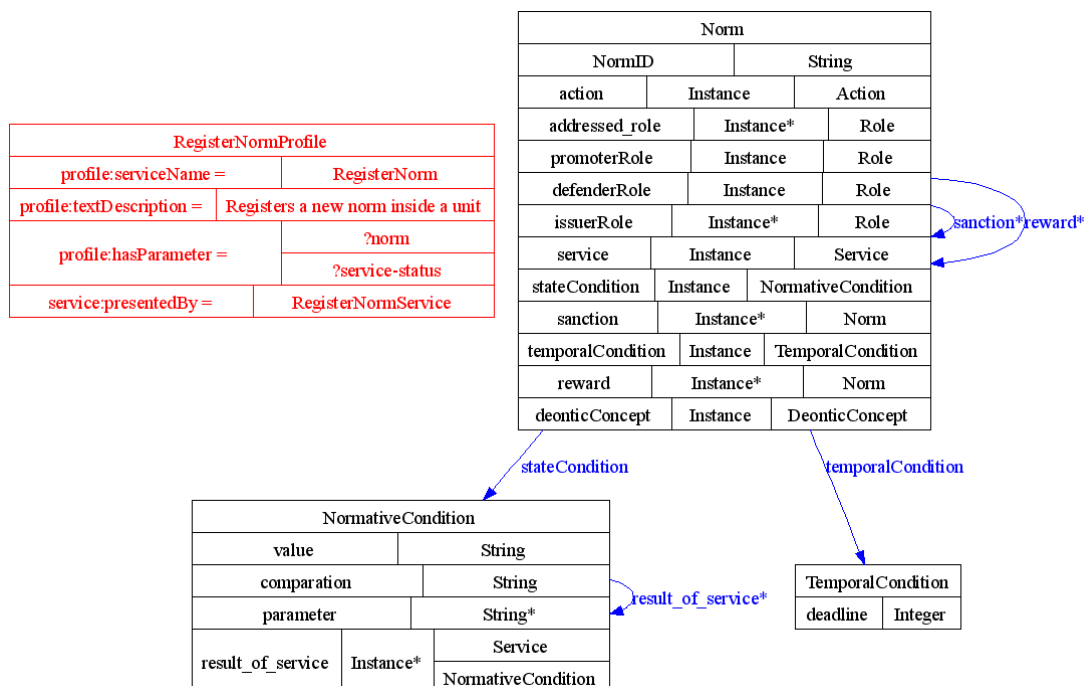
#### 1) RegisterRole

Se utiliza para solicitar el registro de un nuevo rol dentro de la unidad. Este servicio tiene como parámetro de entrada la instancia de la clase *Role* correspondiente al nuevo rol que se quiere registrar. El parámetro de salida *service-status* indica si la operación se ha realizado con éxito o en caso de producirse un error qué error se ha producido.



#### 2) RegisterNorm

Se utiliza para solicitar el registro de una nueva norma dentro de la unidad. Tiene como parámetros de entrada la instancia de la clase *Norm* correspondiente a la nueva norma. El parámetro de salida *service-status* indica si la operación se ha realizado con éxito o qué error se ha producido si la operación no se ha podido realizar.

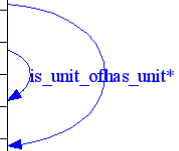


### 3) RegisterUnit

Se utiliza para solicitar el registro de una unidad vacía en el OMS. Recibe como parámetro de entrada la instancia de la unidad que se pretende crear y tiene como parámetro de salida una instancia de la clase *ServiceStatus* que indica el resultado de la ejecución del servicio.

RegisterUnitProfile	
profile:serviceName =	RegisterUnit
profile:textDescription =	Registers a new unit inside the organiza...
profile:hasParameter =	?unit ?service-status
service:presentedBy =	RegisterUnitService

OrganizationUnit		
UnitID	String	
has_goal	Instance*	OrganizationalGoal
plays	Instance	Role
defines_position	Instance*	DefinedRole
has_relationship	Instance*	hasRelationshipRelation
has_norm	Instance*	Norm Norm-Atom
has_domain_ontology	Instance*	Ontology
is_unit_of	Instance	OrganizationUnit
has_member	Instance*	Member
has_unit	Instance*	OrganizationUnit



### 4) DeregisterRole

Se utiliza para solicitar el desregistro de un rol dentro de la unidad. Este servicio tiene como argumento de entrada el nombre del rol que se pretende desregistrar (*roleID*), junto con el nombre de la unidad de la cual se quiere desregistrar el rol (*unitID*).

DeregisterRoleProfile	
profile:serviceName =	DeregisterRole
profile:textDescription =	Deregisters role from a unit
profile:hasParameter =	?unitID ?roleID ?service-status
service:presentedBy =	DeregisterRoleService

### 5) DeregisterNorm

Se utiliza para solicitar el desregistro de una norma. Tiene como input el nombre de la norma que se quiere desregistrar y como output una instancia de la clase *ServiceStatus* que indica el resultado de la operación.

DeregisterNormProfile	
profile:serviceName =	DeregisterNorm
profile:textDescription =	Deregisters a unit norm
profile:hasParameter =	?normID ?service-status
service:presentedBy =	DeregisterNormService

### 6) DeregisterUnit

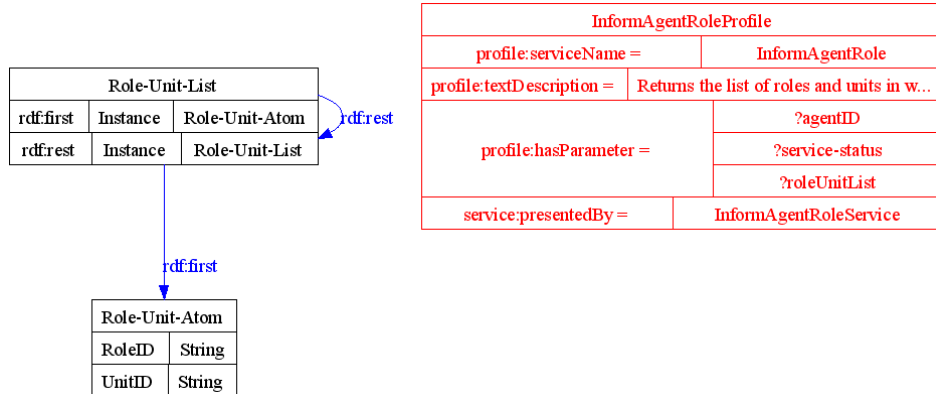
Se utiliza para solicitar el desregistro de una unidad en el OMS. Recibe como argumento de entrada un string con el nombre de la unidad que se quiere desregistrar. El argumento de salida service-status indica si el servicio se ha ejecutado con éxito o si se ha producido un error.

DeregisterUnitProfile	
profile:serviceName =	DeregisterUnit
profile:textDescription =	Deregisters a unit
profile:hasParameter =	?unitID ?service-status
service:presentedBy =	DeregisterUnitService

## InformationServices

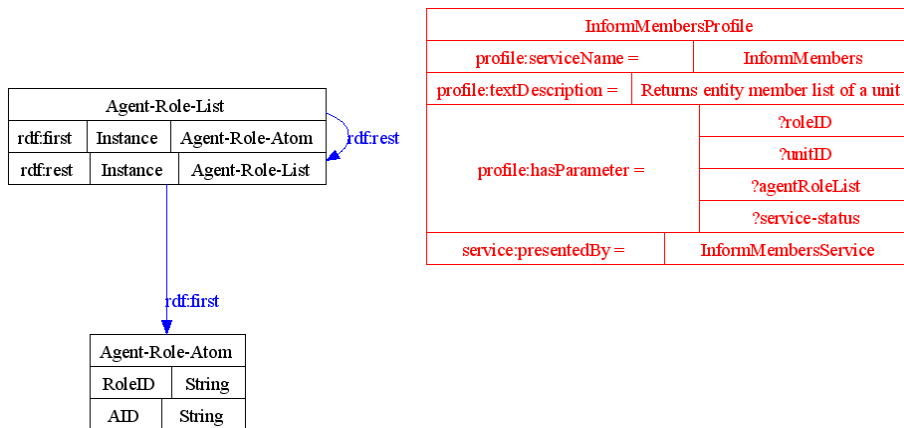
### 1) InformAgentRole

Se utiliza para solicitar el listado de roles y unidades en las que se encuentra un agente en un momento determinado. Recibe como argumento de entrada el identificador del agente correspondiente (*agentID*) y devuelve una instancia de la clase *ServiceStatus* indicando el estado de la ejecución del servicio y una lista con los pares (*RoleID, UnitID*). La clase *Role-Unit-List* hereda de la clase *rdf:List* (clase predefinida para la creación de listas en owl).



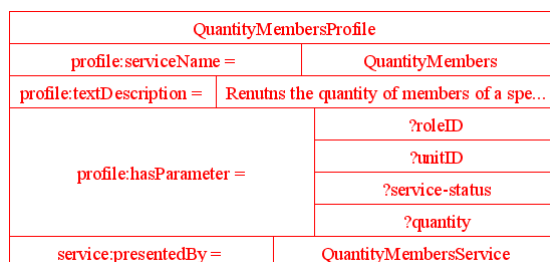
### 2) InformMembers

Se emplea para solicitar el listado de entidades miembro de una unidad determinada. De forma opcional se puede indicar un rol específico de esa unidad. Si no se indica ese rol, se devolverá el listado de todos los miembros, independientemente del rol que tengan asociado. Recibe como argumento de entrada un string con el nombre de la unidad (*unitID*), opcionalmente también puede recibir otro string con el nombre del rol (*roleID*), y devuelve una instancia de la clase *ServiceStatus* indicando el estado de la ejecución del servicio y una lista con los pares (Rol,AID).



### 3) QuantityMembers

Se emplea para solicitar la cantidad de miembros actuales de una unidad determinada. De forma opcional se puede indicar un rol específico de esa unidad, sobre el que se desea realizar el cálculo. Tiene como argumento de salida la variable *service-status*, que no indica el estado de la ejecución del servicio, y un entero que contendrá el número de agentes calculado por el servicio.



#### 4) InformUnit

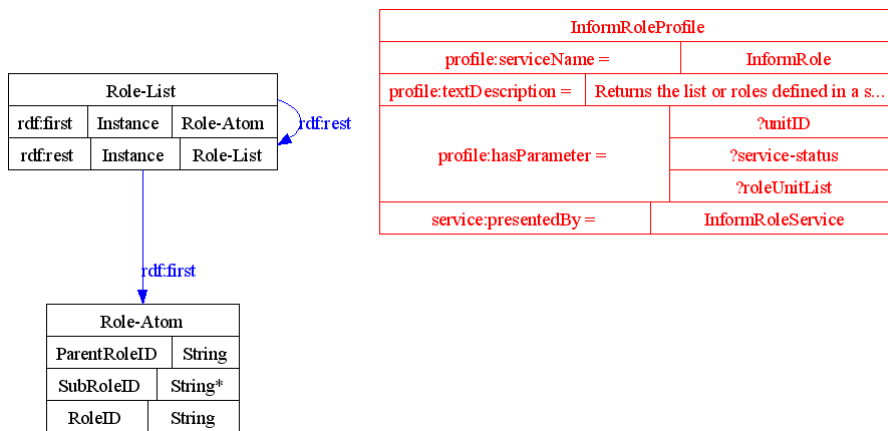
Se emplea para solicitar la información sobre una unidad concreta. Este servicio tiene como input un string con el nombre de la unidad (*unitID*). Como output tiene el estado de la ejecución y una instancia de la clase Unit-Atom. Esta clase contiene propiedades con información acerca de una unidad.

InformUnitProfile		
profile:serviceName =		InformUnit
profile:textDescription =	Returns the information of a unit	
profile:hasParameter =		?unitID
		?service-status
		?unitInformation
service:presentedBy =		InformUnitService

Unit-Atom		
ParentUnitID		String
UnitID		String
Type		String
has_goal	Instance*	OrganizationalGoal

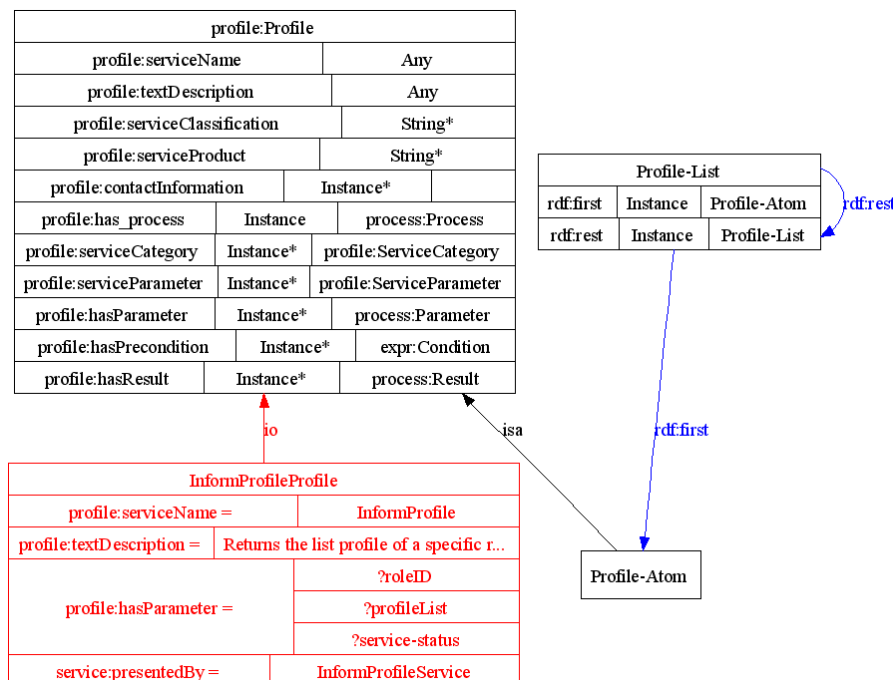
#### 5) InformRole

Se emplea para solicitar cuáles son los roles que existen en una unidad concreta. Devuelve una lista con la información sobre cada rol definido en la unidad.



#### 6) InformProfile

Se emplea para solicitar el listado de perfiles de los servicios asociados a un rol determinado (*roleID*). Devuelve una lista con las instancias de dichos perfiles.



## 7) InformNorm

Se emplea para solicitar el listado de normas asociado aun rol determinado. Devuelve una lista con la información sobre cada norma que afecta a dicho rol.

Norm-List		
rdf:first	Instance	Norm-Atom
rdf:rest	Instance	Norm-List

rdf:rest

rdf:first

Norm-Atom		
AdressedRoleID		String*
ServiceID		String
DefenderRoleID		String*
PromoterRoleID		String*
NormID		String
issuerRoleID		String
stateCondition	Instance*	NormativeCondition
temporalCondition	Instance*	TemporalCondition
deonticConcept	Instance	DeonticConcept
action	Instance	Action
norm_defined_in	Instance	OrganizationUnit

InformNormProfile	
profile:serviceName =	InformNorm
profile:textDescription =	Returns the list of norms that affects a ...
profile:hasParameter =	?roleID
	?normList
	?service-status
service:presentedBy =	InformNormService

## DynamicServices

### 1) RegisterAgentRole

Se utiliza para registrar una entrada en el listado EntityPlayList indicando que un agente determinado juega un rol específico en una unidad. Recibe como parámetros de entrada los strings correspondientes a los nombres del agente (*agentID*), unidad (*unitID*) y rol (*roleID*) correspondientes.

RegisterAgentRoleProfile	
profile:serviceName =	RegisterAgentRole
profile:textDescription =	Registers an agent playing a new role
profile:hasParameter =	?roleID
	?agentID
	?unitID
	?service-status
service:presentedBy =	RegisterAgentRoleService

### 2) DeregisterAgentRole

Se utiliza para desregistrar una entrada en el listado indicando que un agente determinado ya no juega un rol específico en una unidad.

DeregisterAgentRoleProfile	
profile:serviceName =	DeregisterAgentRole
profile:textDescription =	Registers that an agent has left a role
profile:hasParameter =	?agentID
	?roleID
	?unitID
	?service-status
service:presentedBy =	DeregisterAgentRoleService

### 3) AcquireRole

Se utiliza este servicio para solicitar la adquisición de un rol en una unidad determinada. Recibe como argumentos de entrada los nombres del rol y unidad correspondientes.

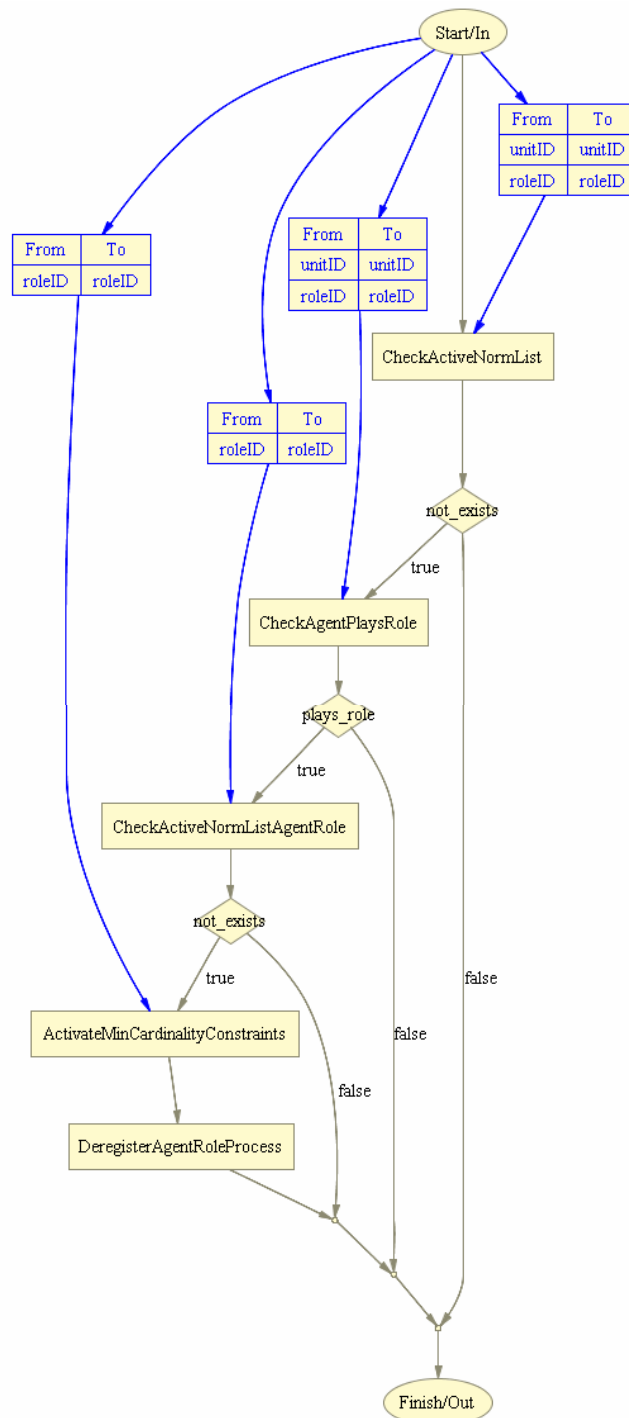
AcquireRoleProfile	
profile:serviceName =	AcquireRole
profile:textDescription =	Request role acquisition
profile:has_process =	AcquireRoleProcess
profile:hasParameter =	?unitID
	?roleID
	?service-status
service:presentedBy =	AcquireRoleService



#### 4) LeaveRole

Se utilizar para solicitar dejar de jugar un rol en una unidad determinada. Tiene como inputs los parámetros *unitID*, *roleID* que indican el nombre de la unidad y rol respectivamente.

LeaveRoleProfile	
profile:serviceName =	LeaveRole
profile:textDescription =	An agent leaves a role
profile:has_process =	LeaveRoleProcess
profile:hasParameter =	?unitID
	?roleID
	?service-status
service:presentedBy =	LeaveRoleService

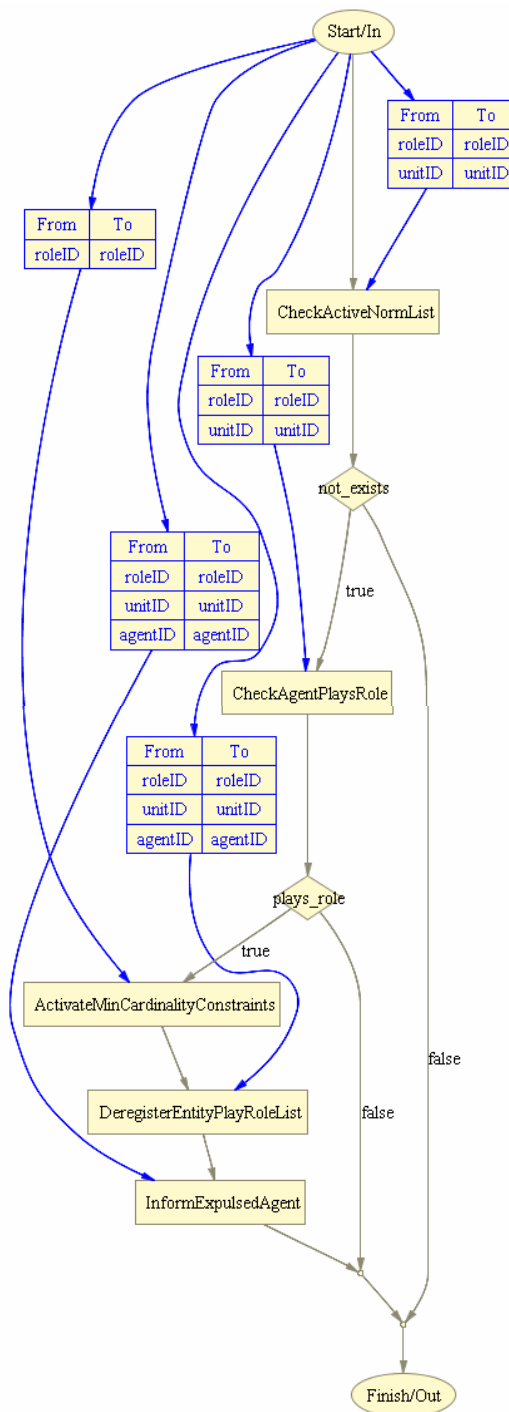




### 5) Expulse

Servicio para expulsar a un agente determinado, obligándolo a que deje la posición indicada. Recibe como argumentos de entrada los nombres de la unidad, agente y rol correspondientes.

ExpulseProfile	
profile:serviceName =	Expulse
profile:textDescription =	Requests the expulsion of an agent
profile:has_process =	ExpulseAgentProcess
profile:hasParameter =	?agentID
	?unitID
	?roleID
	?service-status
service:presentedBy =	ExpulseService



## Servicios SF

### 1) REGISTER PROFILE:

Se utiliza cuando un agente o servicio quiere registrar una descripción de servicio en el directorio para ello debe completar la descripción del servicio mediante la siguiente estructura:

<ServiceID, Providers, ServiceGoal, ServiceProfile>

Providers= <ProviderIDList, ServiceModel, ServiceGrounding>

SFServiceRegisterProfile	
profile:serviceName =	RegisterProfile
profile:hasParameter =	?ServiceID
	?ServiceProfile
	?ServicePurpose
	?ServiceStatus
service:presentedBy =	SFServiceRegisterProf

### 2) REGISTER PROCESS:

Se utiliza cuando un agente quiere registrar una implementación particular de un servicio. Debe indicarse el identificador de la entidad que oferta el servicio. Dado que pueden existir varios proveedores para una misma implementación de servicio. La primera vez que se invoca a RegisterProcess se indica un Proveedor (EntityID, que puede ser tanto una organización, o un agente). Los demás proveedores se añaden o modifican con llamadas a los servicios AddProvider y RemoveProvider.

SFServiceRegisterProcessProfile	
profile:hasParameter =	?ServiceModel
	?ServiceGrounding
	?ProviderID
	?ServiceStatus
service:presentedBy =	SFServiceRegisterProcess

### 3) DEREGISTER:

Se utiliza para eliminar una descripción de servicio del directorio.

SFServiceDeregisterProfile	
profile:serviceName =	Deregister
profile:hasParameter =	?ServiceID
	?ServiceStatus
service:presentedBy =	SFServiceDeregister

### 4) MODIFY PROCESS:

Se utiliza para modificar la descripción de la implementación de un servicio ya registrado. Un agente o un servicio indica la parte de la descripción del servicio que desee modificar siguiendo la estructura descrita al principio. El servicio mantendrá el mismo ID. Si hay más de un proveedor de la implementación del servicio, entonces no se podrá realizar un ModifyProcess.

SFServiceModifyProcProfile	
profile:serviceName =	ModifyProcess
profile:hasParameter =	?ProviderID
	?ServiceGrounding
	?ServiceModel
	?ServiceID
	?ServiceStatus
service:presentedBy =	SFServiceModifyProc

**5) MODIFY PROFILE:**

Se utiliza para modificar la descripción de un servicio ya registrado. Un agente o un servicio indica la parte de la descripción del servicio que desee modificar siguiendo la estructura descrita al principio. El servicio mantendrá el mismo ID.

SFServiceModifyProfProfile	
profile:serviceName =	ModifyProfile
profile:hasParameter =	?ServiceGoal
	?ServiceProfile
	?ServiceID
	?ServiceStatus
service:presentedBy =	SFServiceModifyProf

**6) SEARCH:**

Se utiliza para encontrar una descripción de servicio que satisfaga los requisitos que el cliente está buscando. En el proceso de búsqueda se pueden aplicar técnicas de matchmaking, composición de servicios, etc.

SFServiceSearchProfile	
profile:serviceName =	Search
profile:hasParameter =	?ServicePurpose
	?ServiceList
	?ServiceStatus
service:presentedBy =	SFServiceSearch

**7) SEARCH WHO:**

Se utiliza para encontrar el proveedor de un servicio determinado.

SFServiceSearchWhoProfile	
profile:serviceName =	SearchWho
profile:hasParameter =	?ServiceID
	?ProviderList
	?ServiceStatus
service:presentedBy =	SFServiceSearchWho

**8) ADD PROVIDER:**

Se utiliza para añadir un nuevo proveedor a una implementación de un servicio determinado.

SFServiceAddProviderProfile	
profile:serviceName =	AddProvider
profile:hasParameter =	?ServiceID
	?ProviderID
	?ServiceStatus
service:presentedBy =	SFServiceAddProvider

**9) REMOVE PROVIDER:**

Se utiliza para eliminar un proveedor de una implementación de un servicio determinado.

SFServiceRemoveProviderProfile	
profile:serviceName =	RemoveProvider
profile:hasParameter =	?ServiceID
	?ProviderID
	?ServiceStatus
service:presentedBy =	SFServiceRemoveProvider