



Diseño de una Arquitectura Satelital Segmentada basada en Sistemas Multiagente para la Gestión de Emergencias

Tesis presentada ante la Universidad Nacional de La Matanza y
la Unidad de Formación Superior de la Comisión Nacional de Actividades Espaciales
para la obtención del grado de
MAGISTER EN DESARROLLOS INFORMÁTICOS DE APLICACIÓN ESPACIAL
Por

Mg. Ezequiel González

Director

Dr. Jorge Ierache

UNLaM, Buenos Aires, Argentina

Co-Director

Dr. Ricardo Medel

UTN, Córdoba, Argentina

Julio, 2019

DEPARTAMENTO DE INGENIERÍA E INVESTIGACIONES TECNOLÓGICAS
UNIVERSIDAD NACIONAL DE LA MATANZA

—
UNIDAD DE FORMACIÓN SUPERIOR
COMISIÓN NACIONAL DE ACTIVIDADES ESPACIALES

©UFS-CONAE 2019

©UNLAM 2019

DEDICATORIA

*A Rocio y a Teo, por su apoyo incondicional
y por regalarme el bien más valioso: Tiempo.*

AGRADECIMIENTOS

A mi familia, por sus palabras de aliento.

*A mis directores de tesis, el Dr. Ricardo Medel y el Dr. Jorge Ierache,
por su predisposición, orientación y soporte.*

A Carlos Barrientos y a Sergio Masuelli, por su calidad humana y sus consejos.

A mis compañeros de la MDIAE, por los momentos compartidos.

A CONAE y a UNLaM, por darme la oportunidad de iniciar mi camino en el mundo espacial.

RESUMEN

En esta tesis se presenta la arquitectura de un sistema espacial distribuido basada en sistemas multiagente para el intercambio de servicios satelitales ociosos, tanto en tierra como en vuelo. El diseño abarca todos los elementos que forman parte de una misión satelital nominal, incluye satélites con cierto grado de autonomía para la planificación y ejecución de sus actividades, aborda el diseño interno de todos los agentes del sistema y su comunicación, y describe el concepto de operaciones para dos casos de aplicación relacionados con la gestión de emergencias. Asimismo, la arquitectura incorpora un elemento novedoso, una plataforma web para el intercambio de servicios satelitales en tierra, que permitiría contar con una industria espacial más colaborativa, más eficiente, y más accesible al público general.

Palabras clave: Sistemas Espaciales Distribuidos, Sistemas Satelitales Federados, Arquitecturas Segmentadas, Sistemas Multiagente, Autonomía, Economías Colaborativas, Arquitecturas Espaciales Innovadoras, Ingeniería de Sistemas Espaciales.

ABSTRACT

This thesis presents the architecture of a multiagent distributed space system for the exchange of unused satellite services, both in ground and flight segment. The design comprises all the elements that belongs to any nominal satellite mission, including satellites with a certain degree of autonomy for planning and execution activities, addresses the internal design of all system agents and their communication, and describes the operations concept for two cases related to emergency management. Also, the architecture includes a novel element, a web platform for the exchange of satellite services in ground, which would allow for a more collaborative and more efficient space industry, and more accessible to the general public.

Keywords: Distributed Space Systems, Federated Satellite Systems, Fractionated Spacecraft, Multiagent Systems, Autonomy, Sharing Economy, Innovative Space Architectures, Space Systems Engineering.

Tabla de Contenidos

1 INTRODUCCIÓN	1
1.1 Comentarios Preliminares	1
1.2 Aportes de la Tesis	2
1.3 Método de desarrollo de la Tesis	3
1.4 Paradigmas, Técnicas, Estándares y Herramientas utilizadas	4
1.5 Visión General de la Tesis	4
2 MARCO TEÓRICO	7
2.1 Elementos de una Misión Espacial	7
2.2 Sistemas Espaciales Distribuidos	9
2.2.1 Constelaciones	12
2.2.2 Trenes	12
2.2.3 Clusters	12
2.2.4 Enjambres	13
2.2.5 Arquitectura Segmentada	13
2.2.6 Satélites Federados	14
2.2.7 Internet en el Espacio	14
2.3 Inteligencia Artificial y Agentes Inteligentes	15
2.3.1 Introducción	15
2.3.2 Características de los Agentes	17
2.3.3 Tipos de Agentes	18
2.3.4 Entornos: Tipos y Características	20
2.3.5 Sistemas Multiagente	21
2.3.6 El modelo de referencia FIPA	21
2.4 Economías Colaborativas	23

3	ESTADO DEL ARTE	25
3.1	Sistemas Espaciales Distribuidos	25
3.1.1	Arquitectura Satelital Segmentada	25
3.1.2	Sistemas Satelitales Federados	27
3.2	Sistemas Multiagente en la industria Espacial	30
4	DESCRIPCIÓN DEL PROBLEMA	33
5	SOLUCIÓN – ARQUITECTURA DEL SISTEMA	35
5.1	Introducción al Sistema	35
5.1.1	Diseño Conceptual del Sistema	36
5.1.2	Product Tree	37
5.2	AgentSat	38
5.2.1	Plataforma del Satélite/AgentSat	38
5.2.2	AgentSat Payload	40
5.2.3	AgentSat Data & Service Manager	41
5.2.4	Platform Data & Services Manager	43
5.3	Agentes del AgentSat	44
5.3.1	Agente sat-Planner	44
5.3.2	Agente sat-Controller	48
5.3.3	Agente sat-Trader	50
5.3.4	Agente sat-Learner	52
5.3.5	Interfaces externas del AgentSat	53
5.4	Plataforma de Intercambio de Servicios Satelitales (SSEP)	54
5.5	Agentes del SSEP	57
5.5.1	Agente usr-Trader	57
5.5.2	Agente msc-Trader	58
5.5.3	Agente gst-Trader	60
5.5.4	Agente pvd-Trader	61
5.5.5	Agente Trade Manager	62
5.6	Plataformas Agente	63
5.6.1	Agent Management System	64
5.6.2	Service Directory Facilitator	64

5.6.3	Message Transport Service	64
5.7	Estaciones Terrenas (ET)	64
5.8	Caracterización de los Agentes del Sistema	66
5.8.1	Caracterización del Entorno para cada Agente	69
5.9	Comunicación entre los Agentes del Sistema	70
5.9.1	Protocolo de Interacción entre dos AgentSats en órbita	70
5.9.2	Protocolo de Interacción para Solicitud y Oferta de Servicios (2 agentes)	74
5.9.3	Protocolo de Interacción para Subasta inversa de Servicios (n agentes)	75
6	VALIDACIÓN DE LA ARQUITECTURA	77
6.1	Caso de Aplicación 1: Gestión de una Emergencia detectada en Órbita	77
6.1.1	Presentación del caso	77
6.1.2	Descripción detallada del caso	78
6.2	Caso de Aplicación 2: Soporte a una Emergencia detectada en Tierra	89
6.2.1	Presentación del caso	89
6.2.2	Descripción detallada del caso	90
7	CONCLUSIONES Y TRABAJO FUTURO	99
7.1	Conclusiones	99
7.2	Futuras líneas de Trabajo e Investigación	100
	REFERENCIAS	103

Índice de Figuras

Figura 1.1	Representación conceptual de varios tipos de SEDs	2
Figura 1.2	Diagrama de proceso correspondiente al desarrollo de la Tesis	3
Figura 2.1	Arquitectura nominal de una misión satelital	8
Figura 2.2	Clasificación de los Sistemas Espaciales Distribuidos	10
Figura 2.3	Clasificación de los SEDs según la precisión en la navegación	11
Figura 2.4	Esqueleto de un Agente que interactúa con su entorno	15
Figura 2.5	Resumen gráfico de Agentes: Características, Tipos y Entornos	16
Figura 2.6	Modelo de Referencia FIPA para la administración de Agentes	21
Figura 2.7	Sintaxis para el intercambio de mensajes ACL-FIPA	23
Figura 5.1	Raíces de la idea del SSEP	36
Figura 5.2	Diseño Conceptual del Sistema	36
Figura 5.3	Product Tree del Sistema	37
Figura 5.4	Arquitectura del AgentSat	39
Figura 5.5	Arquitectura del AgentSat Data & Service Manager (ADSM)	41
Figura 5.6	Arquitectura del Platform Data & Services Manager (PDSM)	44
Figura 5.7	Arquitectura del agente Planificador (sat-Planner)	45
Figura 5.8	Diagrama de Actividad del Proceso de Planificación del AgentSat	47
Figura 5.9	Arquitectura del agente Controlador (sat-Controller)	48
Figura 5.10	Arquitectura del agente Comercial (sat-Trader)	50
Figura 5.11	Arquitectura del agente Aprendiz (sat-Learner)	52
Figura 5.12	Esquema de las Interfaces Externas del AgentSat	53
Figura 5.13	Ontología de la Información intercambiada por el AgentSat	54
Figura 5.14	Arquitectura del SSEP	56
Figura 5.15	Arquitectura del agente comercial del Usuario (usr-Trader)	58

Figura 5.16	Arquitectura del agente comercial del MOC (msc-Trader)	59
Figura 5.17	Arquitectura del agente comercial de la Estación Terrena (gst-Trader)	60
Figura 5.18	Arquitectura del agente comercial del Proveedor de Servicios (pvd-Trader)	61
Figura 5.19	Arquitectura del agente administrador del comercio en el SSEP (Trade Manager)	62
Figura 5.20	Arquitectura de la Estación Terrena con la extensión para el Comercio de Servicios	66
Figura 5.21	Protocolo de Interacción IOSSE: Intercambio de servicios satelitales en órbita	72
Figura 5.22	Ejemplo de mensajes FIPA-ACL para el intercambio de estados de intercambio	73
Figura 5.23	Ejemplos de mensajes FIPA-ACL: AgentSat i ofreciendo servicios satelitales en órbita	73
Figura 5.24	Ejemplos de mensajes FIPA-ACL: AgentSat j aceptando y rechazando servicios ofrecidos	74
Figura 5.25	Protocolos FIPA-Request y FIPA-Propose: Solicitud y Oferta de servicio en el SSEP	75
Figura 5.26	Protocolo FIPA-ContractNet: Llamado a propuestas para Solicitud de servicio en el SSEP	76
Figura 6.1	Esquema: Gestión de una Emergencia detectada en órbita	78
Figura 6.2	Extracto de la Arquitectura del AgentSat resaltando las cajas activadas durante el Paso 1	80
Figura 6.3	Asociación entre los subpasos del Paso 1 y las unidades internas del sat-Planner	81
Figura 6.4	Diagrama de Actividad: Planificación a bordo de Recursos Ociosos	81
Figura 6.5	Asociación entre los subpasos del Paso 2 y las unidades del sat-Controller	82
Figura 6.6	Asociación entre los subpasos del Paso 3 y las unidades internas del sat-Trader	83
Figura 6.7	Diagrama de Actividad: Gestión de Intercambio de Servicios en Órbita	84
Figura 6.8	Protocolo de Interacción y Mensaje con la información del Incendio Detectado	84
Figura 6.9	Asociación entre los subpasos del Paso 4 y las unidades del sat-Trader y sat-Controller	85
Figura 6.10	Diagrama de Actividad: Ejecución del Servicio de Mensajería a Tierra.	86
Figura 6.11	Unidades de la ET involucradas en la recepción y transmisión del mensaje a la Plataforma	86
Figura 6.12	Asociación entre los subpasos del Paso 6, 8 y 9, y las unidades internas del Trade Manager	87
Figura 6.13	Asociación entre los subpasos del Paso 7 y las unidades internas del usr-Trader	88
Figura 6.14	Diagrama de Actividad: Gestión de un mensaje generado en órbita dirigido a un Usuario.	89
Figura 6.15	Esquema: Soporte a una Emergencia detectada en Tierra	90
Figura 6.16	Asociación entre los subpasos del Paso 2 y las unidades internas del usr-Trader	91
Figura 6.17	Asociación entre los subpasos del Paso 4 y las unidades internas del msc-Trader	92

Figura 6.18	Asociación entre los subpasos del Paso 4 y las unidades internas del Trade Manager	93
Figura 6.19	Asociación entre los subpasos del Paso 5 y las unidades internas del GSt Agent Extension	93
Figura 6.20	Asociación entre los subpasos del Paso 6 y las unidades internas del AgentSat	94
Figura 6.21	Asociación entre los subpasos del Paso 6 y las unidades internas del AgentSat Data & Service Manager y el sat-Controller	95
Figura 6.22	Asociación entre los subpasos del Paso 7 y las unidades internas del GSt Agent Extension	95
Figura 6.23	Asociación entre los subpasos del Paso 8 y las unidades internas del SSEP	96
Figura 6.24	Asociación entre los subpasos del Paso 9 y las unidades internas del SSEP	97

Índice de Tablas

Tabla 2.1	Análisis comparativo de las distintas arquitecturas SED	11
Tabla 5.1	Modos del Servicio de Intercambio del AgentSat (TradeServiceMode)	42
Tabla 5.2	Modos de Autonomía del AgentSat (AutonomyMode)	43
Tabla 5.3	Estados del AgentSat (AgentSatStatus)	49
Tabla 5.4	Estados de la Orden de Intercambio (TradeOrderStatus)	51
Tabla 5.5	Caracterización del agente AgentSat	66
Tabla 5.6	Caracterización del agente sat-Planner	67
Tabla 5.7	Caracterización del agente sat-Trader	67
Tabla 5.8	Caracterización del agente sat-Controller	67
Tabla 5.9	Caracterización del agente sat-Learner	68
Tabla 5.10	Caracterización de los agentes Traders del SSEP (usr, msc, gst y pvd)	68
Tabla 5.11	Caracterización del agente Trade Manager del SSEP	68
Tabla 5.12	Caracterización del entorno de acuerdo a cada uno de los agentes del sistema	69
Tabla 5.13	Conjunto de combinaciones de estados individuales que generan ‘Match’ potencial	71
Tabla 6.1	Esquema del Plan elaborado para el caso de aplicación	79

Lista de Acrónimos

ABM	Altas, Bajas y Modificaciones
ACC	Agent Communication Channel (Canal de Comunicación de Agentes)
ACL	Agent Communication Language (Language de Comunicación de Agentes)
ACM	ACcounts Manager (Administrador de Cuentas)
ADMS	Agentsat Data and Service Manager (Gestor del Servicio y de los Datos del Agentsat)
AI	Artificial Intelligence (Inteligencia Artificial)
AMM	Agentsat Mass Memory (Memoria Masiva del Agentsat)
AMS	Agent Management System (Sistema de Gestión de Agentes)
AP	Agent Platform (Plataforma Agente)
ASS	Arquitectura Satelital Segmentada (en inglés: Fractionated Spacecraft)
AUM	AUtonomy Manager (Gestor de la Autonomía)
CDH	Command and Data Handling subsystem (Subsistema de Comando y Manejo de Datos)
CA	Communicative Act (Actos Comunicativos)
CONAE	COMisión Nacional de Actividades Espaciales
COTS	Commercial Off-The-Shelf products (productos Comerciales sacados del estante)
CSC	Current State Calculator (Calculador del Estado Actual)
DARPA	Defense Advanced Research Projects Agency
DMM	Data and Messages Manager (Gestor de Datos y Mensajes)
EO	Earth Observation (Observación de la Tierra)
ESA	European Space Agency (Agencia Espacial Europea)
ESM	Event and Status Monitor (Monitor de Eventos y Estados)
ET	Estación Terrena
FAP	Flight Agent Platform (Plataforma Agente de Vuelo)
FIPA	Foundation for Intelligent Physical Agents

FOS	Federation Of Systems (Federación de Sistemas)
FSE	Future State Estimator (Estimador del Estado Futuro)
GAP	Ground Agent Platform (Plataforma Agente de Tierra)
GSDH	Ground Station Data Handler (Manejador de Datos de la Estación Terrena)
GSS	Ground Station Services (Servicios de la Estación Terrena)
IDH	Instrument Data Handler (Manejador de los Datos de los Instrumentos)
IMM	IMage Manager (Gestor de Imágenes)
IOSSE	In Orbit Satellite Services Exchange (Intercambio de Servicios Satelitales en Órbita)
IP	Interaction Protocol (Protocolo de Interacción)
ISL	Inter-Satellite Links (Enlaces Intersatelitales)
ISLP	Inter-Satellite Links Payload (Carga útil para Enlaces Intersatelitales)
ISS	International Space Station (Estación Espacial Internacional)
LEO	Low Earth Orbit (Órbita Terrestre Baja)
MANET	Mobile Ad-hoc NETworking Techniques (Técnicas específicas de redes móviles)
MC	Mission Center (Centro de Misión)
MCDM	Mission Center Data Manager (Gestor de Datos del Centro de Misión)
MCS	Mission Center Services (Servicios del Centro de Misión)
MCTM	Mission Center Trade Manager (Gestor de Intercambios del Centro de Misión)
MDC	Mission Data Center (Centro de Datos de Misión)
MDH	Market Data Handler (Manejador de los Datos de Mercado)
MDM	Market Data Manager (Gestor de Datos de Mercado)
MKR	MarKet Researcher (Analista de Mercado)
MOC	Mission Operation Center (Centro de Operación de Misión)
MRH	Market Reports Handler (Manejador de Reportes de Mercado)
MTP	Message Transport Protocol (Protocolo de Transporte de Mensajes)
MTS	Message Transport Service (Servicio de Transporte de Mensajes)
MUM	Mission Utility Maximizer (Maximizador de Utilidad de Misión)
NASA	National Aeronautics and Space Administration (Agencia Espacial de Estados Unidos)
OBC	OnBoard Computer (Computadora de a bordo)
ODB	Orders DataBase (Base de datos de la Órdenes)
ORM	ORders Manager (Gestor de Órdenes)

ORS	ORders Scheduler (Programador de Órdenes)
P2P	Peer-To-Peer (Entre pares o entre iguales)
PDH	Platform Data Handler (Manejador de Datos de Plataforma)
PDSM	Platform Data & Services Manager (Gestor de Servicios y Datos de Plataforma)
PEA	Power Estimation Apprentice (Aprendiz de Estimación de Potencia)
PEX	Plan EXecutor (Ejecutor de Planes)
PKI	Public Key Infrastructure (Infraestructura de Clave Pública)
POM	Profile and Orders Manager (Gestor de Perfil y Órdenes)
PRH	Platform Requests Handler (Manejador de Solicitudes a Plataforma)
PRM	PRoduct Manager (Gestor de Productos)
PVS	ProVider Services (Servicios de los Proveedores)
ROI	Return On Investment (Retorno de la Inversión)
RPH	RePort Handler (Manejador de Reportes)
RPM	RePorting Manager (Administrador de Informes)
RTM	RaTing Manager (Administrador de Calificaciones)
RTP	Real-Time Propagator (Propagador de Tiempo Real)
SatSES	SATellite Services Exchange System (Sistema de Intercambio de Servicios Satelitales)
SDF	Service Directory Facilitator (Facilitador del Directorio de Servicios)
SDR	Software-Defined Radio (Radio Definida por Software)
SED	Sistema Espacial Distribuido
SMA	Sistemas Multi-Agente (en inglés: Multi-Agent Systems)
SoS	Systems-of-Systems (Sistemas de Sistemas)
SSEP	Satellite Services Exchange Platform (Plataforma de Intercambio de Servicios Satelitales)
SSO	Sun-Synchronous Orbit (Órbita heliosincrónica)
SSF	Sistemas Satelitales Federados (en inglés: Federated Space Systems)
SVM	SerVice Manager (Gestor del Servicio)
TLE	Two-Line Element set (Conjunto de elementos -orbitales- de dos líneas)
TM	Trade Manager (Administrador del Comercio)
TRM	TRansactions Manager (Administrador de Transacciones)
TSM	Trade Service Manager (Administrador del Servicio de Intercambio)
TTC	Telemetry, Tracking and Command subsystem (Subsistema de Telemetría, Seguimiento y Comando)

TT-TC	Time-Tagged TeleCommands (Telecomandos programados en tiempo)
UNSAM	Universidad Nacional de SAn Martin
USS	USer Services (Servicios del Usuario)
VSM	Virtual Satellite Mission (Misión Satelital Virtual)
XDH	eXtension Data Handler (Manejador de Datos de la Extensión)
XMH	eXtension Message Handler (Manejador Mensajes de la Extensión)

Capítulo 1

INTRODUCCIÓN

En este capítulo se presentan algunos comentarios preliminares sobre la arquitectura diseñada (sección 1.1), se identifican los aportes de la tesis (sección 1.2), se describen los métodos y el proceso llevado a cabo durante el transcurso del trabajo (sección 1.3), se enumeran las metodologías empleadas (sección 1.4) y se presenta la estructura y organización general de la tesis (sección 1.5).

1.1 Comentarios preliminares

El propósito inicial de la tesis fue diseñar una arquitectura satelital segmentada (ASS) basada en sistemas multiagente (SMA) que sea de utilidad para la gestión de emergencias ambientales. En este sentido, la tesis nacía del entrecruzamiento de tres elementos (SMA, ASS y gestión de emergencias) que no habían sido sintetizados aún en la literatura científica.

Luego de la revisión detallada de la literatura llevada a cabo durante el desarrollo de la tesis, se llegó a la conclusión de que era más ventajoso emplear otro tipo de sistema espacial distribuido (SED) como base para la arquitectura propuesta: los sistemas satelitales federados (SSF). Estos pueden concebirse como una generalización de las arquitecturas segmentadas, en donde los satélites en órbita no necesariamente son operados o administrados por la misma organización (figura 1.1). Aunque la descripción detallada de las características, semejanzas y diferencias entre las ASS y los SSF se presentan en el capítulo 2, vale la pena dejar claro lo siguiente:

- En lo que respecta a esta tesis en particular, los SSF son más apropiados que las ASS para la gestión de emergencias ya que involucra potencialmente una infinidad mayor de satélites en órbita, resultando en una mayor cobertura, menor latencia (Golkar y Lluch, 2015) y mayor actualización en la información provista a los usuarios finales.

- La comunicación entre los segmentos de una ASS y los satélites de un SSF se basa en el mismo principio: en la utilización de componentes que permitan establecer enlaces intersatelitales (ISL, por sus siglas en inglés “*Inter-Satellite Links*”) en órbita. La única diferencia es que en el último caso las comunicaciones son oportunísticas, y, por ende, más intermitentes.
- Este pequeño ajuste de aproximación no implica de ninguna manera un cambio en la naturaleza de la tesis ni una reducción del alcance de la misma. Por el contrario, no solo la hace más ventajosa para la aplicación en cuestión, sino que incorpora un nuevo elemento a la arquitectura: una plataforma para el intercambio de servicios satelitales en tierra (SSEP).

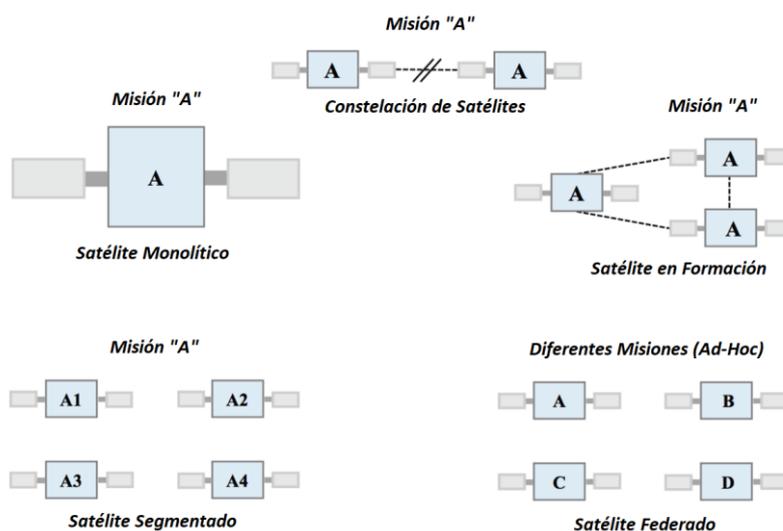


Figura 1.1 – Representación conceptual de varios tipos de SEDs (adaptado de [Poghosyan et al., 2016a])

En lo que refiere al paradigma multiagente, se afirmó a lo largo del trabajo la idea de que los SMA son una perfecta aproximación para el diseño de la tesis debido a (i) la naturaleza distribuida del sistema satelital propuesto y a (ii) la existencia de múltiples actores, con distintos objetivos, interactuando en un entorno en común, ya sea el espacio para el segmento de vuelo o el SSEP (*Satellite Services Exchange Platform* o Plataforma de Intercambio de Servicios Satelitales) en tierra.

1.2 Aportes de la Tesis

Los aportes de la tesis se enumeran a continuación:

- Un diseño de una arquitectura satelital que incluye todos los *stakeholders* de una misión tipo y, que, al mismo tiempo, llega a nivel de detalle en lo que refiere al diseño interno de los agentes de software.
- Una aproximación al diseño que resulta de una combinación entre la ingeniería de sistemas y el paradigma de sistemas multiagente.
- La incorporación de un elemento novedoso dentro de la arquitectura (el SSEP) que haría posible la explotación comercial colaborativa de recursos satelitales a través de una única plataforma web.

- Una descripción detallada del concepto de operaciones del sistema en términos de caso de aplicación particular a un dominio de gestión de emergencias ambientales a fin de validar la arquitectura propuesta.

1.3 Método de desarrollo de la Tesis

Los métodos empleados durante el desarrollo de la tesis fueron:

- *Revisión sistemática de bibliografía:* Las revisiones sistemáticas de artículos científicos siguen un método explícito para resumir la información sobre determinado tema o problema. Se diferencia de las revisiones narrativas en que provienen de una pregunta estructurada y de un protocolo previamente elaborado.
- *Prototipado evolutivo experimental (método de la Ingeniería):* El prototipado evolutivo experimental (Basili, 1993) consiste en desarrollar una solución inicial para un determinado problema, generando su refinamiento de manera evolutiva por prueba de aplicación de dicha solución a casos de estudio (problemáticas) de complejidad creciente. El proceso de refinamiento concluye al estabilizarse el prototipo en evolución.

La figura 1.2 resume el proceso llevado a cabo para el desarrollo de la tesis. En él se observan los ajustes en el diseño mencionados en la sección 1.1 e identifica los entregables (capítulos) por cada actividad realizada.

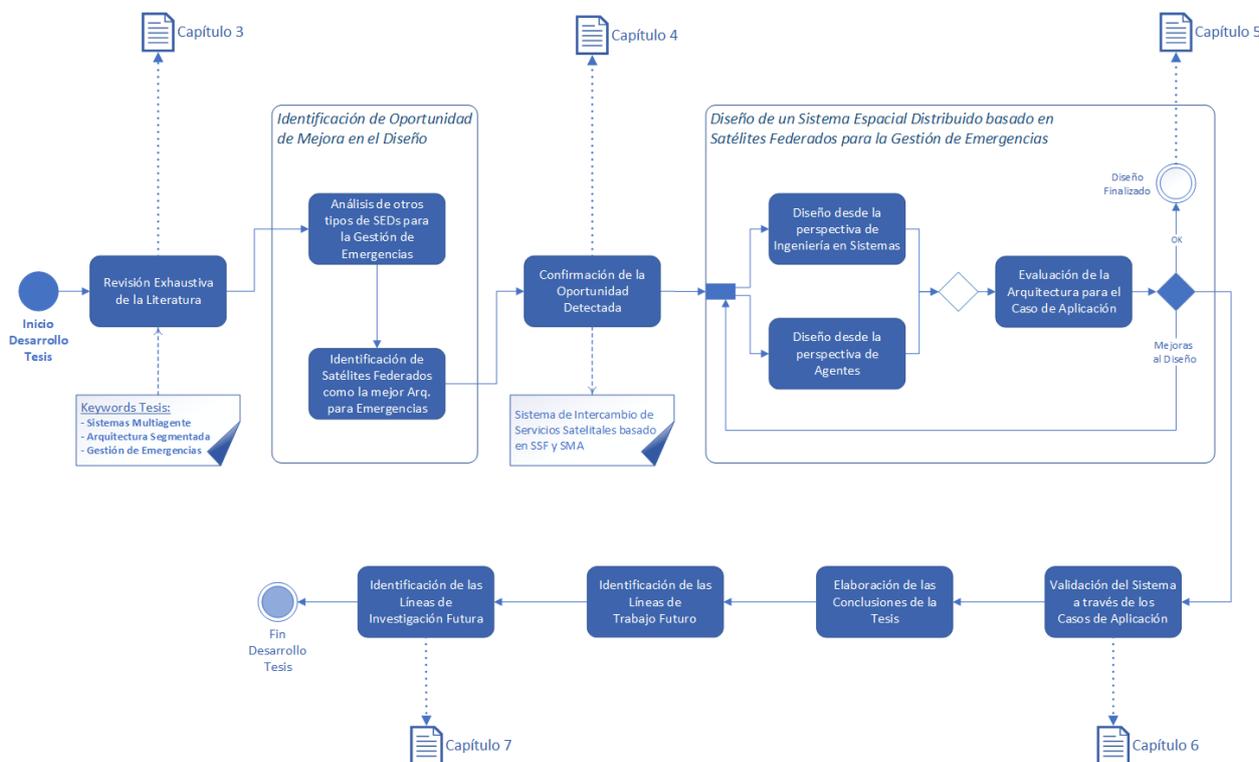


Figura 1.2 – Diagrama de proceso correspondiente al desarrollo de la Tesis

1.4 Paradigmas, Técnicas, Estándares y Herramientas utilizadas

Los paradigmas abordados a lo largo de la tesis fueron:

- Sistemas Multiagente
- Arquitectura Satelital Segmentada
- Sistemas Satelitales Federados

Las técnicas empleadas para el desarrollo de la tesis fueron:

- Lenguaje para el modelado de sistemas: *SysML* (Object Management Group, 2017)
- Técnicas para el modelado de agentes: *AUML* (Odell *et al.*, 2000)

Los estándares empleados para el desarrollo de la tesis fueron:

- Estándar para la comunicación entre Agentes: *FIPA* (FIPA, 2002a)

Las herramientas utilizadas para el desarrollo de la tesis fueron:

- Herramienta para el modelado de sistemas: *Enterprise Architect*
- Herramientas para la redacción de la tesis: *LaTeX*, *LibreOffice* y *Microsoft Word*
- Herramienta para la elaboración de figuras: *Microsoft Visio*
- Herramientas para la elaboración de tablas: *Microsoft Excel*

1.5 Visión General de la Tesis

El contenido del trabajo se estructura del siguiente modo. Hasta aquí se han presentado los comentarios preliminares sobre el diseño, se han enumerado los aportes de la tesis, se ha descrito el método de trabajo llevado a cabo durante el desarrollo de la tesis y se han mencionado los paradigmas, técnicas, estándares y herramientas empleados. La estructura organizativa del resto de la tesis se presenta a continuación.

En el capítulo 2 se brinda una introducción a los conceptos que serán utilizados a lo largo de la tesis. En primer lugar, se da una breve introducción a la ingeniería espacial, en donde se mencionan los distintos tipos de misiones, se presentan los elementos básicos que componen una misión espacial, se muestra la arquitectura típica de una misión satelital y los subsistemas de un satélite (sección 2.1). A continuación, se introduce el concepto de sistema espacial distribuido (SED), se enumeran las distintas configuraciones existentes, se los clasifica, se los describe brevemente y se presenta el tema de Internet en el espacio (sección 2.2). Luego se presenta una reseña del concepto de agente, se abordan las distintas clases de agentes y de entornos, se definen los sistemas multiagente y se presenta el modelo de referencia FIPA que es el elegido

para la comunicación de los agentes en la tesis (sección 2.3). Por último, se introduce el concepto de economía colaborativa (sección 2.4), ya que sobre él se apoya la arquitectura de SSF.

En el capítulo 3 se elabora una revisión bibliográfica de los tópicos centrales de la tesis. Luego de presentar algunas referencias relacionadas con los SEDs en general (sección 3.1), se elabora una revisión de los trabajos relacionados con la arquitectura segmentada (inciso 3.1.1). A continuación, se presenta una revisión detallada de la bibliografía que trata acerca de los sistemas satelitales federados, último tipo SED que ha surgido en la literatura (inciso 3.1.2). Finalmente, se presenta una reseña de aquellas aplicaciones del paradigma multiagente en la industria espacial (sección 3.2).

En el capítulo 4 se describe el problema en términos de la oportunidad detectada durante el proceso de análisis de la literatura. Aquí se menciona el ajuste en el diseño disparado por esta revisión y se enumeran algunos aspectos que aún no han sido abordados por la literatura científica actual.

En el capítulo 5 se describe la arquitectura del Sistema de Intercambio de Servicios Satelitales o SatSES. En primer lugar, se introduce conceptualmente la arquitectura y se presenta el árbol del producto (sección 5.1). Luego se detalla el segmento de vuelo, describiendo la arquitectura del AgentSat (sección 5.2) y el diseño interno de los agentes que lo componen (sección 5.3). Posteriormente se presenta la plataforma de intercambio de servicios satelitales o SSEP (sección 5.4) y el diseño interno de sus agentes (sección 5.5). A continuación, se presentan las características de las plataformas agente, tanto de vuelo como de tierra (sección 5.6) y se presenta la arquitectura de las estaciones terrenas que se suscribirán al SSEP (sección 5.7). Finalmente, se caracterizan todos los agentes del sistema (sección 5.8) y se aborda la comunicación entre los agentes (sección 5.9).

En el capítulo 6 se valida el sistema usando dos casos de aplicación relacionados con emergencias. Primero se detalla el concepto de operación de un ejemplo de gestión de una emergencia que ha sido detectada en órbita (sección 6.1) y luego se replica el procedimiento con un ejemplo de uso del sistema para dar soporte a una emergencia detectada en tierra (sección 6.2).

Finalmente, en el capítulo 7 se presentan las conclusiones del trabajo (sección 7.1) y se enuncian las líneas de trabajo e investigación futura que se desprenden de la tesis (sección 7.2).

Capítulo 2

MARCO TEÓRICO

En este capítulo se describen los conceptos que sentarán las bases teóricas de la presente tesis. Primeramente, se introducen los elementos básicos de una misión espacial (sección 2.1), luego se presenta una reseña sobre los sistemas espaciales distribuidos (sección 2.2) y, por último, se introduce el concepto de agente, de sistemas multiagente y se presenta el modelo de referencia FIPA (sección 2.3).

2.1 Elementos de una Misión Espacial

Una misión espacial puede definirse como todo proyecto cuyo propósito sea lanzar al espacio un objeto construido por el hombre (tripulado o no) para cumplir con un determinado objetivo, normalmente adquirir información científica. El artefacto lanzado puede ser un satélite, una sonda espacial, un *rover* o una nave tripulada; y el destino final será una trayectoria suborbital, la órbita terrestre, el sistema solar o el espacio interestelar. Aunque los objetivos de una misión espacial pueden ser muy variados, pueden resumirse en la siguiente clasificación:

- I. Científicas
 - a. Observación de la Tierra (ej. *Landsat program, SAC-D Aquarius*)
 - b. Astronomía (ej. *Hubble Space Telescope*)
 - c. Exploración del Sistema Solar (ej. *Mars Science Laboratory, conocida como Curiosity*)
 - d. Exploración del Espacio Profundo (ej. *New Horizons*)
- II. Pruebas Tecnológicas (ej. *Deep Space 1*)
- III. Comunicaciones (ej. *Iridium*)
- IV. Posicionamiento Satelital (ej. *GPS, Galileo*)
- V. Defensa e Inteligencia (ej. *USA-241*)

De todas formas, más allá del objetivo y el tipo, toda misión espacial está compuesta por los siguientes 4 elementos básicos:

- Segmento de vuelo
- Lanzador
- Segmento de tierra
- Usuarios

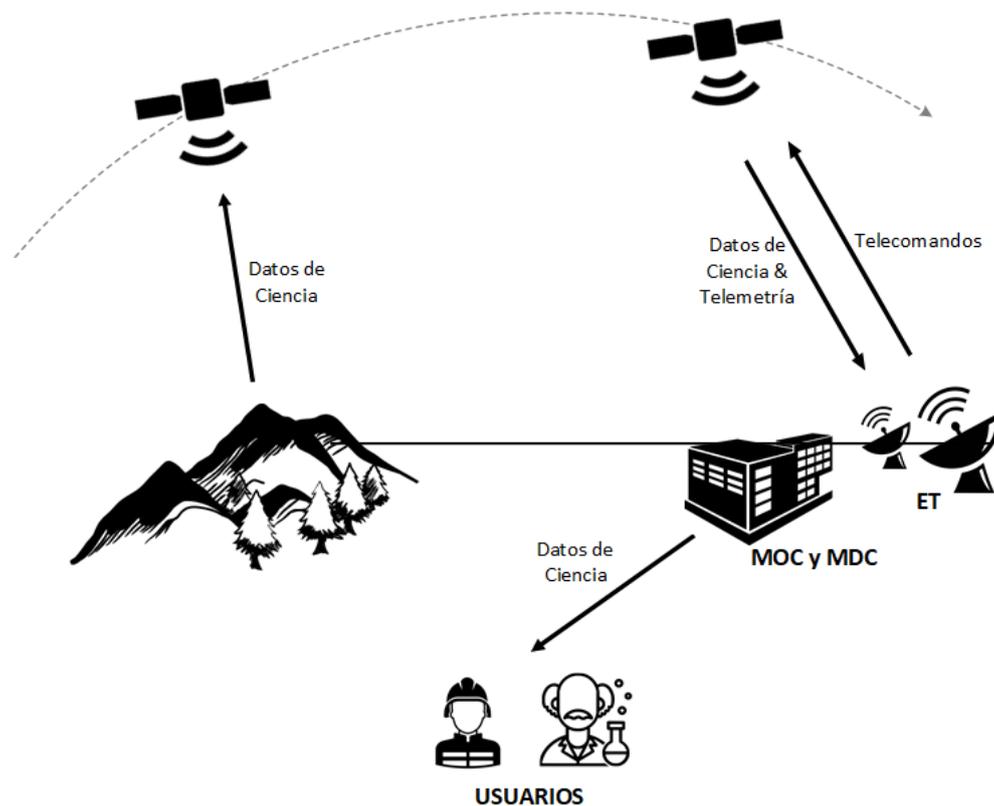


Figura 2.1 – Arquitectura nominal de una misión satelital

Un esquema general del concepto de operación de una misión satelital tipo se presenta en la figura 2.1

El segmento de vuelo se compone por el o los satélites que integran la misión. El término satélite es el que aplica a la mayoría de los elementos de la clasificación presentada, sin embargo, en lo referente a misiones de exploración del espacio, las expresiones vehículos espaciales, sondas espaciales o naves tripuladas serían las más apropiadas. Los dos elementos principales de un satélite son: la plataforma y su carga útil. La carga útil es la razón de ser de la misión y se compone de uno o varios instrumentos, de acuerdo a las necesidades de la misión. Por otro lado, la plataforma es la responsable de la operación general del satélite y de proveer los servicios requeridos por los instrumentos (potencia, cómputo, control térmico, etc.). La plataforma está compuesta por 7 subsistemas (Wertz *et al.*, 2011; Fortescue *et al.*, 2003), a saber:

- Subsistema Control de Actitud, *para controlar la orientación del satélite*
- Subsistema de Propulsión, *para darle impulso al satélite,*

- Subsistema de Generación de Potencia, *para permitir el funcionamiento del satélite,*
- Subsistema de Control Térmico, *para mantener la temperatura en los rangos deseados,*
- Subsistema de Comando y Manejo de Datos, *para darle inteligencia al satélite,*
- Subsistema de Comunicaciones, *para permitir la comunicación con tierra o con otros satélites,*
- Subsistema de Estructura, *para darle soporte mecánico al satélite.*

Cabe mencionar que no todos los satélites incluyen los siete subsistemas. Por ejemplo, una gran parte de los pequeños satélites (en particular, los *CubeSats*) no tienen propulsión. Pero más allá de los casos particulares, conceptualmente todos los satélites contemplan los subsistemas mencionados.

Por otro lado, el segmento de tierra está integrado por (i) las estaciones terrenas (ET), (ii) el centro de operación de misión (MOC, por sus siglas en inglés Mission Operation Center) y (iii) el centro de datos de misión (MDC, por sus siglas en inglés Mission Data Center). Las ET son la interfaz entre el segmento de vuelo y el de tierra ya que a través de ellas se lleva a cabo la comunicación entre el MOC y el satélite. El MOC es el responsable de llevar a cabo la planificación de las pasadas, de comandar el satélite (enviar telecomandos), monitorear su estado de salud y descargar los datos de ciencia y de telemetría. Por último, el MDC se ocupa del procesamiento de los datos de ciencia adquiridos por los instrumentos para su posterior publicación y/o entrega a los usuarios de la misión.

El lanzador, vector o vehículo de lanzamiento es un segmento de transferencia (Ley *et al.*, 2009) ya que es el responsable de transportar y poner en órbita el satélite y su carga útil.

Por último, los usuarios o el segmento de usuario de la misión, son todos aquellos destinatarios que consumirán la información generada por la misión. Los usuarios pueden ser de todo tipo: las agencias espaciales, la comunidad científica, los gobiernos nacionales, empresas privadas, instituciones o cualquier persona interesadas en los productos generados por la misión.

2.2 Sistemas Espaciales Distribuidos

Un sistema espacial distribuido (SED) es aquel que asigna funcionalidad entre múltiples satélites que interactúan entre sí para alcanzar los objetivos deseados (Selva *et al.*, 2017). Aunque hay más de una definición todas tienen una característica en común: la cooperación entre los distintos elementos. De acuerdo al Goddard Space Flight Center de la NASA, un SED se define como un sistema *end-to-end* que incluye uno o más vehículos espaciales y una infraestructura cooperativa para mediciones científicas, adquisición de datos, procesamiento, análisis y distribución (Leitner, 2002). Por otra parte, Poghosyan *et al.* (2016a) hacen hincapié en la complejidad y las propiedades emergentes de los SED en su definición, describiéndolos como arquitecturas de misión que consisten en múltiples elementos espaciales que interactúan, cooperan y se comunican entre sí, y que generalmente resultan en nuevas propiedades del sistema y/o funciones

emergentes. Por último, D'Amico *et al.* (2015) conciben a los SEDs de un modo más genérico, similar a la primera definición: como aquel sistema espacial que involucra la interacción de dos o más satélites para cumplir con el objetivo de la misión.

En la figura 2.2 se introduce una clasificación de los SED y una representación gráfica de alto nivel para cada uno de los distintos tipos. Como parte de un análisis comparativo de las distintas arquitecturas de los SED, Poghosyan *et al.* (2016a) analizaron y evaluaron cinco dimensiones para cada una de sus elementos: (i) objetivos, (ii) cooperación, (iii) homogeneidad, (iv) distancia e (v) independencia. El resultado de su estudio se describe en la tabla 2.1.

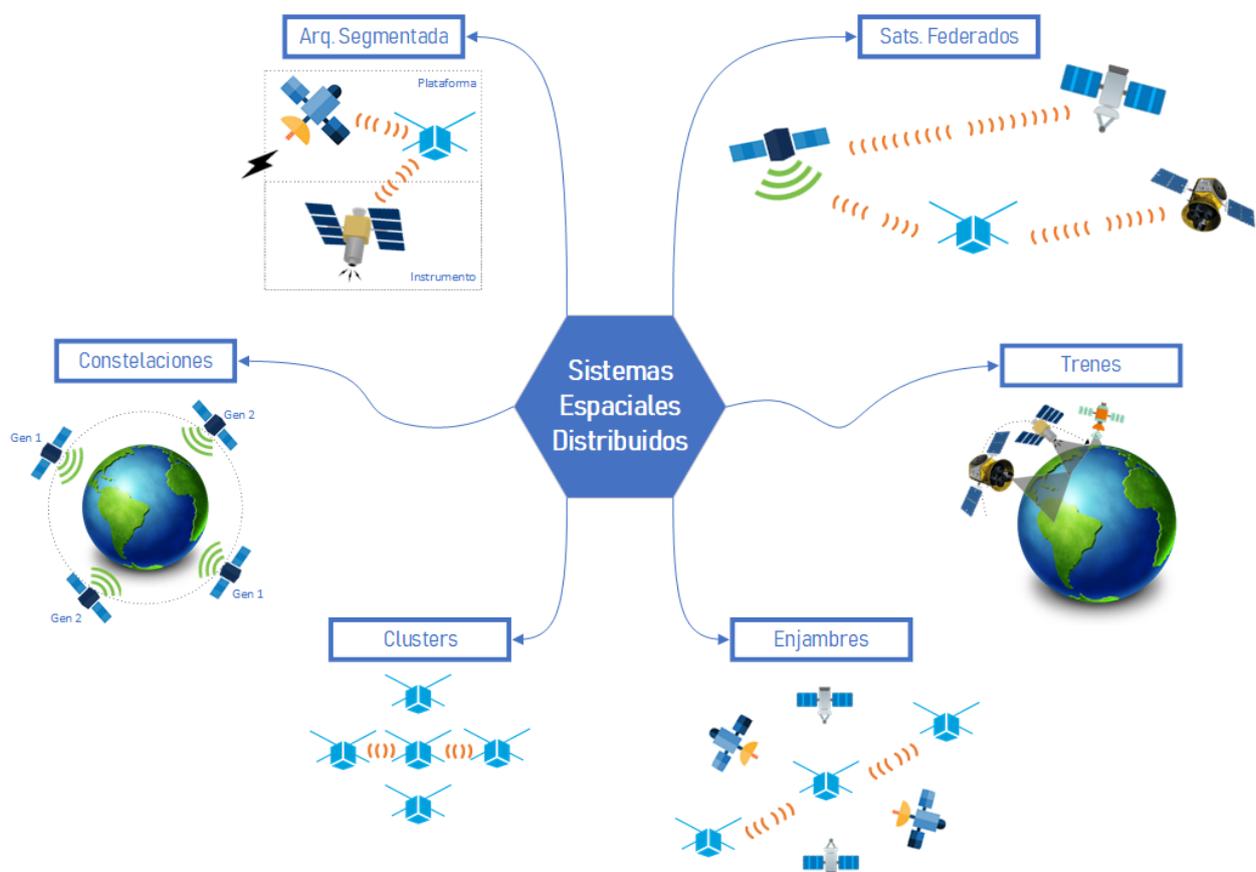


Figura 2.2 – Clasificación de los Sistemas Espaciales Distribuidos

Arquitectura	Objetivo	Cooperación	Homogeneidad	Distancia	Independencia
Constelaciones	Compartido	Requerida	Homogéneos (posibles dif.)	Regional	Independiente
Trenes	Independiente (o compartido)	Opcional o Requerida	Heterogéneos	Local	Independiente
Clusters	Compartido	Requerida	Homogéneos	Local	Independiente o co-dependiente
Enjambres	Compartido	Requerida	Homogéneos o Heterogéneos	Local o Regional	Independiente o co-dependiente
Arquitectura Segmentada	Compartido	Opcional o Requerida	Heterogéneos	Local	Independiente o co-dependiente
Satélites Federados	Independientes (misiones dif.)	Opcional (ad-hoc)	Heterogéneos	Local o Regional	Independiente

Tabla 2.1 – Análisis comparativo de las distintas arquitecturas SED

En el contexto de los SEDs, también se suele hacer mención al concepto de *vuelo en formación* (*formation flying*, en inglés). Sin embargo, no se lo ha incluido en la tabla anterior ya que, aunque también es un caso particular de SEDs, se refiere, en particular, a los requerimientos de navegación y control del conjunto de elementos y puede contener a más de una de las arquitecturas mencionadas. El vuelo en formación es el seguimiento o mantenimiento de una deseada distancia relativa de sus elementos, orientación o posición (Alfriend *et al.*, 2010) y, de acuerdo con D'Amico (2016), requiere de una precisión de navegación que va desde moderada a alta y una separación moderada de los satélites (figura 2.3).

Respecto a la dimensión *distancia* de la tabla, puede tomarse como valor de referencia lo que se define en (Selva *et al.*, 2017): Local si la separación es menor a 100 km o Regional si es mayor.

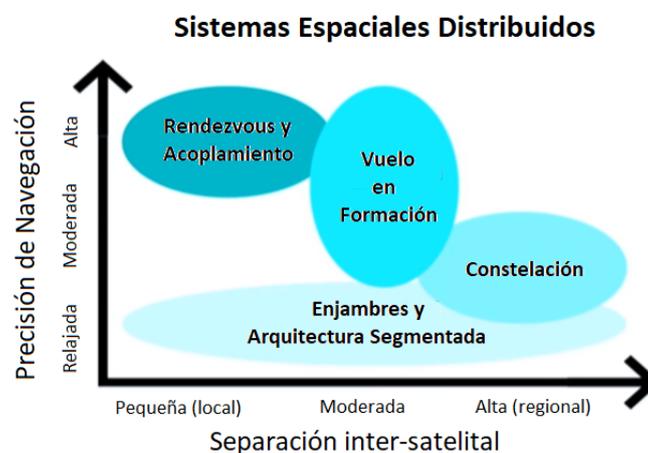


Figura 2.3 – Clasificación de los SEDs según la precisión de navegación (adaptado de [D'Amico, 2016])

2.2.1 Constelaciones

Una constelación está compuesta por múltiples satélites distribuidos en el espacio de un modo específico con el propósito de maximizar la cobertura terrestre. La cobertura terrestre es la porción de la tierra que un instrumento o antena puede ver en un determinado instante o durante un período de tiempo (Wertz *et al.*, 2011). En este tipo de SED, todos los elementos comparten el mismo objetivo de misión y aunque su coordinación es necesaria para cumplirlo, operan de forma independiente. Los satélites de una constelación son homogéneos, aunque son posibles algunas diferencias debido, en particular, a las distintas generaciones de un mismo satélite, como es el caso de la constelación *GPS - Global Positioning System* (Hofmann-Wellenhof *et al.*, 2012). La principal aplicación de las constelaciones es la provisión de servicios de telecomunicaciones y de posicionamiento satelital con cobertura mundial.

2.2.2 Trenes

Los trenes están formados por un conjunto de satélites que se desplazan a través de la misma órbita, con una separación temporal y espacial relativamente corta. Son arquitecturas híbridas integradas por varias misiones con componentes heterogéneos, que al combinarse generan mediciones científicas sinérgicas que satisfacen los objetivos globales (Poghosyan *et al.*, 2016a), aun cuando la cooperación entre ellos puede no existir. En este tipo de arquitectura, cada satélite puede tener uno o más instrumentos en común, pero individualmente fueron diseñados para alcanzar distintos objetivos (D'Errico, 2013). Desde la perspectiva global, el valor de este tipo de formaciones es la capacidad de adquirir sobre un determinado sitio de interés con diferentes instrumentos y/o desde diferentes ángulos, de modo tal de mejorar la resolución espectral y cobertura. Los trenes son especialmente útiles para aplicaciones meteorológicas o del medioambiente, ya que permite, por ejemplo: estudiar en detalle la evolución de los huracanes o incendios forestales. Un ejemplo de este tipo de configuración es *Afternoon Constellation* o *A-Train* de la NASA (Schoeberl, 2002) para observación de la tierra.

2.2.3 Clusters

Los clusters son grupos de satélites homogéneos que vuelan a una distancia corta entre ellos y que coordinan sus actividades para alcanzar un objetivo en común. En lo que refiere a su operación, los clusters pueden ser totalmente independientes unos de otros o, bien, requerir de sus pares para funcionar correctamente. Además, en este tipo de formación, los requerimientos de precisión para la navegación y control relativos son altos, y aunque los satélites no siguen la misma trayectoria, sus órbitas son elegidas para mantener constante (o cambiarlas, si es necesario) las distancias relativas entre ellos (D'Errico, 2013). Los clusters tienen muchas aplicaciones potenciales, como, por ejemplo, el uso de interferometría y técnicas de apertura sintética para simular un único satélite y de gran tamaño (Mohammed, 2001). Un ejemplo de esta configuración es la misión *Cluster II* de la ESA en colaboración con NASA (Escoubet *et al.*, 2001).

2.2.4 Enjambres

El concepto de *enjambre* surge en la biología y es utilizado en ingeniería (Bonabeau *et al.*, 1999) para referirse a aquellos sistemas que están inspirados por ciertos comportamientos de animales sociales, como ser: abejas, hormigas, aves y peces. En la ingeniería aeroespacial esta arquitectura ha ganado popularidad luego de la tendencia hacia la miniaturización de satélites y a la producción en serie de satélites pequeños de bajo costo que utilizan componentes COTS (Poghosyan *et al.*, 2016a). La formación enjambre pretende alcanzar grandes funcionalidades u objetivos, partiendo de un gran número de pequeños satélites con limitadas funcionalidades. Esta arquitectura tiene algunas características en común con los clusters, como el hecho de que comparten un mismo objetivo, requieren de una cooperación entre sus componentes, y su operación puede ser independiente o co-dependiente. Sin embargo, su mayor diferencia radica en la gran cantidad de los elementos que la componen (cientos o miles) y su mayor distribución espacial, lo que le permite alcanzar objetivos de escala global. Aunque forma parte de una tecnología que aún está en etapas de desarrollo, ya pueden vislumbrarse muchas aplicaciones, por ejemplo, estimar la composición de los asteroides (Manchester y Peck, 2011).

2.2.5 Arquitectura Segmentada

La arquitectura segmentada (AS) es aquella que se basa en la descomposición de un satélite monolítico en varios satélites más pequeños (llamados módulos, segmentos o nodos), integrados inalámbricamente en órbita, y conformando un satélite virtual con la misma funcionalidad que el del sistema original (Brown y Eremenko, 2006). Esta desagregación se basa en la división clásica por subsistemas de la ingeniería aeroespacial. Un ejemplo de ello, podría ser el de un satélite virtual (o segmentado) que esté integrado por 3 pequeños satélites: uno para la carga útil, otro que aloje el subsistema de CDH (*Command & Data Handling* o Comando y Manejo de Datos) y el tercero para el subsistema de comunicaciones con el MOC. El objetivo de esta clase de SED es el de contar con una industria satelital más responsiva, es decir, que permita responder rápidamente a cualquier cambio de requerimientos, y, lo más importante, hacerlo durante la fase operativa de la misión. La AS también presenta otras tres características importantes: robustez, mantenibilidad y durabilidad. En una arquitectura de este tipo uno podría pensar en reemplazar un instrumento dañado en órbita o mejorar la capacidad de procesamiento a bordo sin necesidad de construir un satélite completo nuevamente; bajo este paradigma sólo bastará con poner en órbita los módulos que quisieran reemplazarse o bien sumarse para ampliar y/o actualizar las funcionalidades del sistema. Aunque al día de hoy muchas de las tecnologías requeridas por la segmentación espacial aún no han alcanzado el nivel necesario de madurez, como la reconfiguración autónoma del clúster en órbita, la comunicación inalámbrica y la potencia inalámbrica, se han hecho avances en la temática (Guo *et al.*, 2009; Chu *et al.*, 2013; 2016).

2.2.6 Satélites Federados

Los sistemas satelitales federados (SSF) son una aproximación a los sistemas espaciales desde la economía colaborativa, y su objetivo es permitir el intercambio de los recursos disponibles entre distintos satélites (consumidores y proveedores) mediante la creación de redes P2P ad-hoc (von Maurich y Golkar, 2018). Mediante estas redes espaciales oportunísticas, distintas misiones (con distintos objetivos y con elementos heterogéneos) pueden resultar beneficiadas del intercambio, al aprovechar sus capacidades ociosas e incrementar la utilidad final de la misión. La capacidad ociosa se define como el tiempo total en que el satélite cuenta con recursos libres para ejecutar tareas adicionales a las de la misión (Matevosyan *et al.*, 2015). Por ejemplo, las misiones intensivas en datos como las que emplean SARs, cámaras hiperespectrales o de alta resolución, podrían incrementar el volumen de descarga de datos y reducir la latencia a través de la federación. Además, la implementación de una arquitectura de este tipo podría relajar muchas restricciones de diseños de los satélites, reducir el costo de desarrollo de los mismos y crear misiones virtuales basadas en software (Golkar y Lluch, 2015). Los FSS son la última instancia de SEDs que han aparecido en la literatura.

2.2.7 Internet en el Espacio

Resulta oportuno mencionar, antes de pasar a la próxima sección, un tema que está estrechamente relacionado con los sistemas espaciales distribuidos: Internet en el espacio. Debajo se resumen los conceptos más importantes que han sido propuestos hasta el momento (Ruiz de Azúa *et al.*, 2018):

- *Space Internet* (Bhasin y Hayden, 2001): propone la creación de un *backbone* para reducir los costos y estandarizar las comunicaciones de todas las misiones de NASA.
- *Internet-of-Space o IoS* (Raman *et al.*, 2016): luego de que se propusiera un proyecto para la puesta en órbita de miles de satélites para dar cobertura global de Internet (De Selding, 2015), se generó el debate sobre el paradigma IoS. Esta idea, basado en el concepto de IoT (Atzori *et al.*, 2010), tendría como objetivo la creación de un *backbone* para proveer Internet a todo el planeta.
- *Internet-of-Sats o IoSat* (Ruiz de Azúa *et al.*, 2018): promueve la creación de una plataforma de comunicación intersatelital esporádica entre satélites heterogéneos, con el propósito de proveer conectividad a satélites autónomos, como los que podrían formar parte de un SSF. A diferencia de las otras propuestas, este paradigma está compuesto por satélites que responden a sus propias misiones y deciden participar de la red. En otras palabras, la propuesta promueve la arquitectura P2P en el segmento espacial.

2.3 Inteligencia Artificial y Agentes Inteligentes

La tesis presenta una arquitectura espacial en la cual sus componentes son modelados como agentes. Por tal motivo, es importante definir algunos conceptos de antemano para dar un contexto y ayudar a la comprensión. Ese es el objetivo de los próximos incisos.

2.3.1 Introducción

La inteligencia artificial (AI, por sus siglas en inglés) es la disciplina que no sólo busca entender la inteligencia sino también crear entidades inteligentes. Estas entidades inteligentes son llamadas *agentes*, y se los define como “cualquier unidad capaz de percibir su entorno a través de sensores y actuar en ese medio utilizando actuadores” (Russell y Norvig, 2010, p. 34). Esta idea, que es la más general posible se representa en la figura 2.4.

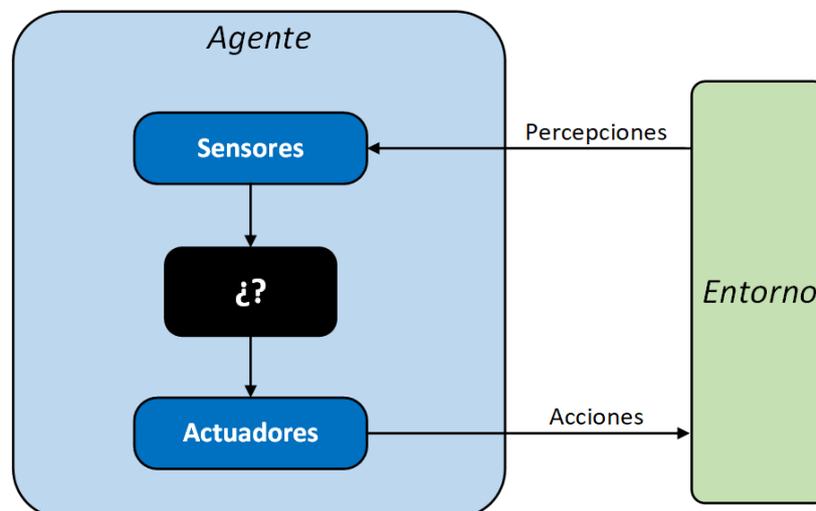


Figura 2.4 – Esqueleto de un Agente que interactúa con su entorno (adoptado de [Russell y Norvig, 2010])

La caja negra del medio (“¿?”) representa la lógica interna o proceso de decisión del agente responsable de generar una determinada acción a partir de las percepciones recibidas del entorno, y puede ser desde los más rudimentario a lo altamente complejo. Esta lógica será implementada por un programa informático y deberá hacerlo sobre un medio físico (un procesador, un robot, etc.) que permita llevar a cabo las acciones que el programa indique. De acuerdo con Russell y Norvig (2010), un agente tiene la siguiente estructura:

$$\text{agente} = \text{arquitectura} + \text{programa}$$

Por supuesto, la relación entre ambos elementos es muy estrecha, lo que significa que la arquitectura que se utilice o desarrolle deberá ir acorde a las funcionalidades incluidas en el programa. Es decir, si se trata de un agente de software que incluye dentro de su repertorio, por ej., la acción de solicitar información vía internet,

la arquitectura deberá contar con una placa de red, o al menos poder hacer uso de una. Del mismo modo, si se está en presencia de un agente físico (robot) que recomienda la acción de avanzar al finalizar la ejecución del programa, la arquitectura deberá tener efectores (ruedas) que permitan el desplazamiento del robot. De modo análogo y volviendo al tema de la tesis, si el agente es un satélite que cuenta con instrumentos, transponder y computadora de a bordo, deberá el programa incluir en su repertorio las acciones de adquirir, transmitir/recibir datos y procesar telemetría respectivamente. Este es un ejemplo genérico porque no se están considerando muchos otros subsistemas o funcionalidades de un satélite tipo, pero su objetivo es dar inicio al mapeo entre las dos áreas temáticas de la tesis: sistemas espaciales y agentes.

Volviendo a la caja negra de la figura 2.2, su representación mediante “¿?” es sin duda la más apropiada en miras de mantener una definición de agente realmente general; cualquier detalle que atinemos a incorporar allí dentro estaría generando un sesgo en la definición. Y justamente esas distintas lógicas, funcionalidades o modelos que pudiesen ser incorporados allí dentro, son las que dan lugar a las distintas clasificaciones de agentes (inciso 2.3.3) que ofrece la literatura y, cuya elección, dependerá de su ámbito de aplicación.

La figura 2.5 resume todos aquellos temas que pretenden ser introducidos en este capítulo relacionado con agentes, en el marco de la presente tesis: [i] las características de los agentes (inciso 2.3.2), [ii] una clasificación de los distintos tipos de agentes (inciso 2.3.3) y [iii] los distintos tipos de entorno con los que un agente puede interactuar (inciso 2.3.4).

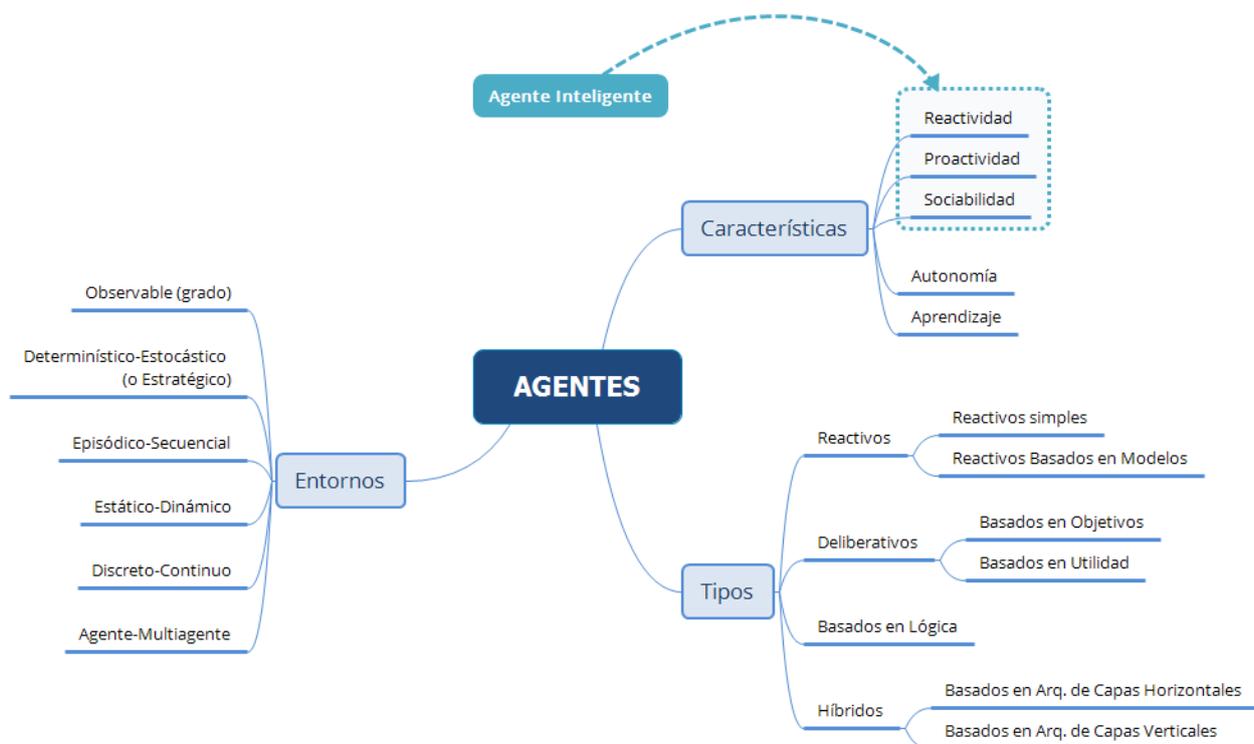


Figura 2.5 – Resumen gráfico de Agentes: Características, Tipos y Entornos

2.3.2 Características de los Agentes

En términos generales, las características que pueden presentar las agentes son las que se han enumerado en la figura 2.5, a saber: Reactividad, Proactividad, Sociabilidad, Autonomía y Aprendizaje (Wooldridge y Jennings, 1995; Wooldridge, 2011; Russell y Norvig, 2010). Por supuesto, no todos los agentes presentan todas ellas y eso dará lugar, en parte, a la distinta clasificación entre ellos. A continuación, se definen cada uno de estos conceptos para evitar ambigüedades al volver sobre alguno de ellos a lo largo de la tesis:

- **Reactividad:** es la capacidad que tiene un agente para responder a los cambios que percibe en su entorno, y hacerlo en tiempo y forma de modo tal de cumplir sus objetivos de diseño (Wooldridge y Jennings, 1995). El tiempo de respuesta es muy importante para la reactividad ya que, en ambientes altamente cambiantes, una respuesta tardía puede no provocar el resultado esperado o bien, haber quedado obsoleta. Dada la importancia de la capacidad de responder a eventos por parte de los agentes, la reactividad no sólo está presente en los agentes reactivos (una de las categorías) sino en la mayoría de ellos.
- **Proactividad:** es la capacidad que tiene un agente para exhibir un comportamiento orientado a objetivos, tomando la iniciativa en pos de la satisfacción de sus objetivos de diseño (Wooldridge y Jennings, 1995).
- **Sociabilidad:** es la capacidad que tiene un agente para interactuar con otros agentes (podrían ser humanos también) para satisfacer sus objetivos de diseño (Wooldridge y Jennings, 1995).
- **Autonomía:** es la capacidad que tiene un agente para actuar sin intervención humana o de otros sistemas. Sin embargo, es más preciso considerar la autonomía como un espectro, en el cual en un extremo se ubican los seres humanos y en el otro, por ej., un servicio de software que sólo ejecuta lo que se le ha solicitado (Wooldridge, 2011). Pero, al momento de diseñar un agente con un alto grado de autonomía, ¿cuál sería el propósito?; es decir, ¿qué se espera de un agente autónomo? ¿que haga lo que quiera cuando quiera? Aquí la clave es la siguiente: como diseñadores de agentes autónomos se espera que se le puedan delegar al agente sus objetivos de diseño, en una representación de muy alto nivel, y luego esperar que sea el propio agente quien encuentre la mejor forma de alcanzarlos mediante la creación de sub-objetivos. Se puede ver entonces que hay una relación restrictiva entre objetivos y autonomía, y esta es que los objetivos limitan la autonomía. También podrían pensarse otras restricciones para el proceder del agente, en vez de dejarlo que simplemente busque la mejor forma de alcanzar su objetivo. En línea con esto, Tambe, Scerri, y Pynadath (2002) incorporan el concepto de autonomía ajustable, cuya idea fundamental es que la toma de decisiones sea transferida del agente a una persona o a otro agente cuando se alcanzan ciertas condiciones, por ejemplo: cuando el agente percibe un alto grado de incertidumbre, o bien, cuando la utilidad esperada de que la decisión sea tomada por un tercero es sustancialmente mayor.

- **Aprendizaje:** es la capacidad que tiene un agente (con una base de conocimiento previa, o no) para aprender de sus propias observaciones/experiencias de forma tal de mejorar su rendimiento en futuras tareas. Russell y Norvig (2010) relatan que Alan Turing, en uno de sus primeros trabajos, consideró la idea de programar agentes inteligentes a mano y, luego de estimar el tiempo que le tomaría, concluyó que un mejor método sería construir maquinas con capacidad de aprender (Turing, 1950). Hoy día, y desde hace varios años, el aprendizaje automático (o más comúnmente llamado Machine Learning) se considera una disciplina aparte, dado que ha avanzado a pasos agigantados y se ha extendido a un sinnúmero de aplicaciones e industrias. Por tal motivo, no es la idea aquí ahondar en el asunto sino simplemente enumerar los distintos modelos de aprendizaje, en un alto nivel, para satisfacer al lector curioso. Ellos son: aprendizaje puramente inductivo, aprendizaje estadístico, aprendizaje basado en lógica, aprendizaje basado en conocimiento y aprendizaje por refuerzo. Otra clasificación de alto nivel también mencionada en la literatura es: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semi-supervisado y aprendizaje por refuerzo. A su vez, dentro de cada uno de estos modelos se utilizan distintos métodos y algoritmos para dar respuesta a los problemas específicos que buscan resolver.

Además de estas características mencionadas, Wooldridge (2011) define como agente **inteligente** a aquel que exhibe (i) reactividad, para percibir los cambios en su entorno y actuar rápidamente, aunque eso signifique interrumpir la acción que está ejecutando; (ii) proactividad, para no perder de vista que toda acción debe ser coherente con su objetivo de largo plazo; y (iii) habilidad social, para interactuar con otros agentes y/o personas cuando sea necesario, cooperando o negociando, para alcanzar el objetivo.

2.3.3 Tipos de Agentes

En la figura 2.5 se presentaron los tipos de agentes más comunes. Aquí se describirá brevemente cada uno de ellos, y en algunos casos, se mencionarán algunas arquitecturas ejemplo.

- **Agentes Reactivos simples:** son los agentes más simples, en los que las acciones ejecutadas sólo dependen de las percepciones actuales sin almacenar ningún estado previo. Las acciones que ejecutará el agente en cada instante dependerán de las reglas de *situación-acción* (o *si-entonces*) que han sido definidas. La ventaja de estos agentes es que son simples de implementar, pero tienen inteligencia limitada. Además, dependen mucho de la observabilidad del entorno ya que el entorno debe ser completamente observable para que puedan tomar correctas decisiones. Un ejemplo de este tipo de agentes es el de una aspiradora automática cuyo objetivo sea limpiar el piso de una habitación (Russell y Norvig, 2010). Las reglas pueden ser tan simples como: si está sucio limpiar, si está limpio desplazarse a otra posición. Uno de los ejemplos más conocidos de este tipo de agentes es la arquitectura subsumida (Brooks 1986; 1991; Ferber, 1996) que incorpora una reactividad simple

(mapea percepción directamente con acción) en una arquitectura de capas ordenadas según prioridad.

- **Agentes Reactivos basados en modelos o con estado:** la diferencia con el caso anterior es que este tipo de agentes almacenan un estado que dependen de su percepción histórica. Por un lado, esto resuelve de un modo efectivo el problema de medioambientes parcialmente observables. Por otro lado, aunque en este caso los agentes también actúan en base a reglas de *situación-acción* predefinidas, ahora la *situación* no es sólo resultado de la última percepción sino del estado del entorno, modelado a partir de la serie de percepciones y de los efectos de las posibles acciones sobre el entorno.
- **Agentes Deliberativos:** los agentes deliberativos son aquellos que incorporan un proceso de planificación en su comportamiento, es decir, son capaces de generar planes de acción (Mas, 2005). Es decir, a veces no alcanza con conocer el estado actual del entorno para saber qué acción tomar, sino cuál es la situación final a la que aspira el agente. Los agentes basados en objetivos planifican, definen los objetivos y actúan para alcanzarlos. Los agentes basados en utilidad, además de planificar y definir sus objetivos, le asignan un valor de utilidad a cada posible situación futura y buscan maximizar su función de utilidad.
En cualquiera de ambos casos, el modelo del entorno se combinará con los objetivos y/o utilidad esperada, y a partir de ahí, se decidirá acerca del plan a ejecutar. La arquitectura basada en objetivos más conocida es la BDI (Bratman *et al.*, 1988; Haddadi y Sundermeyer, 1996).
- **Agentes basados en Lógica:** son aquellos agentes concebidos bajo la aproximación tradicional de la inteligencia artificial, conocida como AI simbólica, que sugiere que “es posible darle inteligencia a un sistema si a este se le da una representación simbólica del entorno y de su comportamiento deseado, y luego se manipula sintácticamente esa representación” (Wooldridge, 2011). Las representaciones simbólicas se refieren a las proposiciones lógicas y la manipulación sintáctica a la deducción o demostración de teoremas.
- **Agentes Híbridos:** son aquellos agentes que presentan tanto reactividad como proactividad; son los más completos. Para diseñar agentes de este tipo, lo más lógico es contar con un sistema de capas (al menos 2) en donde cada una de ellas lidie con los distintos comportamientos. El flujo de control de estas arquitecturas se puede dividir en dos clases: capas horizontales o verticales (Wooldridge, 2011). En las arquitecturas de capas horizontales, cada una de ellas está conectada al sistema sensor y el sistema actuador, y actúa como un agente en sí mismo, generando sugerencias sobre la acción a ejecutar. En el caso de capas verticales (en general de 3 capas), sólo una capa se ocupa de manejar los sensores y actuadores.

2.3.4 Entornos: Tipos y Características

Los distintos entornos con los que un agente interactuará son (Russell y Norvig, 2010) los enumerados en la figura 2.5 y detallados a continuación.

- **Observable a Parcialmente Observable:** si los sensores del agente le proporcionan un conocimiento completo del estado del entorno, entonces se dice que el entorno es totalmente observable. El término completo se refiere a todos los aspectos relevantes para la toma de decisiones. En contrapartida, un entorno se define como parcialmente observable cuando existe ruido en la información, cuando los sensores son de baja calidad, o bien, cuando los datos de los sensores no incluyen toda la información relevante.
- **Determinístico o Estocástico (o Estratégico):** se dice que el entorno es determinista si el siguiente estado del medio está totalmente determinado por el estado actual y la acción a ser ejecutada por el agente. En otro caso se lo considera estocástico. En el caso en el que el medio sea determinista, excepto por las acciones de otros agentes, se lo llama estratégico.
- **Episódico o Secuencial:** un entorno episódico es aquel en el que cada episodio (ciclo de percepción y acción) es independiente, es decir, no depende del episodio anterior ni del posterior. En cambio, en un entorno secuencial las decisiones/acciones actuales afectan las decisiones futuras.
- **Estático o Dinámico:** si el medio puede cambiar mientras el agente está en proceso de deliberación, entonces el entorno es dinámico; de lo contrario, es estático. Es más fácil para los agentes tratar con entornos estáticos, ya que no tienen que estar pendientes de si el estado del medio cambia mientras ellos deliberan, ni tampoco del paso del tiempo. También se puede hablar de entornos semidinámicos cuando el entorno no cambia, pero la performance del agente se ve afectada por el transcurso del tiempo.
- **Discreto o Continuo:** la distinción entre discreto o continuo aplica a los estados del entorno, a la forma en que se maneja el tiempo, y a las percepciones y acciones de los agentes. Por ejemplo, si el entorno tiene un número finito de estados, será discreto, de lo contrario, continuo.
- **Agente o Multiagente:** cualquier detalle parece ser excesivo ya que la diferencia entre ambos tipos de entornos es básicamente si existe un sólo agente interactuando con el entorno o más de uno. Aunque la diferencia resulta evidente, no es tan claro el proceso de decidir cuáles son las entidades del entorno que debieran ser consideradas como agentes y cuáles no.

2.3.5 Sistemas Multiagente

Los sistemas multiagente (SMA) son aquellos compuestos por múltiples agentes que interactúan entre sí, típicamente intercambiando mensajes. En el caso más general, los agentes pueden estar representando a distintos actores con distintos objetivos cada uno, por lo que es importante que los agentes sean capaces de coordinar sus acciones y de cooperar y/o negociar entre ellos (Wooldridge, 2011). La premisa de los SMA es que los agentes pueden ser más efectivos cuando comparten su entorno con otros, ya que esto les permite concentrarse en las tareas en las que son más competentes y delegar otras (Dignum y Padget, 2013).

Una de las razones del creciente interés en los SMA es que son vistos como tecnologías habilitantes de ciertas aplicaciones que dependen del procesamiento distribuido, como el comercio electrónico, telecomunicaciones, redes P2P, computación en la nube, etc. (Weiss, 2013).

2.3.6 El modelo de referencia FIPA

El sistema multiagente que ha sido diseñado en la presente tesis se ha basado en el marco normativo FIPA, es decir, se ha utilizado el modelo de referencia para la gestión de agentes definido en la especificación (FIPA, 2004) - figura 2.6.

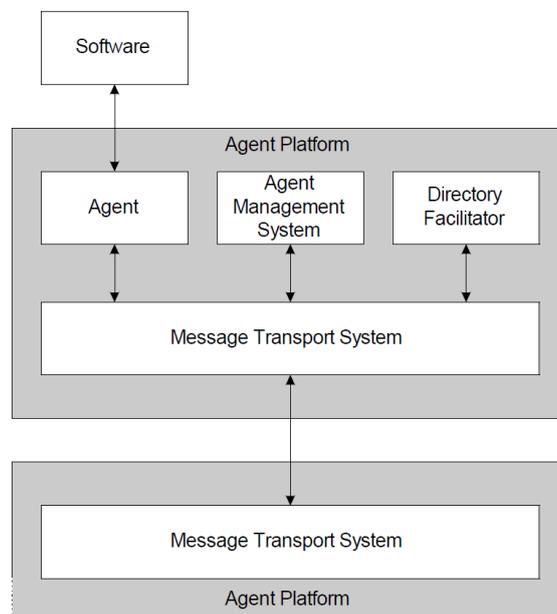


Figura 2.6 – Modelo de Referencia FIPA para la administración de Agentes (FIPA, 2004)

El modelo FIPA contempla 5 componentes lógicos, cada uno de ellos representando un conjunto de capacidades (es decir, servicios) y sin implicar ninguna configuración física específica:

1. *Agent Platform*: provee la infraestructura física en la que los agentes se alojan. Incluye el hardware, el sistema operativo, los cuatro elementos siguientes y cualquier otro software de soporte agentes.

2. *Agent Management System*: responsable de la operación y tareas de control de la plataforma, y de mantener el directorio de los agentes de la plataforma (páginas blancas).
3. *Directory Facilitator*: es opcional, pero si se implementa debe hacerse como un servicio. Brinda el detalle de los servicios ofrecidos por cada agente (páginas amarillas).
4. *Message Transport System*: método de comunicación estándar entre los agentes (FIPA, 2002b).
5. *Agent*

Además de estos cinco elementos, la entidad *Software*, que se ubica fuera de la plataforma agente, engloba todas aquellas instrucciones ejecutables que no son agentes, pero que pueden ser accedidas a través de un agente. Los agentes pueden utilizar ese software para, por ejemplo, agregar nuevos servicios, adquirir nuevos protocolos de comunicación, etc.

FIPA establece el marco general para la correcta operación y comunicación entre los agentes, pero no pone condiciones respecto a la reificación de la arquitectura. Por tal motivo, los detalles de la implementación de las plataformas agentes (AP, por sus siglas en inglés) y de los agentes, quedarán supeditadas a las decisiones de diseño de los desarrolladores de sistemas.

Comunicación entre Agentes: FIPA-ACL

Todo agente se comunica utilizando un lenguaje de comunicación para agentes (ACL, por sus siglas en inglés) y FIPA implementa esta comunicación mediante el envío de mensajes. El lenguaje de comunicación FIPA-ACL define 22 actos de comunicación o performativos (FIPA, 2002c), que pueden pensarse como distintas clases de mensaje que definen su significado independientemente del contexto. Los dos más importantes y, a partir de los cuales se generan todos los demás, son *inform* y *request*.

Generalmente la comunicación entre agentes involucra una secuencia de mensajes (no sólo uno), como así también, muchas veces esa interacción termina resultando en patrones similares. Por tal motivo, FIPA también ha definido algunos patrones, llamados Protocolos de Interacción (IP, por sus siglas en inglés), siendo tan sólo un ejemplo el *FIPA-ContractNet-IP* (FIPA, 2002d). Aunque no es obligatorio su uso, en caso de emplear alguno de los definidos, los agentes deberán comportarse según lo especificado.

En lo que respecta a la semántica de los mensajes, FIPA utilizó el lenguaje formal SL para mapear cada mensaje FIPA-ACL con una fórmula (FIPA, 2002g) y así representar las acciones, creencias y deseos de los agentes. En relación a la sintaxis, en una infraestructura típica que implementa el modelo FIPA-ACL, los mensajes se codifican en formato texto o XML (figura 2.7).

Finalmente, para mayores detalles sobre la estructura, codificación y empaquetamiento de los mensajes FIPA referirse a (FIPA, 2002a; 2002b).

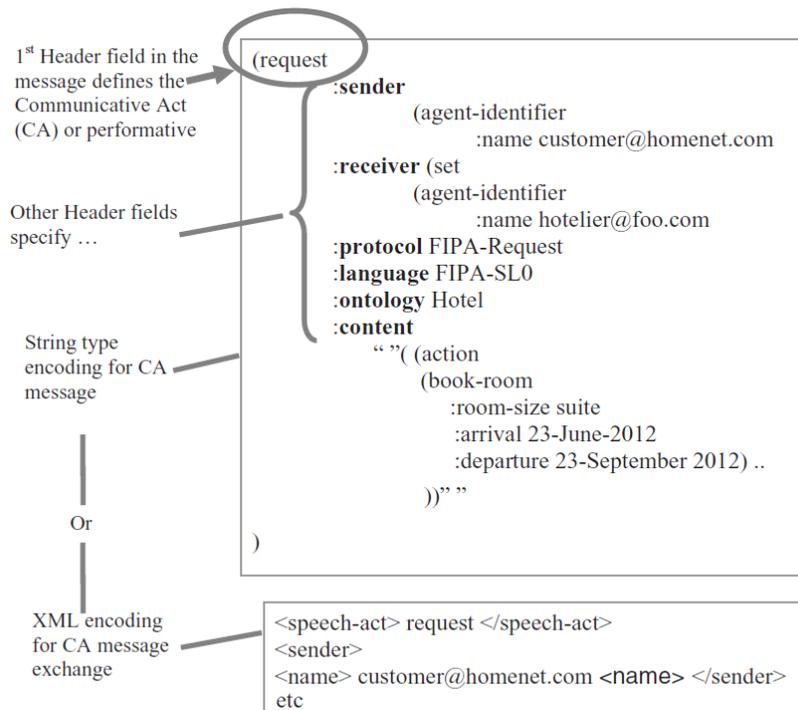


Figura 2.7 – Sintaxis para el intercambio de mensajes ACL-FIPA (Poslad, 2007)

2.4 Economías Colaborativas

Aunque está claro que las economías colaborativas (o consumo colaborativo) son un modo de intercambiar bienes y servicios distinto del tradicional, no hay un consenso en su definición. Algunos autores la definen desde la visión macroeconómica, otros desde la perspectiva microeconómica, algunos abordan el tema desde el proceso, otros desde las tecnologías y también desde lo legal. Para una revisión detallada de las múltiples visiones, referirse a los trabajos de Sundararajan (2016) y Puschmann y Alt (2016).

Aquí se usará la definición de Frenken (2017) ya que ella enfatiza en el uso de las capacidades ociosas, concepto fundamental de los satélites federados y de la arquitectura propuesta. Dicha definición se apoya más en la microeconomía y dice así:

“las economías colaborativas son aquellas en las que los consumidores se conceden acceso temporal a activos físicos subutilizados (capacidad ociosa), posiblemente por dinero”

En el ámbito espacial, Matevosyan, *et al.* (2015) analizaron el caso de intercambio de recursos ociosos en satélites de observación de la tierra y concluyeron que dicha dinámica puede generar un cambio positivo y de alto impacto en la industria. Asimismo, de acuerdo con Pica y Golkar (2017), las economías colaborativas ofrecen la posibilidad de maximizar el ciclo de vida de los sistemas a través de un diseño personalizado de su modelo de operaciones.

Capítulo 3

ESTADO DEL ARTE

En este capítulo se incluyen las reseñas de dos clases particulares de sistemas espaciales distribuidos: la arquitectura satelital segmentada (inciso 3.1.1) y los sistemas satelitales federados (inciso 3.1.2). Luego se detallan las distintas aplicaciones del paradigma multiagente en la industria espacial (sección 3.2).

3.1 Sistemas Espaciales Distribuidos

La arquitectura de la tesis se enmarca dentro de los sistemas espaciales distribuidos y, en particular, está ligada a dos clases de SEDs: la arquitectura satelital segmentada y los sistemas satelitales federados. Para ambos tipos, se llevó a cabo un exhaustivo relevamiento histórico de la literatura con el objetivo de profundizar el análisis preliminar y detectar nuevas oportunidades de investigación. Los resultados se presentan en los incisos 3.1.1 y 3.1.2, correspondientemente.

Para una descripción detallada de los distintos tipos de SEDs, sus tecnologías relacionadas y su revisión histórica, referirse a los trabajos de Selva *et al.* (2017), Poghosyan *et al.* (2016a; 2016b), Corbin (2015) y D'Errico (2013), entre otros.

3.1.1 Arquitectura Satelital Segmentada

Respecto al paradigma de arquitectura satelital segmentada (ASS), el término se acuñó en el año 2006 por investigadores del DARPA (Brown y Eremenko, 2006a), aunque algunas de las ideas subyacentes ya se habían tratado antes (Molette *et al.*, 1984; Kusza y Paluszek, 2000; Mathieu y Weigel, 2005). Brown y Eremenko definen a la ASS como la arquitectura que resulta del fraccionamiento de un único satélite monolítico en distintos segmentos, nodos o módulos, y en donde cada uno de ellos cumple distintas funciones

(cómputo, comunicación, almacenamiento de datos, generación de datos de ciencia y generación de energía). La originalidad de la idea radica en la división de funciones en distintos módulos que son integrados de forma inalámbrica directamente en órbita. La propuesta de valor que brinda la segmentación se describe en (Brown y Eremenko, 2006b), donde sus bondades son presentadas como solución a la espiral creciente en costo, tamaño, complejidad y fragilidad de los satélites monolíticos tradicionales.

Más adelante, un proyecto del *Advanced Concepts Team* de la Agencia Espacial Europea (ESA) estudió el tema del intercambio de recursos en órbita entre pequeños satélites, caracterizando a los segmentos como consumidores y proveedores de recursos (Leitner, 2010). Luego, O'Neill y Weigel (2011) desarrollaron la herramienta SET (*Spacecraft Evaluation Tool*) para evaluar la propuesta de valor de las arquitecturas segmentadas en comparación con los satélites monolíticos, para el caso particular de misiones de observación de la tierra. Por su parte, Dubey *et al.* (2012) introdujeron una novedosa arquitectura de software de tres capas llamada F6MDA, específicamente diseñada para abordar las particularidades de las ASS como el hardware distribuido, interrupciones en la comunicación, entre otros. Chu *et al.*, 2014 abordaron el diseño organizacional o global de una arquitectura segmentada con el fin de desarrollar satélites de observación de la tierra que permitan prestaciones a más largo plazo y luego la implementaron (Chu *et al.*, 2016) para el estudio de la configuración en vuelo de clústeres mediante el paradigma segmentado vs. el monolítico, obteniéndose mejores resultados con el primero.

Por último, existe un antecedente en la Argentina de una misión satelital que ha sido concebido sobre la base de las arquitecturas segmentadas: el proyecto SARE de la CONAE (CONAE, 2010; 2019), formada por satélites livianos a ser puestos en órbita con lanzadores argentinos. El proyecto incluye dos series de instrumentos, ópticos y microondas, las cuales conformarán misiones satelitales de observación de la tierra. El estado actual del proyecto se desconoce.

Como corolario, aunque las primeras ideas acerca de la segmentación satelital surgieron en la década del ochenta, el concepto tomó vigor en el ámbito académico cuando el departamento de defensa de los Estados Unidos, a través de la agencia DARPA, decidió impulsar el programa F6 (*Future, Fast, Flexible, Fractionated, Free-Flying Spacecraft united by Information eXchange*), basado en este nuevo concepto (Brown *et al.*, 2009). Aunque dicho programa finalmente fue cancelado en el 2013 (Ferster, 2013), la arquitectura ASS sigue siendo de interés en la academia. Sin embargo, a pesar de las múltiples bondades del concepto de la segmentación (robustez, escalabilidad, responsabilidad, etc.), al día de hoy no se ha puesto en órbita ningún sistema satelital segmentado. En (Selva *et al.*, 2017) se mencionan algunas posibles razones del por qué esto puede estar sucediendo, y principalmente están relacionadas con el factor humano.

3.1.2 Sistemas Satelitales Federados

El paradigma de Sistemas Satelitales Federados (SSF) es uno de los últimos conceptos que ha sido introducido (Golkar y Lluch, 2015) en el marco de los SEDs y, al igual que en el caso de la ASS, hasta el momento no ha sido implementado ni ha sido parte de ninguna misión de demostración tecnológica en órbita. La idea que subyace detrás del paradigma SSF es el concepto de *Systems-Of-Systems*, SoS (Maier, 1998), que implica dos características fundamentales para cada elemento del SoS: (i) operación independiente y (ii) administración independiente. Es decir, cada elemento puede ser propiedad u operado por una organización independiente y, al mismo tiempo, todo elemento debe poder llevar a cabo sus operaciones normalmente si se lo separa del SoS. La administración independiente o descentralización de la propiedad es una de las características que distingue a los SSF de la ASS, ya que en una ASS los activos (o módulos) son propiedad de una misma organización, mientras que los activos de un SSF forman parte de distintas organizaciones. Y justamente esta resignación en el control de los activos, junto con el intercambio de recursos, es la razón por la cual las organizaciones pueden alcanzar sus objetivos de un modo más efectivo en costo (Selva *et al.*, 2017).

Un tipo de descentralización de la propiedad, que no es precisamente un SSF, es el de los *hosted payloads* o *piggyback payloads* (cargas útiles alojadas). En estos casos, los clientes vuelan (alojan) sus cargas útiles en satélites de terceros y pagan por los recursos que consumen de la plataforma: potencia, transferencia de datos, masa, etc. Aunque estos módulos están conectados con el satélite, operan de forma independiente. Un ejemplo de implementación de este concepto de misión es el programa *Iridium Next* (Kramer, 2019).

En lo que resta de este inciso se resumirán aquellos trabajos que tratan específicamente del concepto SSF. Los primeros trabajos, previos a la definición formal del concepto (Golkar y Lluch, 2015), se desarrollaron hace pocos años atrás, entre el 2013 y 2014. En el primero de ellos (Golkar, 2013a) se presentan los tempranos esfuerzos del proyecto de investigación (del instituto *Skoltech*, en colaboración con el *MIT*), en su objetivo de innovar en el diseño de sistemas espaciales mediante un nuevo paradigma. En él se incluye un ejemplo de aplicación para el caso de una misión de observación de la tierra (EO, por sus siglas en inglés) con un radioaltímetro en órbita, y se demuestran las ventajas y la factibilidad de un SSF en ese contexto. A continuación, se exploran las posibilidades y potenciales problemas asociados al concepto SSF y se desarrolla (Golkar, 2013b) un modelo completo de los *stakeholders* involucrados, para finalmente hacer un análisis de costo-beneficio de una eventual implementación del concepto. Posteriormente, se introduce (Golkar, 2013c) una metodología para evaluar, desde el punto de vista técnico y económico, el uso de cargas útiles alojadas en satélites de terceros en el marco de un SSF, y se describen los beneficios de una eventual participación de la estación espacial internacional (ISS, por sus siglas en inglés “International Space Station”) como proveedor de recursos en órbita.

El primer prototipo de un simulador de alto nivel para estudiar aspectos claves del concepto SSF, como la latencia, ancho de banda, y costo, se presenta en (Grohan *et al.*, 2014), con el objeto de servir como base

para futuras decisiones de arquitectura. El simulador incluye elementos de vuelo y tierra y sus correspondientes transpondedores, y permite computar la geometría y visibilidad satélite-satélite y satélite-tierra. Aquí también se emplea a la ISS como caso de aplicación, la cual actúa como una plataforma que provee servicios espaciales a misiones EO de órbitas sólo sincrónicas (SSO, por sus siglas en inglés). Como parte de este cuerpo de trabajos preliminares que van estudiando y refinando los aspectos técnicos del paradigma SSF, aparece uno que propone (Lluch y Golkar, 2014a) un método para la optimización de la cobertura satélite-satélite, orientado hacia las comunicaciones ISL oportunísticas entre satélites de una federación. Asimismo, se desarrolla (Lluch y Golkar, 2014b) un modelo más sofisticado de las operaciones de un SSF y se realiza la primera la cuantificación del impacto en el diseño a nivel sistema (en particular, masa y potencia) de un satélite que quisiera operar una carga útil ISL para participar en una federación de satélites y así poder intercambiar sus recursos ociosos.

Golkar y Lluch (2015) definen finalmente a los SSF como una nueva arquitectura dentro de los SEDs, la cual daría comienzo al *cloud computing* (o computación en la nube) en entornos espaciales y cambiaría rotundamente la forma en que las misiones espaciales son concebidas y operadas. El concepto promete una reducción en los costos, un aumento en el rendimiento y confiabilidad de las misiones satelitales, a través del *networking* entre distintas misiones en órbita, creando un conjunto de recursos a ser intercambiados por los distintos participantes de la federación. El trabajo incluye ciertas definiciones formales de los elementos que componen una federación y provee una taxonomía del conjunto de opciones para aquellos satélites que deseen participar del SSF. Asimismo, los autores desarrollan una aproximación que integra el análisis de misión y la evaluación económica, analizan la factibilidad del modelo de negocio y presentan resultados prometedores.

En lo que respecta al tema de comunicación en los SSF, Lluch *et al.* (2015) desarrollaron un protocolo del tipo MANET (Hoebeke *et al.*, 2004) para operar la red de satélites de una federación. Entre otras cosas, el trabajo demostró los beneficios del uso de un SSF para mejorar la latencia de datos de las misiones satelitales. Por otro lado, Lluch y Golkar (2015) profundizaron el análisis de impacto (bajando hasta el nivel subsistema) sobre el diseño de un satélite que requiere un módulo ISL para operar dentro de un SSF, y abordaron el problema de optimización de la geometría orbital para aquellos satélites especialmente dedicados a la provisión de servicios de SSF. Matevosyan *et al.* (2015) extienden la idea de “satélites intercambiando recursos en órbita” al concepto de Misiones Satelitales Virtuales (VSM, por sus siglas en inglés), definidas como misiones espaciales oportunísticas (por un determinado período de tiempo) basadas en el uso compartido de las capacidades ociosas presentes en los satélites de una federación. Los autores postulan que las VSMs pueden ser consideradas como *missions-as-a-service* (misiones como servicio) y el foco del trabajo es estudiar cómo estas misiones virtuales pueden ser diseñadas y planificadas, en comparación con las misiones tradicionales. Su modelo define a la capacidad ociosa como la cantidad de tiempo del que dispone el satélite para llevar adelante otras actividades, además de los de la propia misión.

El primer experimento en un ambiente aeroespacial para analizar la propuesta de valor del concepto FSS, se llevó a cabo en 2015 (Akhtyamov *et al.*, 2016). En él se utilizaron dos globos aerostáticos que emplearon radios definidas por software de uso comercial (COTS SDRs) para intercomunicarse en vuelo e interoperar con una estación terrena. El trabajo caracterizó la red establecida en vuelo y analizó el rendimiento de la retransmisión de datos de largo alcance (del estilo de un FSS) ejecutadas a altas altitudes. El experimento proveyó una prueba de concepto de la utilización de COTS SDRs en el ambiente aeroespacial y mostró un ejemplo en que la utilización de la perspectiva de FSS incrementaba la confiabilidad del sistema. Por su parte, Grogan *et al.* (2016) analizaron el comportamiento de los múltiples *stakeholders* en un SSF desde la perspectiva de la teoría de juegos (Osborne y Rubinstein, 1994) y abordaron la brecha entre la ingeniería en sistemas y la ingeniería de SoS, definiendo nuevos métodos de modelado para los SSF. Asimismo, el consorcio ONION, financiado por el programa Horizon 2020 de la Unión Europa publicó un trabajo identificando los servicios de observación de la tierra que se verían beneficiados por el desarrollo de dos tipos específicos de SEDs: arquitecturas segmentadas y sistemas satelitales federados, lo que muestra un claro interés en la temática (Matevosyan *et al.*, 2016).

La primera mención al concepto de *agente* en el marco de los SSF se encuentra en el trabajo de Pica y Golkar (2017), en el cual se estudiaron los factores macroeconómicos y las políticas de precios como reguladores de los SSF. En él se desarrolló un *framework* basado en agentes para simular y evaluar los mecanismos de precios en el comercio de *commodities* en órbita como, por ejemplo, el enrutamiento de datos. El análisis utilizó un ejemplo de satélites federados con propósitos comerciales operando en órbitas LEO (*Low Earth Orbit* u Órbita Terrestre Baja) y adoptó el mecanismo dinámico de fijación de precio conocido como *sealed-bid reverse auction* o subasta inversa de oferta sellada (Krishna, 2010). Los resultados contribuyeron al entendimiento de los mecanismos que se requieren para implementar exitosamente una federación de satélites con fines comerciales en órbita. Por otro lado, Von Maurich y Golkar (2018) son los primeros en proponer una caracterización de los requerimientos de seguridad para el intercambio de datos inter-satelitales en una federación. En su trabajo, propusieron un protocolo basado en una infraestructura de clave pública (PKI, por sus siglas en inglés), para las comunicaciones satélite-satélite dentro de la federación, de forma tal de asegurar la autenticación de datos y la integridad y confidencialidad en la transferencia de datos. Luego de los experimentos realizados, se derivaron una serie de recomendaciones para implementar mecanismos de seguridad en los FSS.

Más recientemente, Lluch y Golkar (2019) definen el concepto de Federación de Sistemas (FoS, por sus siglas en inglés) y proponen una metodología específica para su diseño. El trabajo incluye a los FSS como caso de aplicación, dado que estos son considerados una instancia de los FoS. El análisis focaliza en el aspecto económico y busca encontrar las mejores alternativas de arquitectura a partir de las cuales a una determinada misión le resulte ventajoso, en términos de ROI (retorno de inversión), incorporar una interfaz ISL para pasar a intercambiar recursos dentro de la federación.

3.2 Sistemas Multiagente en la industria Espacial

El primer intento de utilizar agentes en el espacio fue a bordo de la sonda espacial *Deep Space 1* (DS1) de la NASA en 1999, una misión principalmente de demostración tecnológica (Rayman, 2000). Un grupo de investigadores de la misión desarrolló una nueva arquitectura para el control y operación autónoma de la nave, llamada *Remote Agent* (Mussettola *et al.*, 1998; Bernard *et al.*, 2000), que fusionaba modelos reactivos y basados en lógica para la planificación y ejecución de planes.

Siguiendo con la temática de autonomía en vuelo, Das *et al.* (1999) elaboraron un modelo SMA para los procesos de planificación y programación de tareas, con el objeto de lograr autonomía en la aeronave. Schetter *et al.* (2003) desarrollaron una arquitectura multiagente y un entorno de simulación (*ObjectAgent*) para propiciar el desarrollo de sistemas satelitales autónomos. Xu *et al.* (2003) diseñaron el sistema multiagente MAPS para la planificación autónoma en vuelo, en el cual cada subsistema del satélite era concebido como un agente y entre todos, interactuaban para alcanzar los objetivos. Más adelante, se cuenta con otro antecedente de NASA (Truskowski *et al.*, 2004) y su modelo de arquitectura de misión llamada ANTS (por sus siglas en inglés Autonomous Nano Technology Swarm), que propone la idea de un conjunto de agentes inteligentes auto-organizados que reproducen el comportamiento de enjambre observado en los insectos y que podrían ser capaces de operar autónomamente por varios años en el espacio. También se diseñó una arquitectura multiagente para el control autónomo de satélites en vuelo en formación (Thanapalan y Veres, 2005), y se realizó un diseño experimental (Sangiovanni *et al.*, 2005) de un *scheduler* a ser incorporado a bordo del satélite, cuya arquitectura estaba basada en SMA y cada agente se asociaba a un subsistema del satélite, cubriendo no sólo la programación de las tareas sino también a ejecución de las mismas. Luego, Bonnet y Tessier (2007) abordan la planificación a bordo de una constelación de satélites de observación de la tierra, mediante una arquitectura SMA que utiliza una aproximación mixta: deliberativa para la planificación individual y reactiva para la colaboración entre los distintos elementos.

Más adelante, Yliniemi *et al.* (2014) propusieron un modelo de múltiples agentes robóticos que utilizan aprendizaje por refuerzo (Sutton y Barto, 1998) para resolver el problema de coordinación a distancia, en particular de *rovers* planetarios. Por su parte, Colby *et al.* (2016) diseñaron una arquitectura multiagente para el sistema de aprendizaje (supervisado) de múltiples robots que deben explorar entornos desconocidos.

En el segmento de tierra también se cuenta con muchas aplicaciones SMA, en su mayor parte relacionados con actividades de planificación y programación de misión. Un ejemplo es el sistema OCAMS (Sierhuis *et al.*, 2009) de NASA, un sistema multiagente operativo en el centro de control de misión de NASA que se encarga de espejar en tiempo real todos los datos que se transfieren desde y hacia la ISS. Mas recientemente, en (Bonnet *et al.*, 2015) se presenta el sistema ATLAS, un SMA autoadaptativo que permite planificar dinámicamente las actividades de una constelación de satélites de observación de la tierra. También se propuso (Ntagiou *et al.*, 2017) la aplicación de un algoritmo de inteligencia de enjambre para diseñar el

sistema de planificación, basado en SMA, de una misión satelital de retransmisión de datos. En (Skobelev *et al.*, 2017) se desarrolló un sistema de planificación basado en sistemas multiagente para la planificación de un enjambre de satélites de observación de la tierra. Y finalmente, el trabajo de Li *et al.* (2018) presenta una plataforma multiagente diseñada específicamente para estudiar el problema de planificación y cooperación entre múltiples satélites autónomos.

Aunque no es precisamente un trabajo basado en SMA, es pertinente mencionar el reciente trabajo de Chien *et al.* (2017), quienes presentan un concepto de misión novedoso: el HypsIRI IPM. Este trabajo, llevado a cabo por el *Jet Propulsion Laboratory* del *California Institute of Technology* automatiza e integra, mediante técnicas de AI, las actividades del segmento de vuelo con las de tierra y, a su vez, prevé que los instrumentos analicen imágenes a bordo para la detección de patrones en la superficie. El trabajo presenta algunas similitudes conceptuales con la arquitectura que se propone en la tesis.

El paradigma SMA también ha sido aplicado a la arquitectura satelital segmentada. Algunos ejemplos son los trabajos de Leitner (2010) y de Chu *et al.* (2014; 2016), que ya han sido descritos en el inciso 3.1.1. En el caso de sistemas federados, sólo se cuenta con el antecedente de Pica y Golkar (2017), presentado en el inciso 3.1.2.

Capítulo 4

DESCRIPCIÓN DEL PROBLEMA

Se presenta en este capítulo la contribución al problema en términos de desafíos que plantean las arquitecturas SSF en función de la revisión del estado del arte efectuada en el capítulo anterior.

Como contribución principal a la problemática de las arquitecturas SSF, se propone desarrollar y validar una arquitectura con mayor alcance y nivel de detalle que la actualmente considerada en el estado del arte. En este orden, se rescata como contribución al problema de los SSF:

- *Alcance del sistema:* no se ha detectado ningún trabajo que incluya en una misma arquitectura todos los elementos que han sido considerados en el SatSES: satélites, estaciones terrenas, plataforma de intercambio de servicios, centros de misión, usuarios y proveedores de servicios satelitales.
- *Detalle del Diseño:* no se ha detectado ningún trabajo que haya presentado un diseño interno de los agentes, incluyendo la comunicación entre ellos, al mismo nivel de detalle que el de esta tesis.
- *Dinámica del sistema:* no se ha detectado ningún trabajo que describa en detalle, como se ha hecho aquí en el capítulo 6, el concepto de operaciones del sistema para los casos de aplicación elegidos.
- *Elementos en común con otros trabajos:* se han podido encontrar algunos elementos en común con dos de los trabajos referidos previamente, a saber:
 - Matevosyan *et al.* (2015): aquí se presenta un concepto de misión basado en VSMs, que les permitiría a los usuarios obtener información satelital comprando y combinando tiempo de instrumentos de distintos satélites mediante una interfaz web. El diseño también incorpora una unidad que centraliza los pedidos de los usuarios y luego genera el perfil de pedidos para cada operador de satélites.

Por lo tanto, los elementos en común son: el concepto de misión VSM, la interfaz web para los usuarios y la unidad centralizadora.

- Chien *et al.* (2017): los autores describen el concepto de misión HypsIRI IPM que automatiza e integra las actividades del segmento de vuelo con las de tierra, de forma tal que los pedidos realizados por los clientes desde una interfaz bien conocida como *Google Earth*, son combinados con el plan de actividades del satélite, mejorando la latencia de la entrega de productos. Asimismo, el diseño prevé que los instrumentos analicen las imágenes a bordo para detectar patrones de incendios, erupciones de volcanes, etc., generando alertas que resulten en una replanificación de las actividades.

Por lo tanto, los elementos en común son: automatización e integración de las operaciones de tierra con las de vuelo, interfaz web para los usuarios, autonomía a bordo y caso de aplicación de emergencias.

En definitiva, aun considerando las similitudes encontradas en los dos trabajos citados, el alcance y el detalle en la arquitectura propuesta, no es comparable. Además, el foco de ambos trabajos son distintos al de la tesis.

Por lo tanto, se detecta una clara oportunidad para contribuir a la literatura científica a través de la arquitectura propuesta: el Sistema de Intercambio de Servicios Satelitales (SatSES) que se presenta en el próximo capítulo, y cuya arquitectura presenta algunas características que aún no han sido sintetizadas en la literatura.

Capítulo 5

SOLUCIÓN – ARQUITECTURA DEL SISTEMA

En este capítulo se brinda una breve introducción al sistema, incluyendo su árbol de producto (sección 5.1), se describe la arquitectura del AgentSat (sección 5.2) y el diseño interno de sus agentes (sección 5.3). Posteriormente se presenta la plataforma de intercambio de servicios satelitales o SSEP (sección 5.4) y el diseño interno de sus agentes (sección 5.5). A continuación, se presentan las características de las plataformas agente, tanto de vuelo como de tierra (sección 5.6) y se presenta la arquitectura de las estaciones terrenas que se suscribirán al SSEP (sección 5.7). Finalmente, se caracterizan todos los agentes del sistema (sección 5.8) y se aborda la comunicación entre los agentes (sección 5.9).

5.1 Introducción al Sistema

El Sistema de Intercambio de Servicios Satelitales (SatSES, por sus siglas en inglés) es la materialización de una idea que nace en el entrecruzamiento de 3 ideas fundamentales (figura 5.1):

- Satélites Federados
- Satélites Inteligentes
- Economías colaborativas

Los sistemas satelitales federados traen consigo la idea del aprovechamiento de la capacidad ociosa de los satélites; los satélites inteligentes presentan autonomía y son capaces de aprender, de forma tal que pueden tomar decisiones a bordo y decidir, entre otras cosas, acerca de la mejor manera de maximizar la utilidad de la misión dado una determinada cantidad de recursos ociosos. Por último, aunque el concepto de economías colaborativas también está embebido en el de los SSF, aquí está haciendo referencia a la nueva forma de intercambiar bienes y servicios que se está dando con éxito en otras industrias pero que aún no ha llegado a la industria espacial.

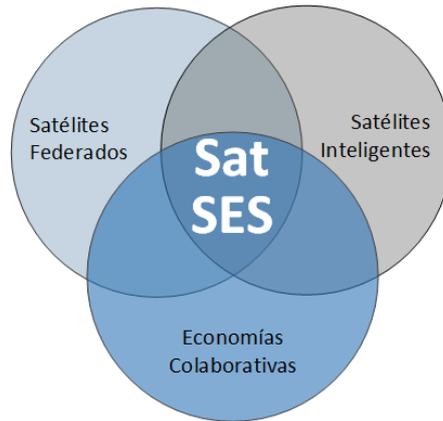


Figura 5.1 – Raíces de la idea del SSEP

5.1.1 Diseño Conceptual del Sistema

Conceptualmente el sistema SatSES puede representarse como el esquema de la figura 5.2. En lo que respecta al segmento de vuelo, el sistema está compuesto por satélites con cierto grado de autonomía llamados AgentSats. En el caso del segmento de tierra, se cuenta con estaciones terrenas (ET), los centros de control de misión (MOC) y una plataforma de intercambio de servicios satelitales (SSEP) que centraliza las operaciones de intercambio. Los clientes del sistema son los usuarios, los propios MOCs y los proveedores de determinados servicios informáticos (procesamiento de imágenes, almacenamiento en la nube, etc.) que son de interés dentro de la comunidad espacial.

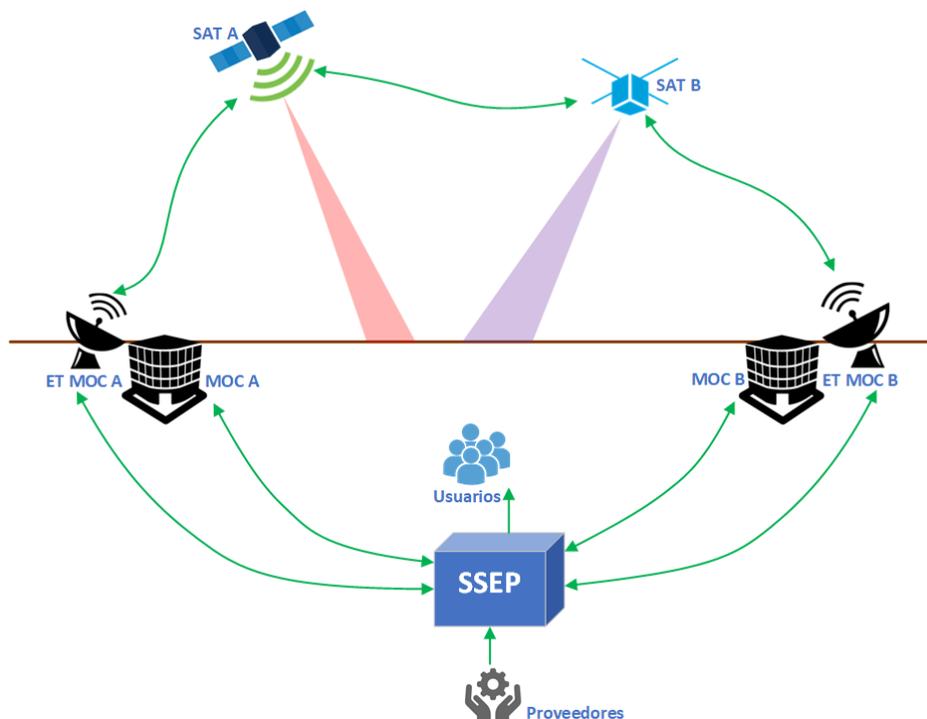


Figura 5.2 – Diseño Conceptual del Sistema

5.1.2 Product Tree

El árbol del producto del sistema se presenta en la figura 5.3. En el primer nivel se ubican 3 elementos: el AgentSat, la Estación Terrena con su extensión para poder operar con el SSEP y la plataforma SSEP. El resto de los niveles no vale la pena mencionarlos aquí ya que se describirán en detalle en las secciones subsiguientes.

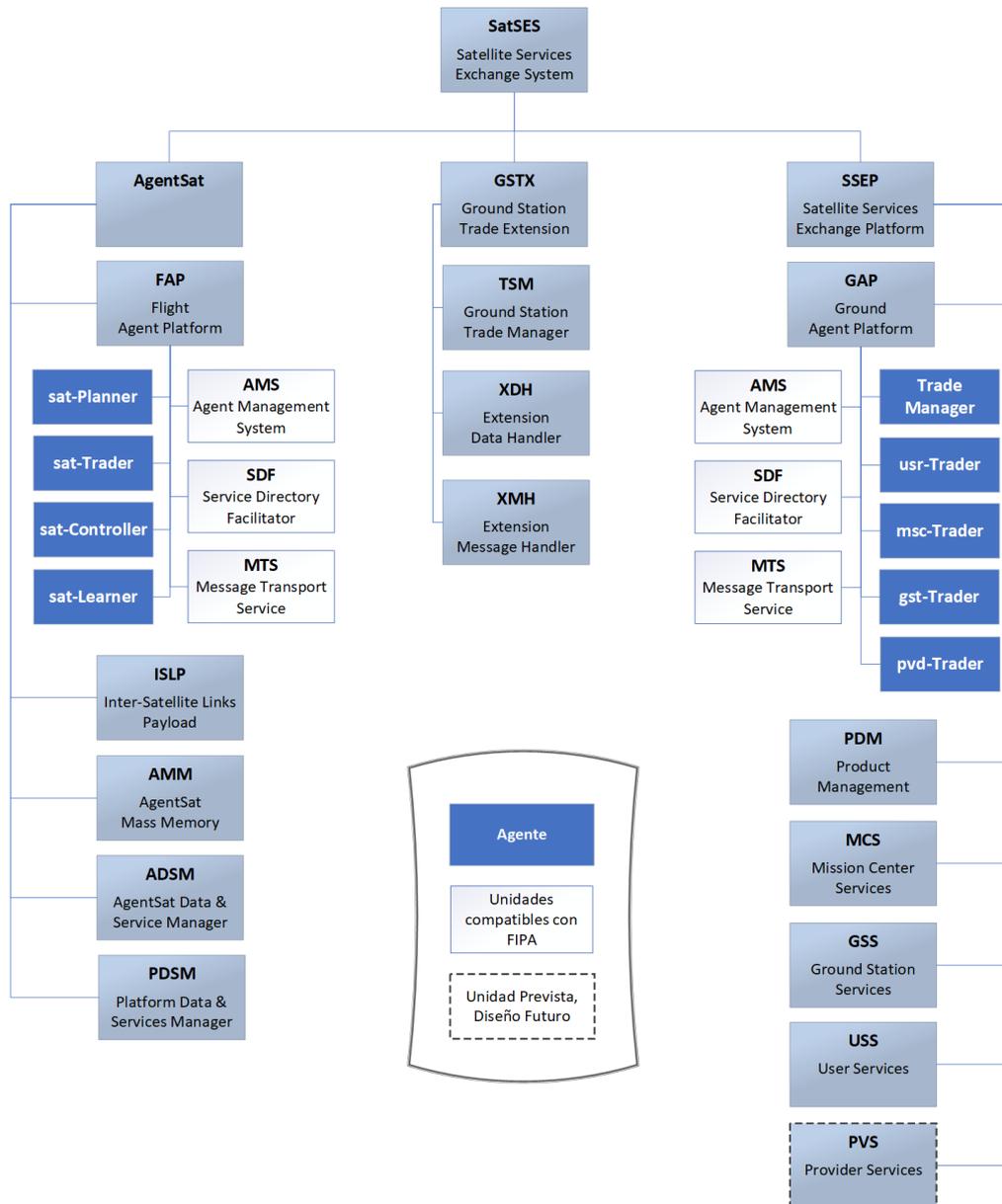


Figura 5.3 – Product Tree del Sistema

La arquitectura del sistema que se describe en este capítulo presenta las siguientes características:

- Es integral, dado que incorpora a todos los *stakeholders* de una misión satelital,
- Es detallada, dado que describe el diseño interno de los agentes del sistema,

- Se basa en el paradigma multiagente, en el cual cada elemento del sistema está representado por al menos un agente. Se utiliza el estándar FIPA para la comunicación entre ellos.
- Incluye una plataforma de software (SSEP) cuya función es facilitar y automatizar el comercio de servicios satelitales en tierra. El modelo de negocios de este elemento en particular, guarda similitud con el de otras plataformas de intercambio de bienes y servicios que pertenecen a otras industrias, con notables casos de éxito.

5.2 AgentSat

El AgentSat es el único elemento de la arquitectura que pertenece al segmento de vuelo. El término *AgentSat* será utilizado a lo largo de la tesis para referirse a cualquier satélite que posea las capacidades para operar dentro del SatSES. Asimismo, el *AgentSat Payload* es la carga útil que deberá incorporar un satélite cualquiera para suscribirse al sistema y operar en el mediante el intercambio de servicios satelitales con el resto de los elementos del sistema.

El AgentSat está pensado para ofrecer y/o solicitar los siguientes servicios en órbita:

- Tiempo de Instrumento,
- Descarga de datos a tierra,
- Mensajería en órbita,
- Procesamiento a bordo y
- Almacenamiento a bordo.

La arquitecta del AgentSat se presenta en la figura 5.4. La agrupación de más alto nivel que puede darse sobre un satélite es la de aquellos elementos que forman parte de la plataforma y aquellos que componen la carga útil del satélite. Esta división se muestra en la figura mediante los rectángulos de color blanco (*Payload* y *Platform/Bus*).

5.2.1 Plataforma del Satélite/AgentSat

La figura 5.4 sólo muestra dos de los subsistemas del satélite, aquellos relevantes al presente trabajo. Ellos son: (1) el subsistema de Comando y Manejo de Datos (*CDH Subsystem*) y (2) el subsistema de Comunicaciones (*Comm Subsystem*).

El subsistema de Comunicaciones del satélite cumple todas las funciones nominales de cualquier misión satelital (establecer un enlace con tierra para recibir telecomandos, enviar telemetría, descargar datos ciencia

de sus instrumentos, etc.), pero en este caso también se ocupará de recibir los telecomandos dirigidos al AgentSat (*AgentSatTelecommands*) y la información del mercado de intercambio (*ExchangeMarketData*), y de intercambiar mensajes (*AgentSatMessages*) con tierra.

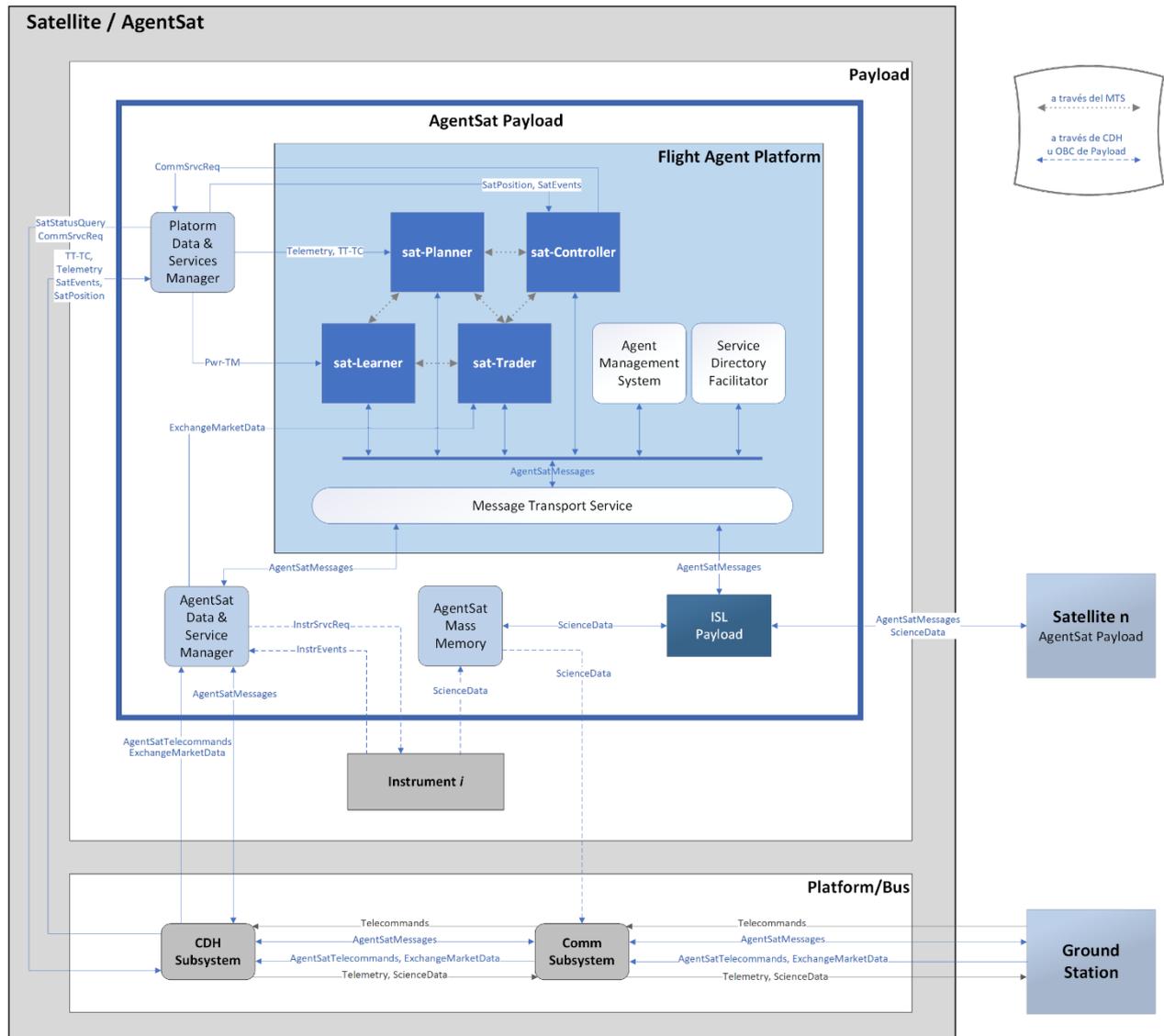


Figura 5.4 – Arquitectura del AgentSat

El subsistema CDH, además de cumplir con las funciones propias nominales, mantendrá dos interfaces con el AgentSat Payload, a través de las cuales llevará adelante las siguientes acciones:

- Interfaz con la unidad *Platform Data & Services Manager* (PDSM):
 - Recibirá consultas sobre el estado actual del satélite y las actividades de misión planificadas,
 - Recibirá solicitudes para el uso del servicio de comunicación con tierra,
 - Proporcionará telemetría de los subsistemas del satélite,

- Proporcionará información sobre los telecomandos Time-Tagged (TT-TC) activos,
 - Proporcionará información sobre los eventos del satélite,
 - Proporcionará información sobre la posición del satélite en órbita.
- Interfaz con el *AgentSat Data & Service Manager (ADSM)*:
 - Entregará los telecomandos dirigidos al AgentSat desde tierra
 - Entregará y recibirá mensajes AgentSat

5.2.2 AgentSat Payload

La arquitectura del *AgentSat Payload* se introduce en la figura 5.4 y está compuesta por los siguientes 5 elementos:

- i. Flight Agent Platform (FAP),
- ii. AgentSat Data & Service Manager (ADSM),
- iii. Platform Data & Services Manager (PDSM),
- iv. Inter-Satellite Links Payload (ISLP),
- v. AgentSat Mass Memory (AMM).

La plataforma agente de vuelo (FAP) es la infraestructura o plataforma en donde se alojarán los cuatro agentes del AgentSat junto con los tres servicios necesarios para su operación y la comunicación entre ellos. El diseño interno de los agentes del AgentSat se presenta en la sección 5.3 y la descripción de la plataforma y sus servicios en la sección 5.6.

El gestor de datos y servicio del AgentSat (ADSM) se ocupa de gestionar todo lo relativo a la configuración y datos del AgentSat y de actuar como interfaz entre los instrumentos del satélite y la plataforma de Agentes, y entre CDH y el MTS. Su diseño interno se describe en el inciso 5.2.3.

El gestor de datos y servicios de la plataforma (PDSM) es la unidad que actúa de interfaz entre la plataforma de agentes (FAP) y la plataforma del satélite. Su diseño interno se presenta en el inciso 5.2.4.

La carga útil para enlaces inter-satelitales (ISLP) es el sistema de hardware y software que permite intercambiar mensajes y datos con otros AgentSats en órbita. Se asume que este módulo se encuentra incorporado en todos los AgentSats.

La memoria masiva del AgentSat (AMM) tiene como objetivo almacenar temporalmente los datos ciencia relacionados con las órdenes de servicio del AgentSat. Estos datos ciencia incluyen aquellos transmitidos por otros satélites como también los generados por los propios instrumentos. Es preciso mencionar que para los datos ciencia propios podría utilizarse la memoria masiva de la plataforma. Si este fuera el caso, la

arquitectura del AgentSat no se verá afectada ya que la unidad AMM seguirá estando presente y la diferencia sólo radicará en los tipos de datos que allí se almacenan.

5.2.3 AgentSat Data & Service Manager

Las funciones principales del ADSM son tres: (i) gestionar la configuración del AgentSat en lo que refiere al estado del servicio de intercambio (modo y servicios habilitados) y el nivel de autonomía, a partir de lo definido en tierra, (ii) actuar de interfaz entre los instrumentos del satélite y la plataforma agente y (iii) actuar como el canal de transmisión de los mensajes que se intercambian entre CDH y los agentes. La arquitectura del ADSM se describe en la figura 5.4 y está compuesto por las siguientes cinco unidades de software:

- i. Service Manager (SVM)
- ii. Autonomy Manager (AUM)
- iii. Market Data Handler (MDH)
- iv. Instrument Data Handler (IDH)

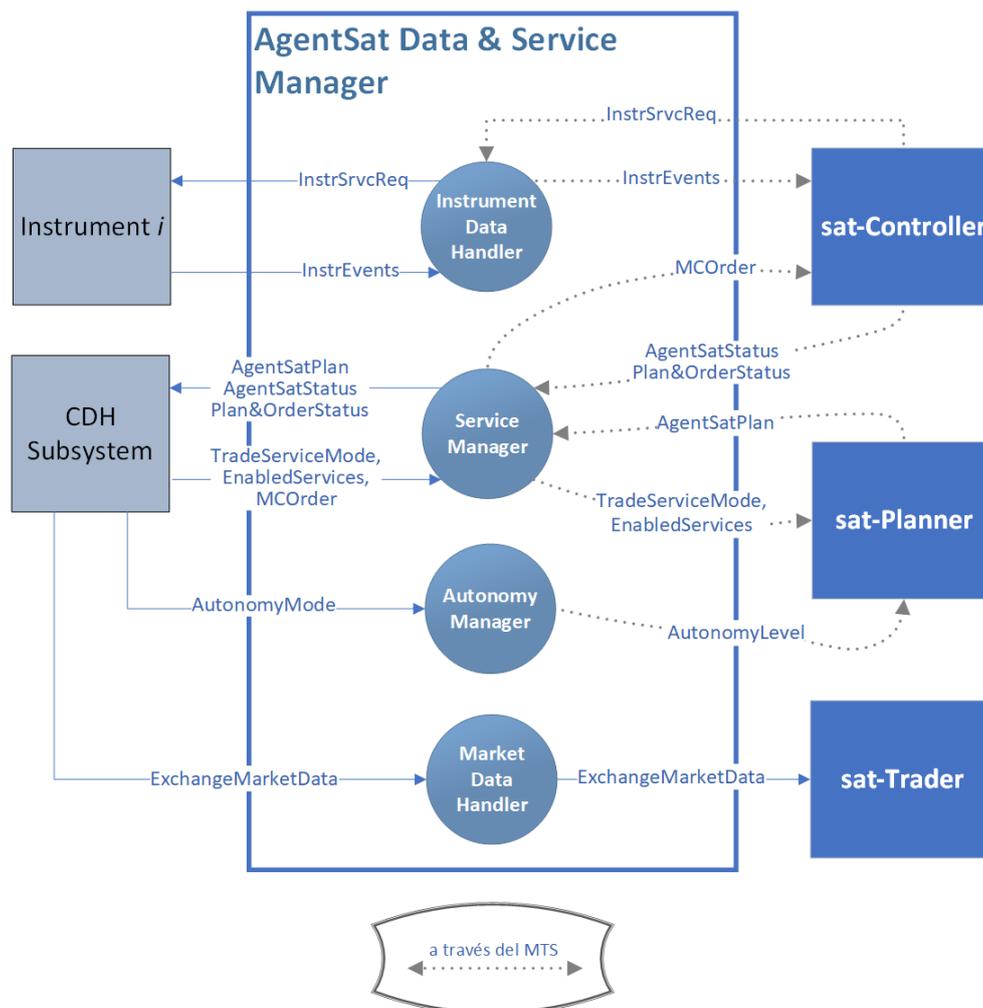


Figura 5.5 – Arquitectura del AgentSat Data & Service Manager (ADSM)

El gestor del servicio (SVM) se encarga de procesar aquellos telecomandos enviados desde tierra (y direccionados desde CDH) que contienen información sobre el modo del servicio (*TradeServiceMode*) y sobre los servicios habilitados para las órdenes de intercambio en órbita (*EnabledServices*). Será responsabilidad del SVM mantener esta información de configuración actualizada y notificar al agente sat-Planner. Asimismo, esta unidad recibirá las ordenes generadas en tierra por el MC y las entregará al agente sat-Controller para su ejecución. Finalmente, toda la información relacionada con los estados, órdenes y planes del AgentSat pasarán por el SVN, quien los entregará a CDH para su transmisión a tierra.

La tabla 5.1 resume los cinco posibles modos del servicio de intercambio del AgentSat. Siempre será responsabilidad del MOC que opera el satélite, el definir el modo en cada instante del tiempo y a lo largo del ciclo de vida de la misión. Tanto en los modos ON-S, ON-SB y ON-B, los servicios que se encuentran habilitados para la compra y/o venta en órbita serán definidos en tierra por el MOC y estarán identificados en el telecomando de configuración *EnabledServices*. Sólo los servicios habilitados serán tenidos en cuenta por el sat-Planner.

El gestor de autonomía del AgentSat (AUM) se encarga de procesar aquellos telecomandos enviados desde tierra (y direccionados desde CDH) que contienen información sobre la autonomía del AgentSat (*AutonomyMode*). Será responsabilidad del AUM mantener el modo del satélite actualizado y notificar al agente sat-Planner. Los modos de autonomía del AgentSat son dos y se definen en la tabla 5.2.

ID Modo	Detalle
OFF	Deshabilitado: El servicio de intercambio de servicios satelitales está deshabilitado. En esta situación, el satélite opera como si no estuviera suscripto al SatSES, no es un AgentSat.
ON-S	Habilitado sólo para Venta de Servicios: El AgentSat se encuentra activo, pero sólo está habilitada la venta de servicios a otros AgentSats, no la compra.
ON-SB	Habilitado para Venta y Compra de Servicios: El AgentSat se encuentra activo tanto para la venta y compra de servicios en órbita.
ON-B	Habilitado sólo para Compra de Servicios: El AgentSat se encuentra activo, pero sólo está habilitada la compra de servicios a otros AgentSats, no la venta.
ON-MC	Habilitado sólo para la Ejecución de órdenes definidas en MOC: El AgentSat se encuentra activo pero sólo responde a las órdenes de servicio que han sido acordadas en el SSEP por el MOC que lo representa, y enviadas desde tierra.

Tabla 5.1 – Modos del Servicio de Intercambio del AgentSat (*TradeServiceMode*)

ID Modo	Detalle
EXE	Ejecutor: En este modo, una vez finalizado el proceso de planificación de recursos ociosos, el plan de intercambio de servicios es enviado directamente al sat-Controller para dar inicio al mismo. Este modo no requiere ninguna instancia de aprobación o revisión del plan por parte del MOC.
AST	Asistente: En este modo, una vez finalizado el proceso de planificación de recursos ociosos, el plan de intercambio de servicios es enviado al MOC en tierra (a través de CDH) como una propuesta. Desde el MOC lo aprobarán o rechazarán y el AgentSat será notificado. El estado en el que quedará el AgentSat desde el momento en que envía la propuesta a tierra hasta que recibe la respuesta, es <i>Waiting</i> (En espera).

Tabla 5.2 – Modos de Autonomía del AgentSat (*AutonomyMode*)

El manejador de datos de mercado (MDH) cumple el rol de interfaz ya que recibe los datos de mercado de CDH y los deriva al agent sat-Trader alojado dentro del FAP.

El manejador de datos de los instrumentos de a bordo (IDH) es principalmente una interfaz entre el agente sat-Controller y los instrumentos. Esta unidad de software recibe las solicitudes de adquisición (a través de CDH o de la OBC del instrumento) por parte del agente y se lo comunica al instrumento. Del mismo modo, todo evento del instrumento relacionado con las órdenes de servicio del AgentSat (ej.: adquisición finalizada) es recibida por esta unidad y comunicada al agente.

5.2.4 Platform Data & Services Manager

La función del PDSM es actuar de interfaz entre el AgentSat y el subsistema CDH del satélite, de modo tal que el AgentSat pueda efectuar todas las solicitudes de servicio o datos necesarias y, al mismo tiempo, recibir información acerca del estado del satélite. La arquitectura del PDSM se describe en la figura 5.6 y está compuesto por dos unidades de software, a saber:

- i. Platform Requests Handler (PRH)
- ii. Platform Data Handler (PDH)

El manejador de solicitudes a plataforma (PRH) es la unidad que se encarga de realizar las consultas de telemetría a CDH y hacer las solicitudes de comunicación para descargar información a tierra. El proceso de pedido de telemetría presenta un ciclo de repetición configurable y es definido por el MOC.

El manejador de datos de plataforma (PDH) es la unidad que se encarga de recibir los datos de telemetría (en respuesta a las consultas realizadas por el PRH), los TT-TC activos, los datos de posición del satélite y todo tipo de eventos relevantes para la operación del AgentSat. Toda la información recibida por el PDH es derivada a los agentes correspondientes, ya que será requerida para la ejecución de sus actividades.

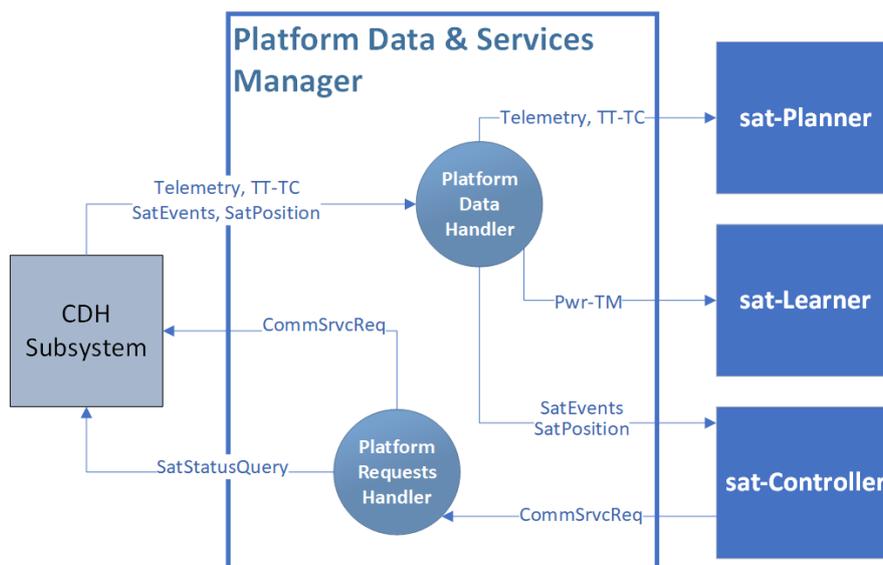


Figura 5.6 – Arquitectura del Platform Data & Services Manager (PDSM)

5.3 Agentes del AgentSat

Los agentes del AgentSat son cuatro y ya han sido introducidos en las figuras 5.3 y 5.4. En esta sección se presentará el diseño interno de cada uno de ellos: agente planificador (inciso 5.3.1), agente controlador (inciso 5.3.2), agente comercial (inciso 5.3.3) y agente aprendizaje (inciso 5.3.4). Para cada uno de ellos se detallará su arquitectura y sus objetivos.

5.3.1 Agente sat-Planner

El objetivo del planificador de a bordo del AgentSat es llevar a cabo la planificación de los recursos ociosos del satélite con el propósito de maximizar la utilidad de la misión. El agente sat-Planner está compuesto por las siguientes tres piezas de software:

- i. Current State Calculator (CSC)
- ii. Future State Estimator (FSE)
- iii. Mission Utility Maximizer (MUM)

Su arquitectura se describe en la figura 5.7 y el diagrama de actividad del proceso completo de planificación se muestra en la figura 5.8.

El calculador del estado actual del satélite (CSC) se encarga de, a partir de la telemetría recibida, computar el estado y las capacidades actuales de cada uno de los servicios que han sido habilitados para el servicio de intercambio en órbita, y la potencia eléctrica disponible. El resultado del cómputo se sintetiza en el producto *SatCurrentState*. El CSC es notificado de cualquier modificación en el modo del servicio de intercambio

(*TradeServiceMode*), ya que sólo ejecutará sus acciones si la planificación a bordo se encuentra habilitada (modos “ON-S”, “ON-SB”, “ON-B”). Si el modo estuviese configurado en “OFF” o “ON-MC”, no sólo no se disparará el computo de la capacidad ociosa, sino que el sat-Planner en general se mantendrá inactivo.

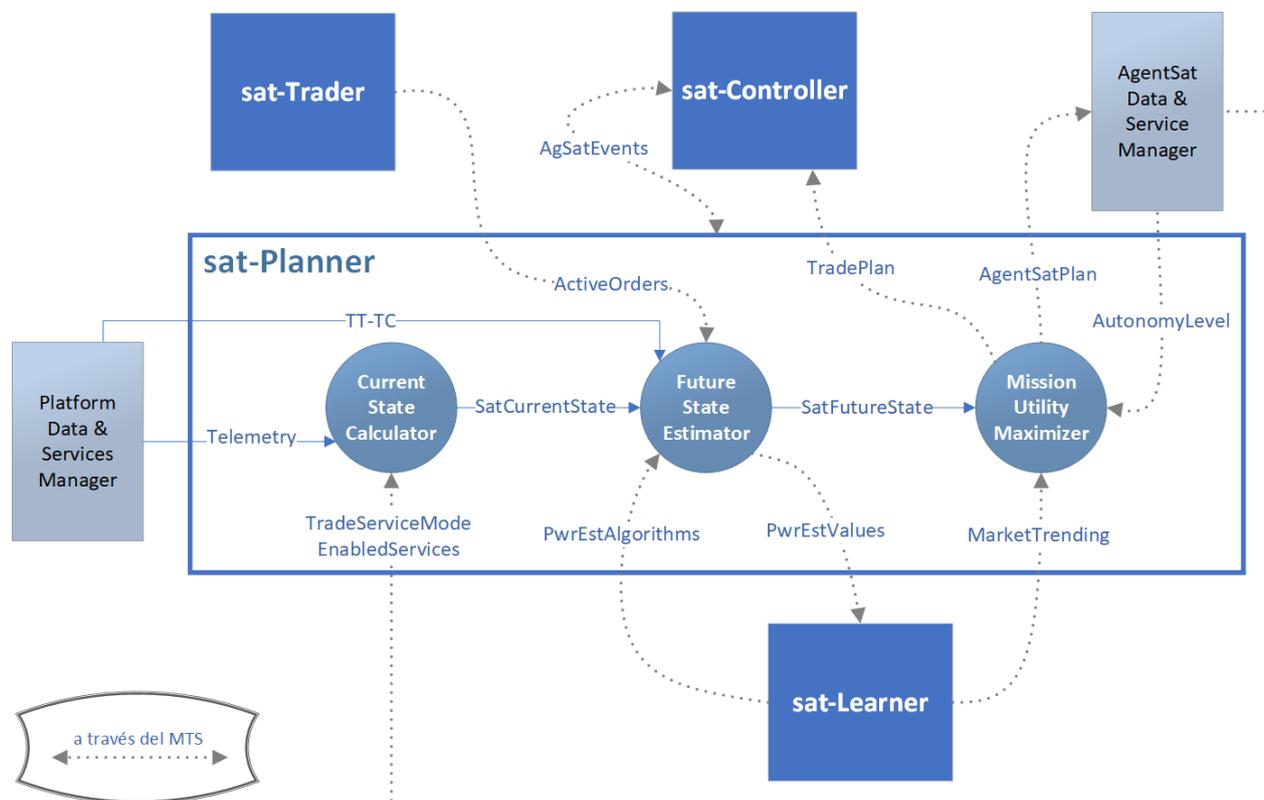


Figura 5.7 – Arquitectura del agente Planificador (*sat-Planner*)

El estimador del estado futuro del satélite (FSE) se encarga de computar la capacidad ociosa del satélite para el próximo ciclo, a partir del estado actual y de las actividades planificadas a futuro. El conjunto de elementos a ser tenidos en cuenta por el estimador incluye: [i] las actividades nominales de misión (definidas en tierra y expresadas en los TT-TC), [ii] las actividades del AgentSat que aún se encuentran activas (contempladas en las *ActiveOrders*) y [iii] el consumo energético de las actividades mencionadas y de las tareas de rutina del satélite. El resultado de la estimación se sintetiza en el producto *SatFutureState*. Asimismo, el FSE interactúa con el agente *sat-Learner*, el cual le irá proveyendo de estimadores de potencia eléctrica más confiables a lo largo de los ciclos.

El maximizador de utilidad de misión (MUM) se encarga de elaborar el plan de intercambio de recursos ociosos en órbita que genere la mayor utilidad para la misión. Para ello se basará en la capacidad futura del satélite (*SatFutureState*) y en la información del mercado de intercambio de tierra (*MarketTrending*). Por ejemplo, suponiendo que el satélite presenta capacidad ociosa de varios de sus instrumentos, una buena estrategia será ofrecer el servicio de *Tiempo de Instrumento* de aquel tipo de instrumento que es más valorado

(los clientes están dispuestos a pagar un precio mayor) o demandado por el mercado. Y para tomar esta decisión, el MUM sólo tendrá que analizar las tendencias del mercado incluidas en los informes periódicos que envía el SSEP desde tierra, y que llegan a él desde el sat-Learner. Finalmente, existen dos consideraciones importantes respecto a la elaboración de los planes de intercambio por parte del sat-Planner:

- i. Tipos de Planes: Los planes elaborados pueden incluir la venta de servicios, la compra de servicios o ambos. Las opciones que serán consideradas a priori dependerán del modo en el que esté configurado el servicio (OFF, ON-S, ON-SB, ON-B o ON-MC). Aunque lo más intuitivo es intentar vender servicios basados en aquellos recursos ociosos del satélite (modo ON-S), es importante mencionar que cabe la posibilidad de que resulte más barato comprar un determinado servicio aun cuando se pudiera adquirir con los recursos propios del satélite. Esto tendrá que ver con una evaluación detallada de los costos unitarios de cada operación y no es el objetivo de la tesis tratar este asunto. De todas formas, sea cual fuese el plan elaborado, está claro que la utilidad de ejecutarlo deberá como mínimo ser mayor a la utilidad de no ejecutar nada. Y para ello, no sólo es importante considerar los consumos de energía sino también los costos de oportunidad.
- ii. Nivel de Autonomía: El producto final del sat-Planner es el plan elaborado. Pero como ya se presentó en la tabla 5.2, dependiendo del modo de autonomía en el que se encuentre configurado el AgentSat, el plan será un (a) plan listo para ejecutar, o bien, (b) una propuesta de plan. El primer caso se corresponde con el AgentSat configurado como *Ejecutor*, donde el plan armado (*TradePlan*) es inmediatamente enviado al agente sat-Controller para dar inicio a su ejecución. El segundo caso se da cuando el AgentSat está configurado como *Asistente*, donde el plan (*AgentSatPlan*) será enviado a tierra para que sea evaluado por el MOC.

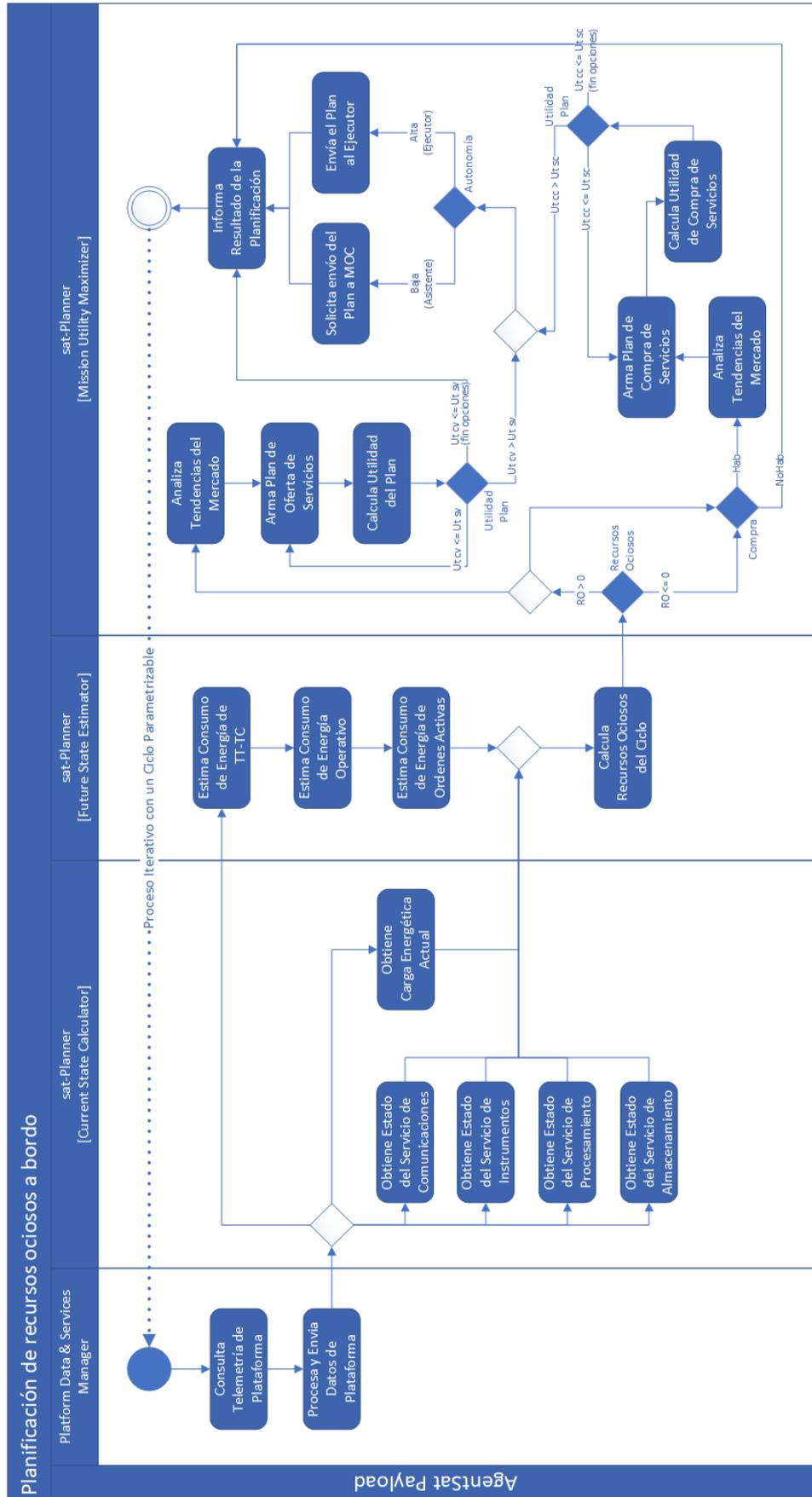


Figura 5.8 – Diagrama de Actividad del Proceso de Planificación del AgentSat

5.3.2 Agente sat-Controller

El objetivo del agente controlador del AgentSat es ejecutar en tiempo y forma los planes elaborados por el planificador, monitorear los eventos externos e internos, y mantener actualizados los estados del servicio y de las ordenes de intercambio. La arquitectura del sat-Controller se describe en la figura 5.9 y está compuesto por tres unidades de software, a saber:

- i. Event & Status Monitor (ESM)
- ii. Plan Executor (PEX)
- iii. Real-time Propagator (RTP)

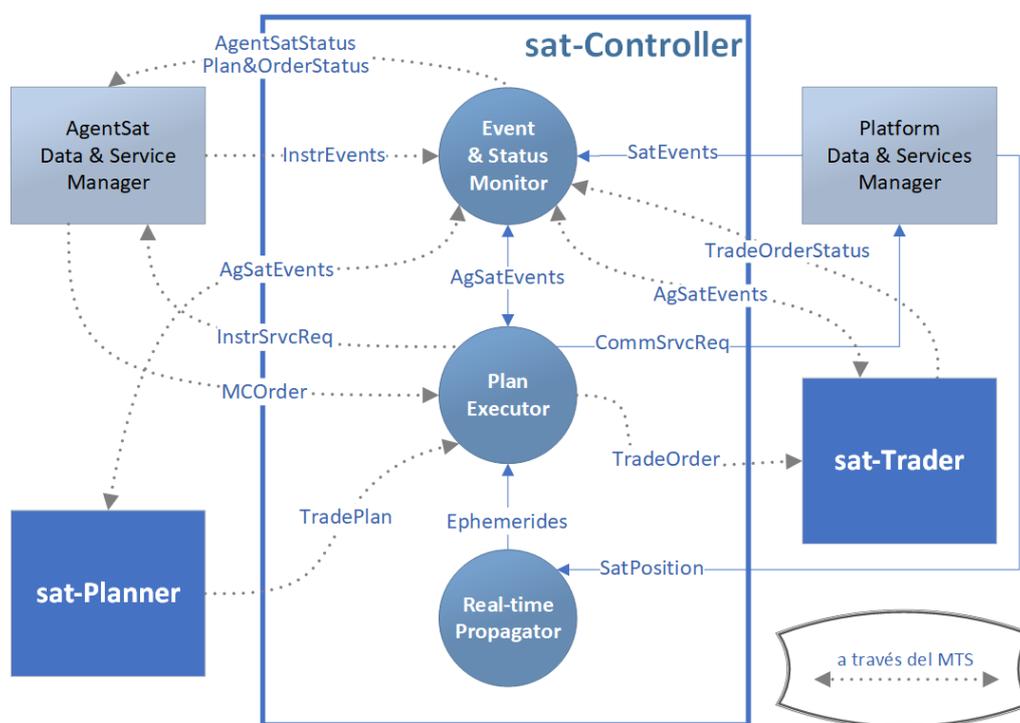


Figura 5.9 – Arquitectura del agente Controlador (*sat-Controller*)

El monitor de eventos y estados (ESM) se encarga de centralizar la gestión de los eventos de la plataforma (*SatEvents*), de los instrumentos (*InstrEvents*) y del AgentSat (*AgSatEvents*), y de mantener actualizado el estado del AgentSat (tabla 5.3). Asimismo, el ESM es el responsable de reunir toda la información de estados y notificar a tierra oportunamente. El producto *Plan&OrderStatus* contiene información actualizada sobre la última acción ejecutada por el plan y el estado de la orden de intercambio, si es que incluye.

Estado	Detalle
Disabled	Deshabilitado: Es el estado que se corresponde con el modo OFF del servicio de intercambio.
Planning	En Planificación: estado que se corresponde con los períodos en los que el agente sat-Planner se encuentra activo, es decir, durante el cómputo de estado actual del satélite, estimación de recursos ociosos y elaboración de los planes de intercambio.
Executing	En Ejecución: estado que se corresponde con cualquier fase de ejecución del plan, que no incluye la etapa del intercambio de servicios.
Trading	En Intercambio: estado que se corresponde con los períodos de actividad del agente sat-Trader. Más concretamente, el AgentSat se encuentra en este estado en la medida que el protocolo de intercambios de servicios en órbita se encuentre en ejecución.
IDLE	Inactivo: el AgentSat entra en este estado cuando el agente sat-Planner no ha podido elaborar ningún plan viable. El estado se mantiene hasta que inicie el próximo ciclo de planificación.
WFA	En Espera de Acción: estado que se corresponde con aquellas actividades incluidas en el plan pero que se escapan del ámbito del <i>AgentSat Payload</i> o del ámbito del propio satélite. Por ejemplo, si el plan incluye la adquisición por parte de un instrumento a bordo sobre una determinada zona, y ya se ha enviado la orden al instrumento, pero el satélite aún no se encuentra en la posición indicada, el estado del AgentSat se encuentra en este estado, esperando la adquisición. Lo mismo ocurre si una orden de intercambio incluye una acción de otro elemento de la arquitectura (ej.: otro AgentSat). En este caso, el estado del AgentSat se encuentra a la espera de una acción de terceros.
WFR	En Espera de Respuesta: estado que se corresponde con el modo de autonomía <i>asistente</i> . Bajo este modo, una vez que ha finalizado el proceso de planificación, el plan es enviado al MOC para su aprobación. Durante ese período de tiempo en el que el AgentSat queda en espera de la respuesta, el estado del AgentSat se encuentra en WFR.

Tabla 5.3 – Estados del AgentSat (*AgentSatStatus*)

El ejecutor de planes (PEX) se encarga de controlar el flujo de ejecución de los planes elaborados por el planificador (*TradePlan*) o por el MOC en tierra (MCOOrder). Es el PEX quien emite las solicitudes de uso de instrumentos (InstrSrvReq), las solicitudes de comunicación a tierra (CommSrvReq) y las ordenes de intercambio en órbita (*TradeOrder*) de acuerdo a lo definido en el plan. Además, el PEX requiere en algunas circunstancias conocer la posición del satélite para ejecutar una determinada acción del plan (ej.: adquirir sobre una determinada ubicación geográfica), y para ello cuenta con una interfaz con una unidad interna, por medio del cual recibe las efemérides.

El propagador de tiempo real (RTP) se encarga de computar las efemérides del satélite en base a los datos de posición y velocidad recibidos de la plataforma. La precisión de las efemérides dependerá del modelo

utilizado y de las fuentes de datos que se usarán para los cálculos. Los detalles técnicos del proceso de determinación de órbita quedan fuera del alcance de la tesis; a los fines del presente trabajo, bastará saber que la arquitectura del AgentSat contempla la inclusión de una unidad capaz de propagar la órbita.

5.3.3 Agente sat-Trader

El objetivo del agente comercial del AgentSat es establecer comunicación con otros AgentSats en órbita para cerrar acuerdos de servicios satélites, en base a lo indicado por los planes de intercambios previamente generados. La transmisión de mensajes entre dos AgentSats se enmarca dentro del protocolo de interacción IOSSE (inciso 5.9.1) definido ad-hoc para la arquitectura en cuestión y se implementa a través del *Payload ISL*. La arquitectura del sat-Trader se describe en la figura 5.10 y está compuesto por las siguientes tres piezas de software:

- i. Broker
- ii. Orders Manager (ORM)
- iii. Market Data Manager (MDM)

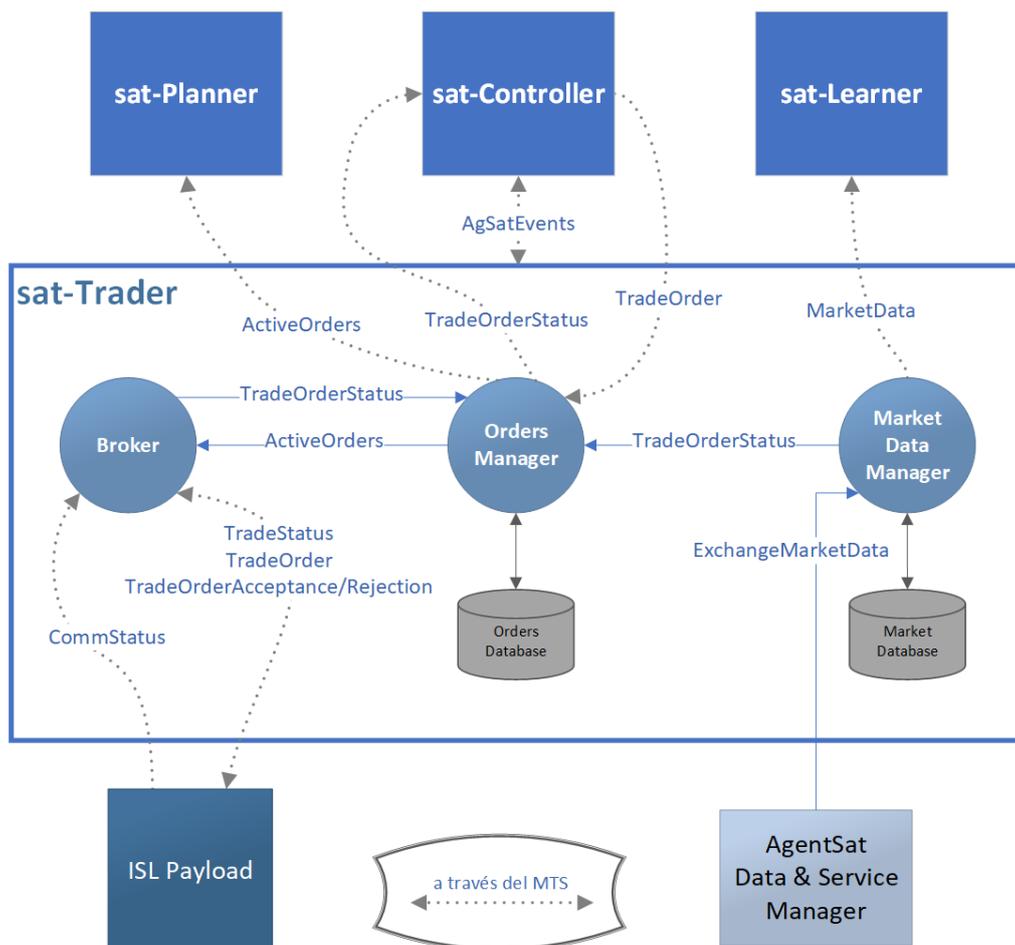


Figura 5.10 – Arquitectura del agente Comercial (sat-Trader)

El *broker* es el encargado de la ejecución del protocolo de interacción IOSSE por cada orden de intercambio activa, en cada oportunidad que se establezca un enlace de comunicación con otro AgentSat en órbita. Previo a la ejecución del protocolo, el *broker* se ocupa también de crear todos los mensajes asociados a las órdenes de intercambio activas. El protocolo de interacción IOSSE junto con los mensajes intercambiados se describen en el inciso 5.9.1.

El gestor de órdenes (ORM) se encarga del alta de las órdenes de intercambio en el repositorio (*Orders Database*) y de la actualización de sus estados (al finalizar el protocolo de interacción IOSSE o bien, en base al último estado de las transacciones enviado desde tierra -contenido en el producto *ExchangeMarketData*). El ORM también deberá informarle al sat-Planner acerca de las órdenes de intercambio activas para que el planificador a bordo las tenga en consideración. Los posibles estados de las órdenes de intercambio se definen en la tabla 5.4.

Estado	Detalle
Planned	Planificada: cuando la orden fue dada de alta, pero aún no se activó porque depende de una precondition (de tiempo o de algún evento que no ocurrió todavía).
Active	Activa: cuando la orden ha pasado a manos del <i>broker</i> a la espera de ser ofrecida a otro agente para establecer un acuerdo de servicio.
Active-TRD	Activa y Comercializada: cuando se ha establecido un acuerdo con otro agente para la ejecución de la orden.
Active-EXE	Activa y en Ejecución: cuando la orden se encuentra en ejecución por el propio AgentSat.
Active-WFA	Activa y esperando Acción: cuando la orden se encuentra en pausa a la espera de una acción externa al AgentSat.
Finished	Finalizada: cuando todas las actividades incluidas en la orden han sido ejecutadas exitosamente.
Aborted	Abortada: cuando la ejecución de la orden ha sido interrumpida y cancelada.
Rated	Calificada: cuando el proveedor ha finalizado su servicio y/o entregado el producto y el cliente ha calificado la orden de servicio.
Closed	Cerrada: cuando, luego de haber sido calificada, se ha emitido la liquidación final de la orden para enviar a los agentes involucrados.

Tabla 5.4 – Estados de la Orden de Intercambio (*TradeOrderStatus*)

El gestor de datos de mercado (MDM) se encarga del procesamiento, almacenamiento y mantenimiento de los datos históricos de mercado (informes periódicos, estados de órdenes, etc.) enviados desde el SSEP. El MDM es responsable de la provisión de dicha información al agente aprendiz cuando éste la requiera.

5.3.4 Agente sat-Learner

El objetivo del agente aprendiz del AgentSat es ejecutar todas las tareas de aprendizaje vinculadas al AgentSat para [i] mejorar los algoritmos de estimación y [ii] extraer información no trivial de los informes de mercado provistos por el SSEP. La arquitectura del sat-Learner se describe en la figura 5.11 y está compuesta por dos piezas de software:

- i. Power Estimation Apprentice (PEA)
- ii. Market Researcher (MKR)

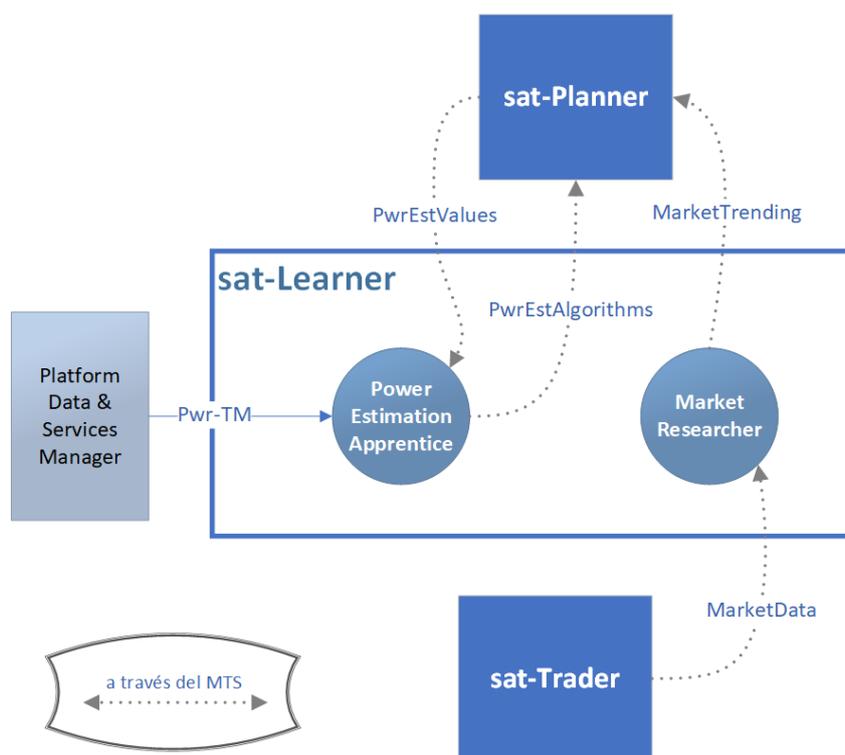


Figura 5.11 – Arquitectura del agente Aprendiz (sat-Learner)

El aprendiz de estimación de potencia (PEA) se encarga de la generación de algoritmos para estimar la potencia que consumirá cada instancia de utilización de instrumentos, subsistemas, tareas operativas de la plataforma y las actividades del AgentSat. Los algoritmos están en constante revisión ya que al final de cada ciclo se retroalimenta con los valores reales de aquellos que han sido estimados. De esta forma, cada ciclo de planificación genera más datos para generar versiones de algoritmos más correctos.

El analista de mercado (MKR) se encargará del procesamiento y de la extracción de información no trivial a partir de la información de mercado provista por tierra. Contar con información desconocida a priori mejorará la toma de decisiones durante el proceso de planificación, resultando en la elección de planes con mayores utilidades. Aunque se prevé el empleo de técnicas de minería de datos y *machine learning* dentro

de esta unidad, no forma parte del alcance de la tesis entrar en una discusión más profunda sobre este tema; sólo basta con su mención para ser tenida en cuenta en una futura implementación.

5.3.5 Interfaces externas del AgentSat

El AgentSat intercambiará información con dos elementos de la arquitectura: (i) otros AgentSats y (ii) Estaciones Terrenas. El resumen gráfico de los productos intercambiados con esos elementos se presenta en la figura 5.12 y la ontología de la información intercambiada en la figura 5.13.

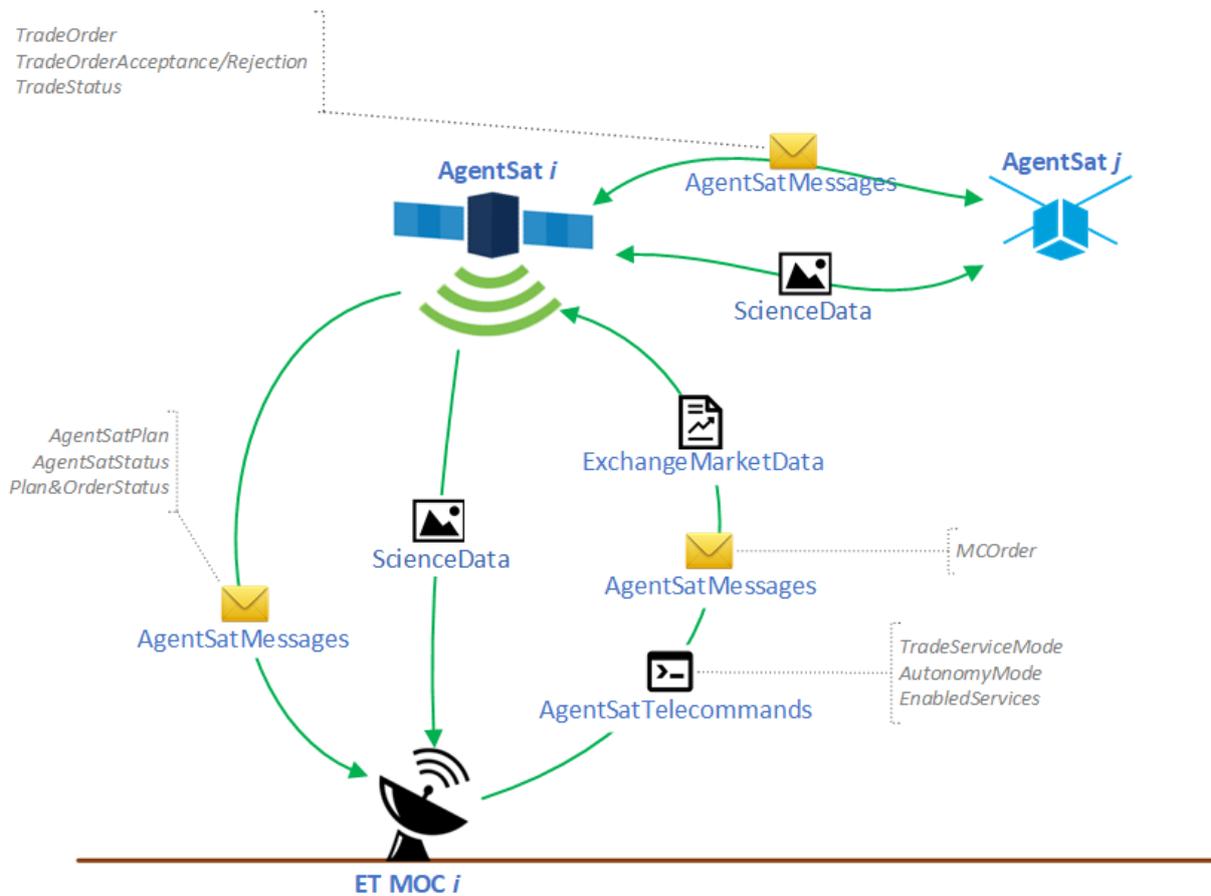


Figura 5.12 – Esquema de las Interfaces Externas del AgentSat

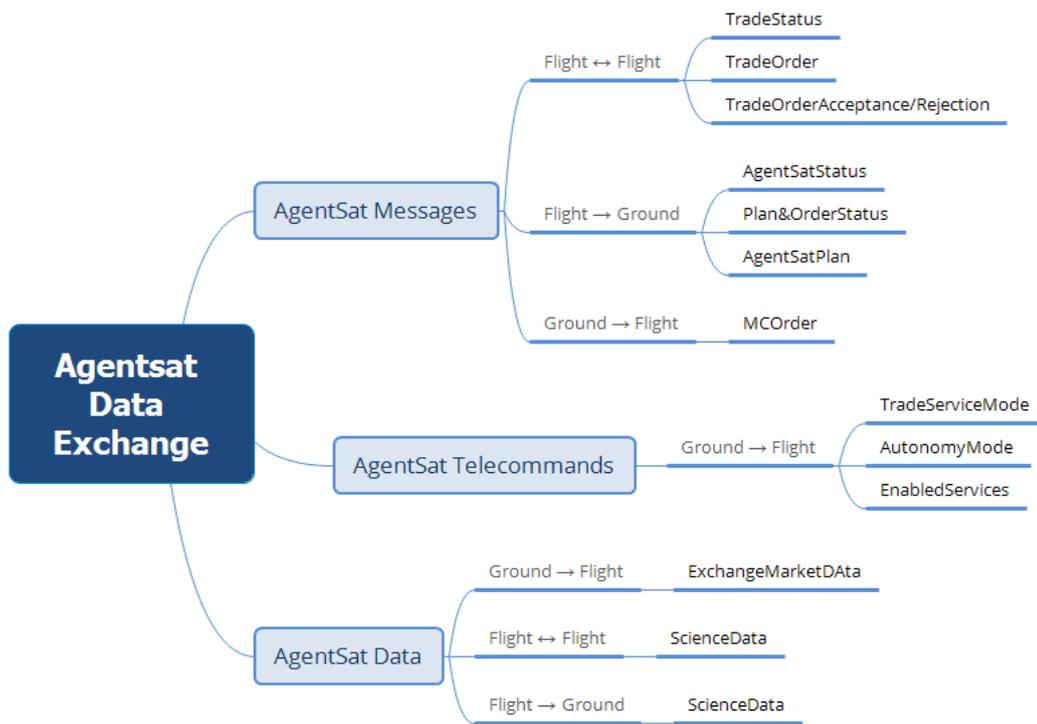


Figura 5.13 – Ontología de la Información intercambiada por el AgentSat

5.4 Plataforma de Intercambio de Servicios Satelitales (SSEP)

La Plataforma de Intercambio de Servicios Satelitales (SSEP, por sus siglas en inglés) es un sistema que ofrece una plataforma de software dedicado al comercio electrónico de recursos satelitales ociosos. Sistemas similares al SSEP existen en otras industrias y, en muchos casos, con un éxito rotundo en los correspondientes mercados. Basta con citar dos casos paradigmáticos como *Airbnb* (Guttentag, 2015) en la industria del alojamiento y *Uber* en la industria del transporte privado.

El SSEP busca centralizar en una única plataforma web todas las ofertas y solicitudes de servicios satelitales, de todos los actores: usuarios finales, MOCs, estaciones terrenas y potenciales proveedores de servicios informáticos de gran utilidad para la industria espacial.

La arquitecta del SSEP se presenta en la figura 5.14 y está compuesta por los siguientes 6 elementos:

- i. User Services (USS),
- ii. Mission Center Services (MCS),
- iii. Ground Station Services (GSS),
- iv. Provider Services (PVS),
- v. Product Manager (PRM),
- vi. Ground Agent Platfrom (GAP).

Los módulos de servicios de los tres clientes (Usuario -USS-, Centros de Misión -MCS- y Estaciones Terrenas -GSS-) comparten la misma estructura interna y prácticamente actúan como interfaces para la administración del sistema, configuración del agente y monitoreo de estado de las órdenes. En términos generales se han incluido, además de la interfaz propiamente dicha, dos unidades de software genéricas para cada uno de los tres módulos:

- i. Profile & Orders Manager (POM): unidad encargada de gestionar el perfil de usuario, de gestionar las ABM de las solicitudes del cliente y la parametrización del agente correspondiente.
- ii. Data & Messages Manager (DMM): unidad encargada de recibir los mensajes y eventos, los datos del mercado e imágenes, y entregarlos al cliente. En el caso particular de las estaciones terrenas, el DMM no gestiona ni imágenes ni mensajes.

El módulo de servicios de los proveedores (PVS) es un elemento que ha sido considerado en la arquitectura, pero su diseño interno no ha sido el foco de la tesis. Su incorporación se debe al incremento sustancial que se ha visto en los últimos años en lo que respecta a las capacidades de procesamiento y almacenamiento en la nube (Ghaffar y Vu, 2015). De tal modo, se ha considerado que para la industria satelital sería sumamente útil que los proveedores de tales servicios (ej.: AWS de Amazon, [Amazon, 2019]) puedan ofrecer sus servicios a través del SSEP.

El gestor de productos (PRM) es un módulo que se encarga de centralizar la gestión de aquellos productos que ingresan y salen del SSEP que no son mensajes. Incluye dos unidades de software:

- i. el gestor de imágenes (IMM), encargado de la recepción, almacenamiento de las imágenes y entrega de las imágenes que ingresan al SSEP, dirigidas a los usuarios y centros de misión,
- ii. el manejador de informes de mercado (MRH), encargado del envío de los informes a las estaciones terrenas, dirigidas a los AgentSats en órbita.

La plataforma agente de tierra (GAP) es la infraestructura o plataforma en donde se alojarán los cinco tipos de agentes que representan a los usuarios del sistema, junto con los tres servicios necesarios para su operación y la comunicación entre ellos. El diseño interno de los agentes del SSEP se presenta en la sección 5.5 y la descripción de la plataforma y sus servicios en la sección 5.6.

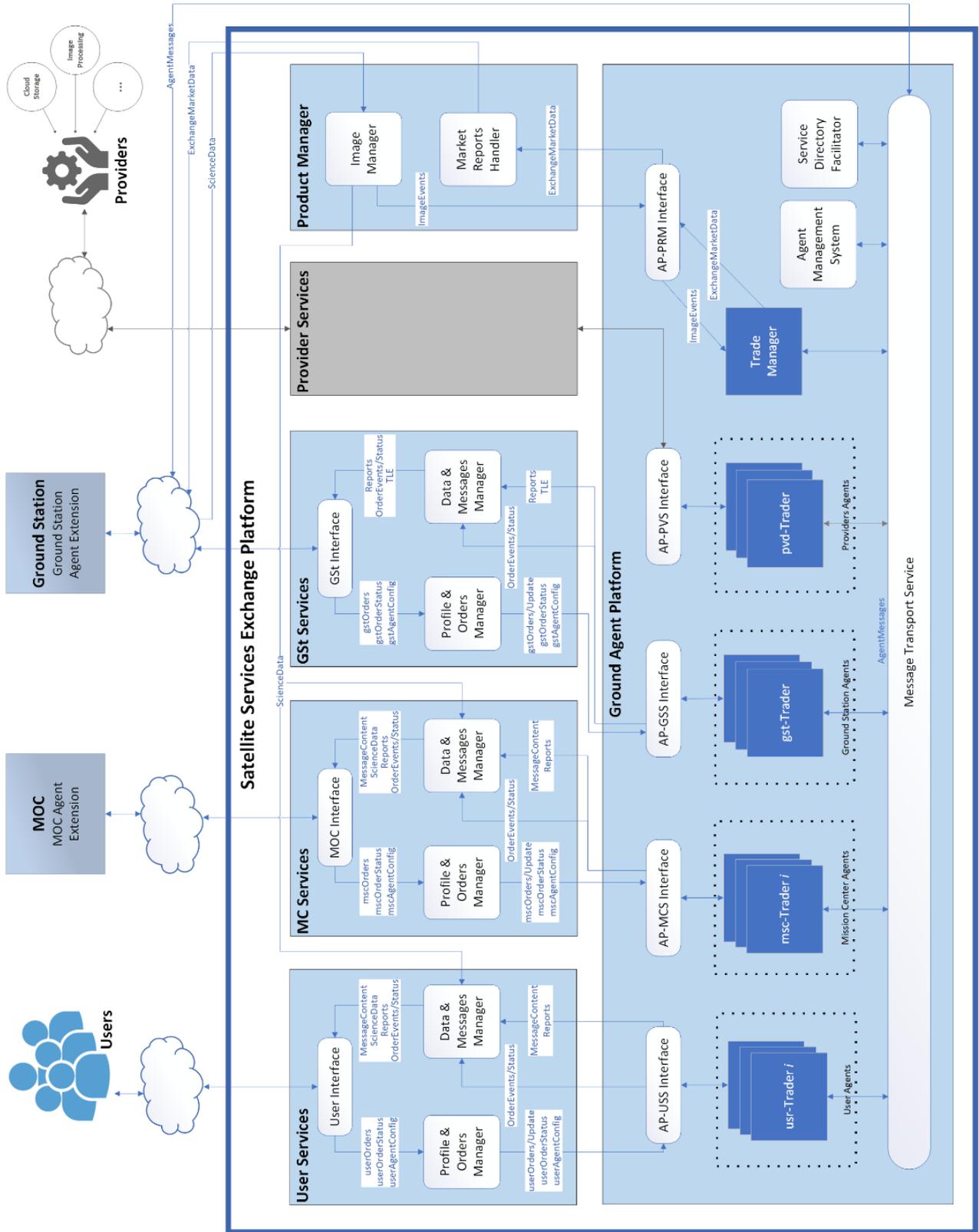


Figura 5.14 – Arquitectura del SSEP

5.5 Agentes del SSEP

Son cinco las clases de agentes del SSEP y ya han sido introducidos en las figuras 5.3 y 5.14. En esta sección se presentará el diseño interno de cada uno de ellos: agente del usuario (inciso 5.5.1), agente del centro de misión (inciso 5.5.2), agente de la estación terrena (5.5.3), agente del proveedor (5.5.4) y el agente administrador del comercio en la plataforma (inciso 5.5.5). Para cada uno de ellos se detallará su arquitectura y sus objetivos. Cada usuario, centro de misión, estación terrena y proveedor tendrá una instancia de su correspondiente clase de agente y podrá parametrizarlo desde su módulo de servicios.

5.5.1 Agente usr-Trader

El agente comercial del usuario (usr-Trader) es el representante del usuario en el mercado de servicios que se desarrolla en la plataforma agente. Su objetivo es gestionar las órdenes que han sido creadas por el usuario (a través del USS) y comunicarlas oportunamente a la comunidad de agentes para lograr cerrar los acuerdos de servicio requeridos. De igual manera, el usr-Trader es responsable de monitorear el estado y los eventos relacionadas a las órdenes que se encuentran activas. La arquitectura del usr-Trader se describe en la figura 5.15 y está compuesto por cuatro unidades de software, a saber:

- i. Orders Manager (ORM)
- ii. Orders Scheduler (ORS)
- iii. Broker
- iv. Report Handler (RPH)

El gestor de órdenes (ORM) se encarga del alta y modificación de las órdenes de intercambio en el repositorio (*User Orders Database*) y la actualización de sus estados. Los posibles estados de las órdenes de intercambio son los mismos que ya han sido definidos en la tabla 5.4.

El programador de órdenes (ORS) se encarga de monitorear el repositorio de órdenes generadas y notificar al *broker* de aquellas que se han activado y requiere ser comercializadas.

El *Broker* es el encargado de la ejecución de los protocolos de interacción (sección 5.9) por cada orden de intercambio activa. Previo a la ejecución del protocolo, el *broker* se ocupa de crear todos los mensajes asociados a las órdenes de intercambio activas y de consultar las calificaciones de los potenciales proveedores del servicio cuando sea necesario. Una vez que ha sido comercializada la orden, el *broker* también se encarga de decodificar los mensajes recibidos y monitorear los cambios de estados de la misma y sus eventos relacionados, para comunicarlos a los destinatarios correspondientes.

El manejador de informes del agente (RPH) es la interfaz entre el *Trade Manager* (TM) y el USS, a través de la cual se transfieren las distintas clases de informes que emite periódicamente el SSEP, cuyos destinatarios finales son los usuarios.

ofreciendo el servicio de adquisición con un instrumento de su AgentSat, deberá recibir del agente del usuario las coordenadas geográficas específicas para la adquisición.

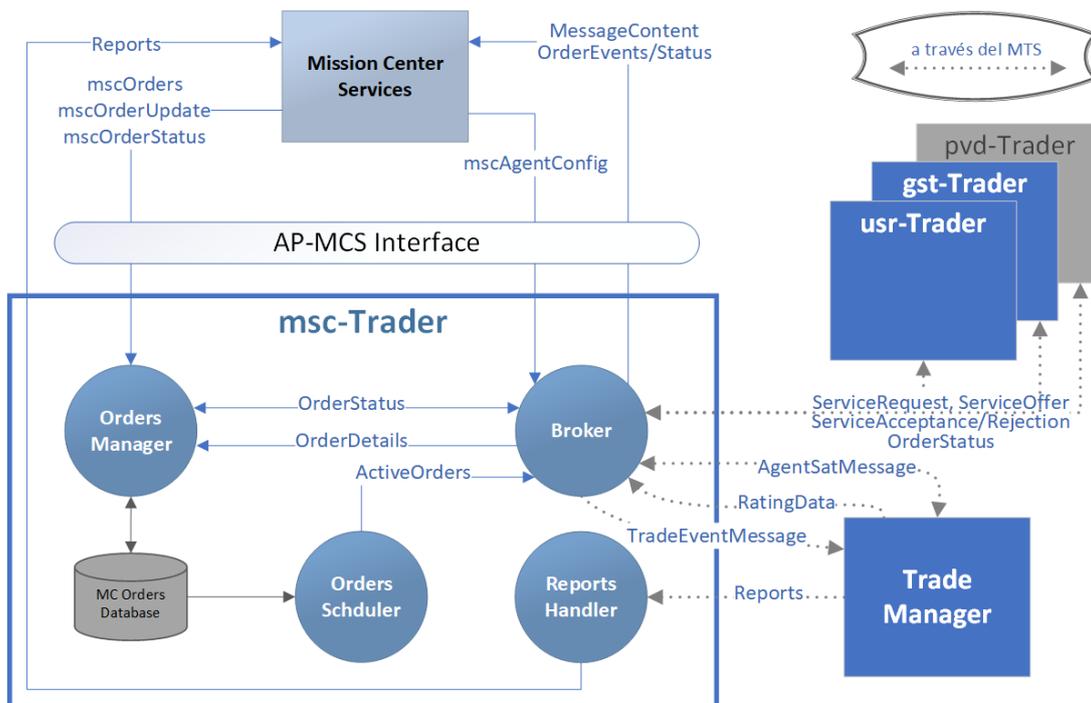


Figura 5.16 – Arquitectura del agente comercial del MOC (msc-Trader)

Por último, es preciso tener en cuenta dos cuestiones importantes:

- El msc-Trader no sólo es el representante del MOC en el SSEP sino, lo que es más importante, es el representante de su/sus AgentSat/s en tierra. Esto significa que al margen de cualquier otro servicio que pueda ofertar el MOC en la plataforma, los más importantes son las capacidades ociosas de su AgentSat, lo cual sucederá cuando el *TradeServiceMode* del AgentSat esté configurado en ON-MC (habilitado sólo para la Ejecución de órdenes definidas en MOC).
- En caso que un MOC sea propietario u operador de una o varias ETs, y quiera ofrecer los servicios de antena a través del SSEP, podrá hacerlo. Sin embargo, la arquitectura fue diseñada para que cada ET (al margen de quien controle u opere sus servicios) se dé de alta de forma independiente en el sistema y que ofrezca sus servicios con el agente asignado para tal fin: el gst-Trader (inciso 5.5.3). Esto tiene que ver con decisiones de diseño que buscan alcanzar un alto grado de *cohesión* en sus elementos de arquitectura, y, por ende, de robustez del sistema.

5.5.3 Agente gst-Trader

El agente comercial del centro de la estación terrena (gst-Trader) es el representante de la ET en el mercado de servicios que se desarrolla en la plataforma agente. Su objetivo es gestionar las órdenes que han sido creadas por la ET (a través del GSS) y comunicarlas oportunamente a la comunidad de agentes para lograr cerrar los acuerdos de servicio requeridos. De igual manera, el gst-Trader es responsable de monitorear el estado y los eventos relacionadas a las órdenes que se encuentran activas. La arquitectura del gst-Trader se describe en la figura 5.17.

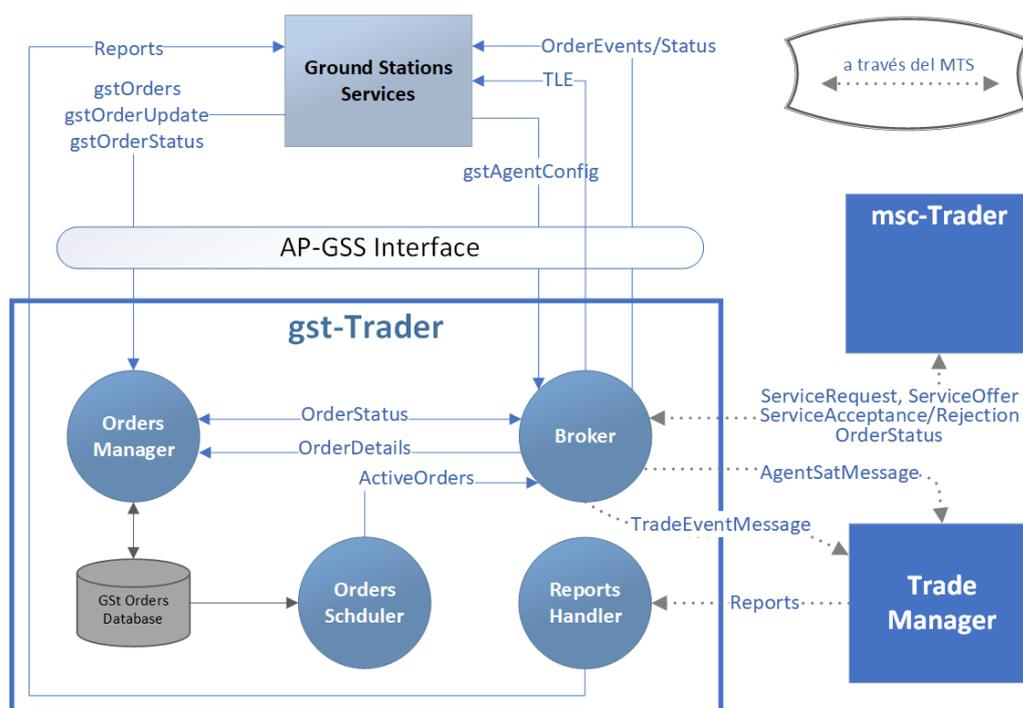


Figura 5.17 – Arquitectura del agente comercial de la Estación Terrena (gst-Trader)

La arquitectura y las unidades de software que componen el gst-Trader son exactamente iguales a las del agente comercial del usuario (inciso 5.5.1) y de los otros dos agentes comerciales (MOC y proveedor de servicios). Las únicas diferencias entre el gst-Trader y el usr-Trader son:

- El *broker* no decodifica los *AgentSatMessages* ya que sólo los envía al TM, no recibe.
- Una vez que una orden de oferta de servicio se cierra, el *bróker* no sólo deberá informar acerca del nuevo estado de la orden al ORM, sino que también deberá proveerle los detalles necesarios (*OrderDetails*) para que la misma pueda ser ejecutada. Por ejemplo, si el agente de la ET está ofreciendo servicio de antena para una determinada frecuencia, deberá recibir del agente del MOC el TLE actualizado del satélite para poder seguirlo y comunicarse con él.
- El gst-Trader sólo comercia con agentes comerciales de MOCs.

5.5.4 Agente pvd-Trader

El agente comercial del proveedor de servicio (pvd-Trader) es el representante del proveedor en el mercado de servicios que se desarrolla en la plataforma agente. Su objetivo es gestionar las órdenes que han sido creadas por el proveedor (a través del PVS) y comunicarlas oportunamente a la comunidad de agentes para lograr cerrar los acuerdos de servicio requeridos. De igual manera, el pvd-Trader es responsable de monitorear el estado y los eventos relacionadas a las órdenes que se encuentran activas. La arquitectura del pvd-Trader se describe en la figura 5.18.

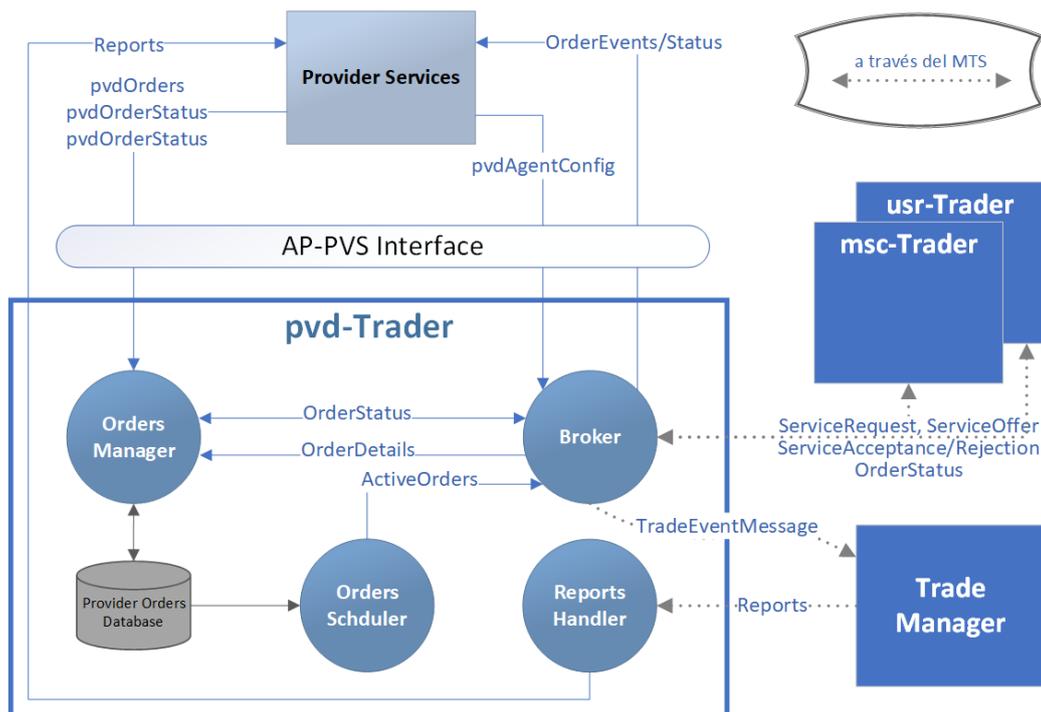


Figura 5.18 – Arquitectura del agente comercial del Proveedor de Servicios (pvd-Trader)

La arquitectura y las unidades de software que componen el pvd-Trader son exactamente iguales a las del agente comercial del usuario (inciso 5.5.1) y de los otros dos agentes comerciales (ET y MOC). Las únicas diferencias entre el pvd-Trader y el usr-Trader son:

- El pvd-Trader no presenta una interfaz para recibir ni entregar *AgentSatMessages* ya que sólo ofrece servicios en tierra a los usuarios y a los centros de misión.
- El pvd-Trader comercia con dos clases de agentes comerciales: de usuarios y de MOCs.
- Una vez que una orden de oferta de servicio se cierra, el *broker* no sólo deberá informar acerca del nuevo estado de la orden al ORM, sino que también deberá proveerle los detalles necesarios (*OrderDetails*) para que la misma pueda ser ejecutada. Por ejemplo, si el agente del proveedor está ofreciendo un servicio de procesamiento de imágenes con su propia infraestructura tecnológica,

deberá recibir del agente del usuario todas las especificaciones y datos auxiliares que sean necesarios conocer para llevar a cabo el procesamiento.

5.5.5 Agente Trade Manager

El administrador del comercio del SSEP (TM) es el agente responsable de ejecutar todas aquellas actividades relacionadas con las operaciones comerciales de la plataforma y, que son vistas como servicios para el resto de los agentes del sistema. Esto incluye la supervisión de todas las órdenes transaccionadas en el sistema, el mantenimiento de los estados de cuenta de los clientes, la elaboración de informes y el procesamiento de las calificaciones de los clientes. Además de estas responsabilidades, el TM actúa como interfaz de entrada y salida para todo mensaje AgentSat que ingresa y sale del SSEP. La arquitectura del *Trade Manager* se describe en la figura 5.19 y está compuesto por las siguientes cuatro unidades de software:

- i. Transactions Manager (TRM)
- ii. Accounts Manager (ACM)
- iii. Rating Manager (RTM)
- iv. Reporting Manager (RPM)

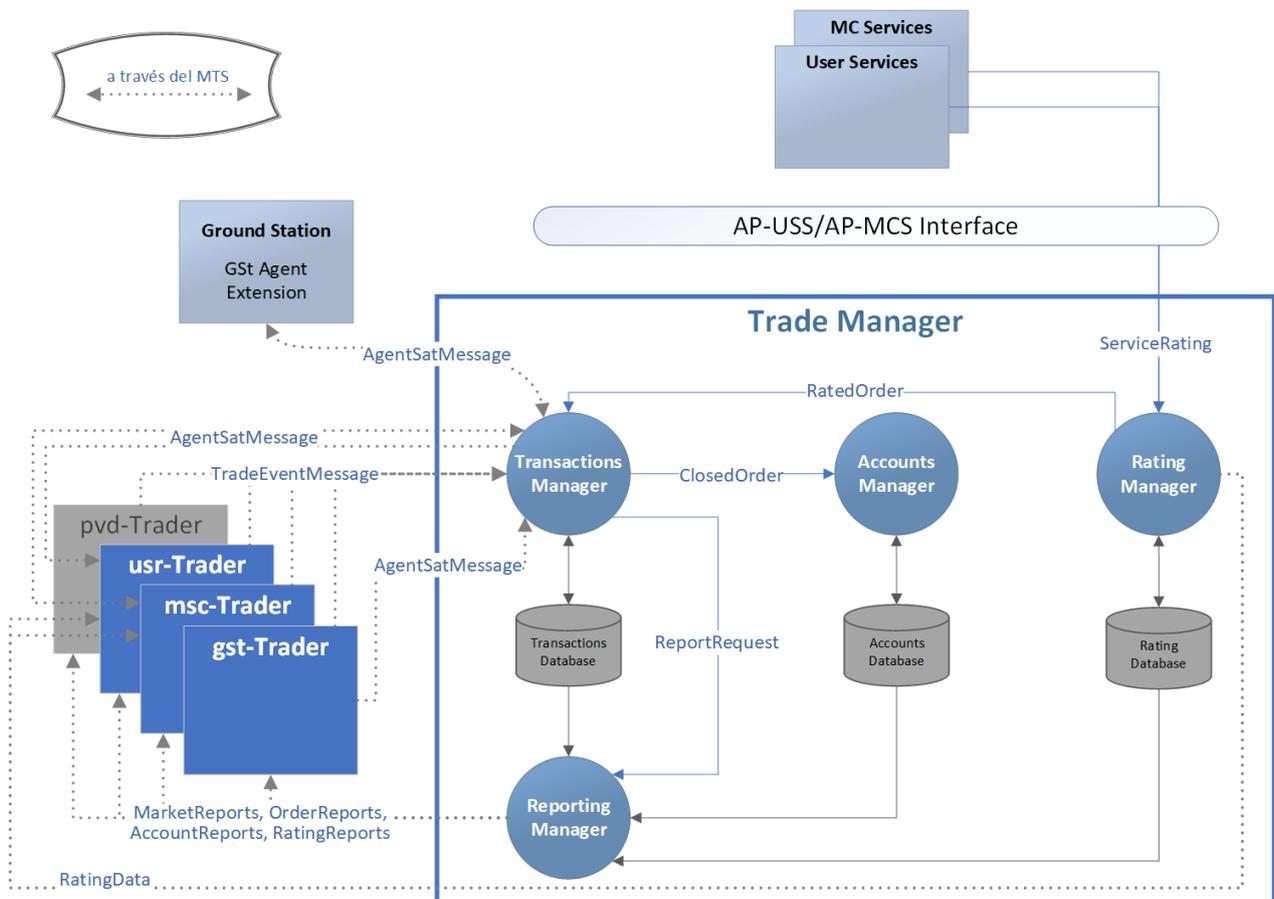


Figura 5.19 – Arquitectura del agente administrador del comercio en el SSEP (Trade Manager)

El administrador de transacciones (TRM) se encarga de la recepción y entrega de todos los mensajes *AgentSatMessages* que ingresan o salen del SSEP desde y hacia las estaciones terrenas. Además, es el responsable registrar cada nueva operación, estado o evento relacionado con las ordenes transaccionadas por los agentes del sistema y de mantener actualizado la base de datos de transacciones.

El administrador de cuentas (ACM) es responsable por mantener actualizado el estado de cuenta de cada cliente del sistema. A partir de la notificación de órdenes cerradas por el TRM, el ACM realiza la liquidación de las órdenes y las contabiliza en su base de datos. Asimismo, es su responsabilidad proveer información al administrador de reportes cuando este lo requiera.

El administrador de calificaciones (RTM) es la unidad encargada de procesar las calificaciones de los aquellos clientes del sistema (usuarios finales y centros de misión) que consumen los servicios ofrecidos por otros agentes. Las valoraciones de los servicios se centralizan en la base de datos de calificaciones y es responsabilidad del RTM su administración. Además, es responsabilidad del RTM la entrega de la información actualizada sobre las calificaciones, cuando los *brokers* de los agentes del usuario y del MOC lo soliciten.

El administrador de informes (RPM) es la unidad de se encarga de la generación de todos los informes del SSEP: (i) informes periódicos de mercado, (ii) informes de estado de órdenes, (iii) informes de estado de cuenta e (iv) informes periódicos de calificaciones. Para su elaboración hace uso de la información almacenada vinculadas al agente.

5.6 Plataformas Agente

La plataforma agente (AP) es la responsable de brindar la infraestructura necesaria para que los agentes que se alojan en ella puedan efectuar sus operaciones e interactuar entre ellos. Asimismo, toda AP tiene la capacidad de establecer canales de comunicación con otras APs, de forma tal que agentes de una AP local pueden interactuar con agentes de una AP remota. Si bien en esta tesis se han identificado dos tipos de APs (la plataforma agente del AgentSat -FAP- y la plataforma agente del SSEP -GAP), ambas presentan las mismas características y sólo difieren en los tipos y cantidad de agentes que alojan (ver secciones 5.3 y 5.5).

Los tres elementos de arquitectura que forman parte de ambos tipos de APs del SatSES son aquellos recomendados por la especificación FIPA para la gestión de los agentes (FIPA, 2004), a saber:

- i. Agent Management System (AMS)
- ii. Service Directory Facilitator (SDF)
- iii. Message Transport Service (MTS)

Todos ellos son servicios y su terminología ha sido adoptada de la especificación citada. A continuación, se describen las funcionalidades de cada uno de ellos.

5.6.1 Agent Management System

El sistema de gestión de agentes (AMS) es el responsable de la operación de la AP (gestión de recursos, invocación/suspensión de agentes, etc.) y de mantener el directorio de los agentes de la plataforma. Todo agente debe registrarse con el AMS de su plataforma para tener un ID válido. El AMS brinda información acerca de los agentes de la plataforma (ID, dirección, descripción) por lo que sería el equivalente a las páginas blancas o *White Pages* en la jerga de internet.

5.6.2 Service Directory Facilitator

El facilitador del directorio de servicios (SDF) es el responsable de mantener el directorio de servicios ofrecidos por cada uno de los agentes de la AP. De este modo, todo agente puede realizar una búsqueda sobre el SDF para obtener la lista de los agentes que brindan servicios de interés.

En la arquitectura propuesta, cada AgentSat decidirá qué servicios ofrecer en un determinado momento para que otros AgentSats tomen conocimiento. Será el agente sat-Planner quien se ocupará de registrar los servicios ofrecidos en el SDF. La información contenida en el SDF de un determinado AP podrá ser consultado por un agente de otra AP mediante el intercambio de mensajes. Esto significa que en el caso de del segmento de vuelo (donde las FAPs se encuentran en infraestructuras distintas), el AgentSat *i* podrá conocer los servicios que ofrece el AgentSat *j*, y viceversa. Respecto a la plataforma agente en tierra (GAP), todos los agentes involucrados en el mercado de intercambio de servicios comparten el mismo SDF.

En la jerga de internet, el SDF sería el componente que provee las páginas amarillas o *Yellow Pages*.

5.6.3 Message Transport Service

El servicio de transporte de mensajes (MTS) es la unidad que soporta la entrega de mensajes a los agentes que residen en la misma o distintas plataformas. En cada AP, el MTS es provisto de un *Agent Communication Channel* (ACC), el cual ejecuta las tareas de transporte. La transferencia física de mensajes entre distintos ACCs se lleva a cabo mediante el *Message Transport Protocol* (MTP). Para mayores detalles acerca del modelo de referencia FIPA de transporte, referirse al documento *FIPA Agent Message Transport Service Specification* (FIPA, 2002b).

5.7 Estaciones Terrenas

Toda estación terrena que quiera suscribirse al SatSES y ofrecer sus servicios mediante el SSEP deberá realizar una adecuación a la misma instalando una extensión (GSTX) que le permita, además de las actividades nominales, subir y bajar mensajes *AgentSatMessages*, subir los informes de mercado

(*ExchangeMarketData*) y transferir los datos ciencia relacionados con las órdenes de intercambio acordadas con otros agentes.

La arquitecta de la Estación Terrena compatible con el SSEP se presenta en la figura 5.20. En ella se incluye la extensión para el comercio de servicios, que está compuesta por los siguientes 3 elementos:

- i. Trade Service Manager (TSM),
- ii. Extension Data Handler (XDH),
- iii. Extension Message Handler (XMH),

El administrador del servicio de intercambio (TSM) es el elemento de arquitectura a partir del cual se administran las órdenes de servicio de la estación y el perfil del agente dentro del SSEP. Asimismo, a través del TSM se reciben las actualizaciones de estado de las órdenes activas, todo tipo de eventos relacionados y los informes periódicos que envía el SSEP para ser consumidos por el subsistema de planificación de la estación (*Ground Station Planning*). Por el lado de la estación, el TSM es la unidad que recibe el detalle de los servicios de antena disponibles para ser ofrecidos en el SSEP y los informes de pasadas, a partir de los cuales se actualiza el estado de las órdenes en curso.

El manejador de datos de la extensión (XDH) es la unidad encargada de recibir los informes de mercado para los AgentSats y los TLEs necesarios para su seguimiento desde tierra. Es responsabilidad del XDH, además, la transferencia de los datos de ciencia (vinculados a órdenes de intercambio) desde la estación hacia la infraestructura del SSEP.

El manejador de mensajes de la extensión (XMH) es la unidad que actúa de intermediario de los mensajes AgentSats, entre la estación terrena y el SSEP. Por un lado, los *AgentSatMessages* que han sido descargados por la estación, son pre-procesados por la *Data Acquisition Unit* y, luego de detectar que se trata de un mensaje del AgentSat, lo deriva al XMH, que a su vez lo transfiere al GAP del SSEP. Por otro lado, todo aquel mensaje recibido por el XMH desde la plataforma de intercambio, es transferido al sistema de antena para su subida, aprobación mediante del equipo de operaciones.

Por lo demás, se han incluido en la figura ciertos bloques (en color gris) e interfaces (con línea punteada) que, aunque no forman parte del núcleo de la explicación, están presentes en la arquitectura estándar de cualquier estación terrena y/o MOCs, y su introducción ayuda a la comprensión de la misma.

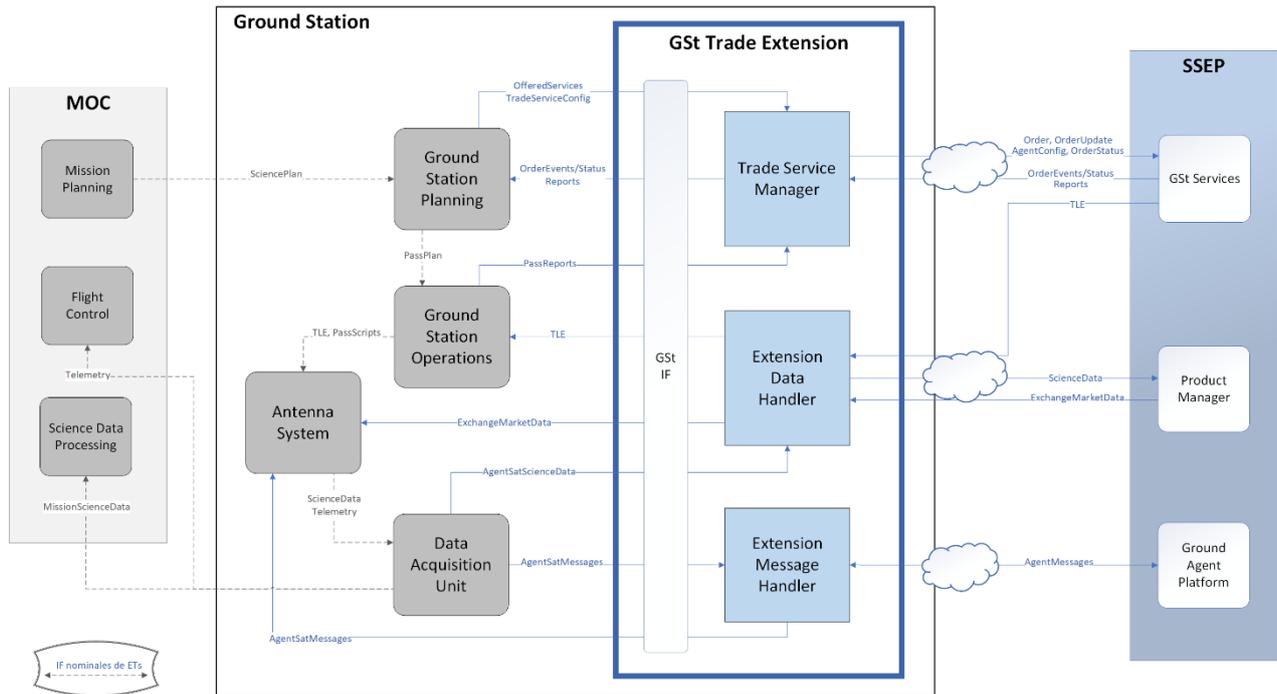


Figura 5.20 – Arquitectura de la Estación Terrena con la extensión para el Comercio de Servicios

5.8 Caracterización de los Agentes del Sistema

En esta sección se caracterizan todos los agentes que componen el sistema, tanto de vuelo como de tierra. Siguiendo los fundamentos teóricos introducidos en la sección 2.3, para cada agente se especifica su tipo, su objetivo, sus percepciones y acciones (tablas 5.5 a 5.11) y, por último, se describe el entorno tal cual es percibido por cada uno de ellos (tabla 5.12).

	AgentSat	
	Tipo de Agente	Reactivo, Basado en Objetivos, Basado en Utilidad.
	Objetivo	Maximizar la utilidad de la misión a partir de la comercialización de los recursos ociosos del satélite.
	Percepciones	Órdenes generadas por MOC ▪ Órdenes de intercambio de otros AgentSats ▪ Aceptación y/o Rechazo a las órdenes enviadas ▪ Estado del servicio de intercambio de otros AgentSats ▪ Telemetría y TT-TC ▪ Datos de Posición del satélite ▪ Eventos del Satélite ▪ Parámetros de configuración del AgentSat ▪ Informes periódicos de mercado.
Acciones	Enviar órdenes de intercambio a otro AgentSats ▪ Aceptar y/o rechazar las órdenes recibidas ▪ Enviar propuesta de plan al MOC ▪ Informar el estado del AgentSat, de las órdenes y de los planes ▪ Transferir datos de Ciencia ▪ Consultar telemetría a CDH ▪ Enviar solicitud de uso de Instrumento y del servicio de Comunicaciones.	

Tabla 5.5 – Caracterización del agente AgentSat

<p style="text-align: center;">sat- Planner</p> 	Tipo de Agente	Reactivo, Basado en Objetivos, Basado en Utilidad.
	Objetivo	Elaborar planes de forma autónoma para maximizar el uso de los recursos del satélite.
	Percepciones	Telemetría y TT-TC ▪ Modo del servicio de intercambio ▪ Servicios habilitados para el intercambio ▪ Nivel de Autonomía ▪ Órdenes de intercambio activas ▪ Eventos del AgentSat ▪ Algoritmos de Estimación ▪ Información de tendencias de mercado
	Acciones	Elaborar plan para ejecutar de forma autónoma a bordo ▪ Enviar propuesta de plan al MOC ▪ Enviar valores reales de las variables estimadas ▪ Notificar sobre los eventos del proceso de planificación.

Tabla 5.6 – Caracterización del agente sat-Planner

<p style="text-align: center;">sat- Trader</p> 	Tipo de Agente	Reactivo, Basado en Objetivos.
	Objetivo	Cerrar acuerdos comerciales con otros AgentSats en órbita.
	Percepciones	Órdenes generadas por el planificador a bordo ▪ Órdenes de intercambio de otros AgentSats ▪ Aceptación y/o Rechazo a las órdenes enviadas ▪ Estado del servicio de intercambio de otros AgentSats ▪ Estado de comunicación ISL ▪ Eventos del AgentSat ▪ Informes de mercado.
	Acciones	Enviar órdenes de intercambio a otro AgentSats ▪ Aceptar y/o rechazar las órdenes recibidas ▪ Informar el estado de intercambio del AgentSat ▪ Notificar sobre las órdenes activas del AgentSat ▪ Notificar sobre el estado de las órdenes ▪ Informar sobre los eventos del proceso comercial ▪ Enviar datos de Mercado

Tabla 5.7 – Caracterización del agente sat-Trader

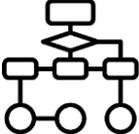
<p style="text-align: center;">sat- Controller</p> 	Tipo de Agente	Reactivo, Basado en Objetivos.
	Objetivo	Ejecutar los planes generados en órbita o en tierra por el MOC.
	Percepciones	Eventos de la Plataforma ▪ Eventos de los Instrumentos ▪ Órdenes generadas por MOC ▪ Planes generados por el planificador a bordo ▪ Estado de las órdenes de intercambio ▪ Eventos del AgentSat ▪ Datos de Posición del satélite.
	Acciones	Notificar sobre el estado del AgentSat, de las órdenes y los planes ▪ Enviar solicitud de uso de Instrumento y del servicio de Comunicaciones ▪ Informar sobre los eventos del AgentSat ▪ Enviar las órdenes a comerciar.

Tabla 5.8 – Caracterización del agente sat-Controller

sat- Learner 	Tipo de Agente	Reactivo, Basado en Objetivos.
	Objetivo	Elaborar estimadores de estados futuros más precisos.
	Percepciones	Valores reales de las variables estimadas ▪ Telemetría del subsistema de potencia ▪ Datos de mercado.
	Acciones	Generar algoritmos de estimación ▪ Informar sobre las tendencias de mercado.

Tabla 5.9 – Caracterización del agente sat-Learner

SSEP- Traders (usr, msc, gst, pvd) 	Tipo de Agente	Reactivo, Basado en Objetivos.
	Objetivo	Cerrar acuerdos comerciales con otros agentes del SSEP.
	Percepciones	Alta y modificación de órdenes de intercambio ▪ Actualización de estado de las órdenes ▪ Órdenes de intercambio de otros AgentSats ▪ Aceptación y/o Rechazo a las órdenes enviadas ▪ Parámetros de configuración del agente ▪ Mensajes de <i>AgentSats</i> ▪ Datos de calificación de los agentes del sistema ▪ Informes periódicos del SSEP.
	Acciones	Enviar órdenes de intercambio a otros agentes ▪ Aceptar y/o rechazar las órdenes recibidas ▪ Enviar el contenido de los mensajes al cliente ▪ Notificar sobre los eventos y estados de las órdenes ▪ Enviar mensajes a <i>AgentSats</i> ▪ Enviar los informes periódicos del SSEP al cliente.

Tabla 5.10 – Caracterización de los agentes Traders del SSEP (usr, msc, gst y pvd)

Trade Manager 	Tipo de Agente	Reactivo, Basado en Objetivos.
	Objetivo	Supervisar y contabilizar las operaciones comerciales efectuadas en la plataforma, y proveer servicios e información a los agentes del sistema.
	Percepciones	Mensajes <i>AgentSat</i> provenientes de estaciones terrenas ▪ Mensajes <i>AgentSat</i> provenientes de los agentes del SSEP ▪ Eventos de órdenes de intercambio ▪ Calificaciones del servicio prestado.
	Acciones	Enviar mensajes <i>AgentSat</i> dirigidos a estaciones terrenas ▪ Enviar mensajes <i>AgentSat</i> dirigidos a los agentes del SSEP ▪ Generar los informes periódicos del SSEP ▪ Informar acerca de los datos de calificación de los agentes del sistema.

Tabla 5.11 – Caracterización del agente Trade Manager del SSEP

5.8.1 Caracterización del Entorno para cada agente

Dado que existe una variedad de agentes interactuando en el sistema, el entorno es *multiagente*. Asimismo, para todos los agentes, el entorno es considerado *observable* (los sensores de los agentes proveen un estado completo del entorno) y *discreto* (el entorno tiene un número finito de estados).

Respecto a la previsibilidad del entorno, el mismo es *determinístico* para el sat-Planner, sat-Controller, sat-Learner y Trade Manager, y es *estratégico* para el AgentSat, sat-Trader y los Traders del SSEP. Recordar que un entorno es estratégico cuando es determinista salvo por las acciones de otros agentes. Y esto sucede para los últimos agentes mencionados, ya que, por ejemplo, si alguno de ellos envía una orden de intercambio, no puede a priori conocer el próximo estado ya que la aceptación o rechazo de la misma, depende del agente destinatario y de otras órdenes que este último pudiera haber recibido de otros agentes.

En lo que concierne a la independencia de las percepciones, el entorno es considerado *episódico* para el sat-Trader y los Traders del SSEP, y *secuencial* para el resto de los agentes. Esto es, para los agentes comerciales no interesa lo que haya sucedido antes, ya que su objetivo es comunicar las órdenes que se encuentran activas e intentar cerrar acuerdos con otros agentes, independiente de los acuerdos previos. Aunque el resultado de los acuerdos previos pudo haber modificado la calificación de los agentes involucrados (y, por ende, tener implicancias en las decisiones actuales), esa información se resume en la información de calificación que el bróker consulta en cada ciclo de ejecución.

Por último, en relación a la variabilidad del entorno, el mismo es considerado *estático* por los agentes del SSEP y *dinámico* por los agentes de vuelo. Esto tiene que con que el *AgentSat Payload* es parte de un satélite y, en consecuencia, cualquier contingencia que afecte a uno, afecta al otro. Por ende, cualquier evento de alta prioridad (nivel crítico de potencia, desperfecto de un subsistema importante, altera de colisión con, etc.) que requiera atención inmediata, hará que las actividades del *AgentSat Payload* sean abortadas o, al menos, pausadas.

Agente	Seg- mento	Observable	Parc. Observable	Determinístico	Estocástico	Estratégico	Episódico	Secuencial	Estático	Dinámico	Discreto	Continuo	Agente	Multiagente
		vs.		vs.	vs.	vs.	vs.	vs.	vs.	vs.	vs.	vs.	vs.	
AgentSat	Vuelo	X				X		X		X	X			X
sat-Planner		X		X				X		X	X			X
sat-Trader		X				X	X			X	X			X
sat-Controller		X		X				X		X	X			X
sat-Learner		X		X				X		X	X			X
user-Trader msc-Trader gst-Trader pvd-Trader	Tierra	X				X	X		X		X			X
Trade Manager		X		X				X	X		X			X

Tabla 5.12 – Caracterización del entorno de acuerdo a cada uno de los agentes del sistema

5.9 Comunicación entre los Agentes del Sistema

Como ya se introdujo previamente, el sistema multiagente que se ha diseñado está basado en el modelo de referencia FIPA y su lenguaje de comunicación FIPA-ACL (inciso 2.3.6).

En esta sección se presenta el protocolo de interacción IOSSE que ha sido definido específicamente para la comunicación entre dos AgentSats en órbita (inciso 5.9.1). Para la comunicación de los agentes en tierra (dentro del SSEP) no ha sido necesario definir ningún protocolo ad-hoc, ya que alcanza con aquellos incluidos en el estándar FIPA. Dentro de ellos, se prevé que los protocolos de interacción que más se utilizarán será el *FIPA-Request* y *FIPA-Propose* (inciso 5.9.2) y el protocolo *ContractNet* (inciso 5.9.3).

5.9.1 Protocolo de Interacción entre dos AgentSats en órbita

La finalidad del protocolo de interacción para el intercambio de servicios satelitales (IOSSE, por sus siglas en inglés) es que dos AgentSats puedan ofrecer sus recursos ociosos en órbita luego de haber establecido un enlace intersatelital y, en caso positivo, establezcan acuerdos de servicios entre ellos. La ejecución del protocolo es responsabilidad del *broker* de cada AgentSat, quien inicia su ejecución cuando es notificado por su ISL Payload, quien implementa la comunicación.

Cada uno de los mensajes incluidos dentro del protocolo IOSSE-IP se corresponden con alguno de los tipos de actos comunicativos (CA, por sus siglas en inglés) incluidos en la librería estándar de FIPA. En este caso, los CA utilizados son:

- *inform*: es el mecanismo básico para comunicar información.
- *cancel*: para informar que un agente no desea seguir ejecutando una determinada acción.
- *propose*: para permitir que un agente le haga una propuesta a otro agente.
- *accept-proposal*: para que un agente acepte la propuesta hecha por otro agente.
- *reject-proposal*: para que un agente rechace la propuesta hecha por otro agente.

El protocolo de interacción se presenta en la figura 5.21 y este inicia luego de que se haya establecido el enlace intersatelital entre los dos AgentSats. Sus pasos se resumen a continuación:

1) Intercambio del Estado de intercambio

Ambos satélites intercambian su estado actual de intercambio (*TradeStatus*) mediante un mensaje de tipo *inform*. En caso que exista un *match* de estados (ver tabla 5.13) se avanza en la comunicación, sino se cancela.

Un *match* de estado significa que potencialmente podría establecerse un acuerdo de servicio, y esto dependerá de las características de los servicios ofrecidos y requeridos. La tabla 5.13 lista todas las posibles combinaciones de estados individuales que genera un *match*; para cualquier otra opción, la

comunicación se cancela ya que no será posible establecer ningún acuerdo (ej.: los dos satélites desean vender servicios y ninguno comprar).

AgentSat <i>i</i>			Match de Estado	AgentSat <i>j</i>		
Compra Servicios	Vende Servicios	Trade Status		Trade Status	Compra Servicios	Vende Servicios
SI	SI	ON-SS	SI	ON-SS	SI	SI
SI	SI	ON-SS	SI	ON-SN	SI	NO
SI	SI	ON-SS	SI	ON-NS	NO	SI
SI	NO	ON-SN	SI	ON-SS	SI	SI
SI	NO	ON-SN	SI	ON-NS	NO	SI
NO	SI	ON-NS	SI	ON-SS	SI	SI
NO	SI	ON-NS	SI	ON-SN	SI	NO

Tabla 5.13 – Conjunto de combinaciones de estados individuales que generan ‘Match’ potencial.

2) El AgentSat *i* ofrece sus servicios disponibles

De acuerdo a los servicios ociosos computados por el planificador a bordo, el AgentSat *i* envía un mensaje del tipo *propose* para ofrecer sus servicios disponibles.

3) El AgentSat *j* acepta o rechaza la propuesta de servicios.

En el caso de aceptación de servicio, toda información que sea necesaria para que el agente *i* ejecute el servicio, es incluida por el agente *j* en su mensaje *accept-proposal*. Por ejemplo, en caso que hubiese sido aceptado un servicio de mensajería, como parte del *accept-proposal* se envía el destinatario del mensaje y el mensaje.

En aquellos casos en los que se rechaza la oferta, puede incluir el motivo del rechazo como parte del del mensaje *reject-proposal*, pero el mismo es opcional.

4) El AgentSat *j* ofrece sus servicios disponibles

Ídem paso 2.

5) El AgentSat *i* acepta o rechaza la propuesta de servicios.

Ídem paso 3.

Luego del paso 5, la comunicación se da por finalizada. En caso de haber establecido acuerdos de servicios satelitales:

- el satélite proveedor del servicio pasará el flujo de ejecución al sat-Controller, quien iniciará la ejecución de orden de servicio.

- el satélite consumidor del servicio, quedará en estado WFA y esperará ser informado del estado de la orden de servicio desde tierra, ya sea mediante los informes periódicos enviados desde el SSEP o por su agente MOC.

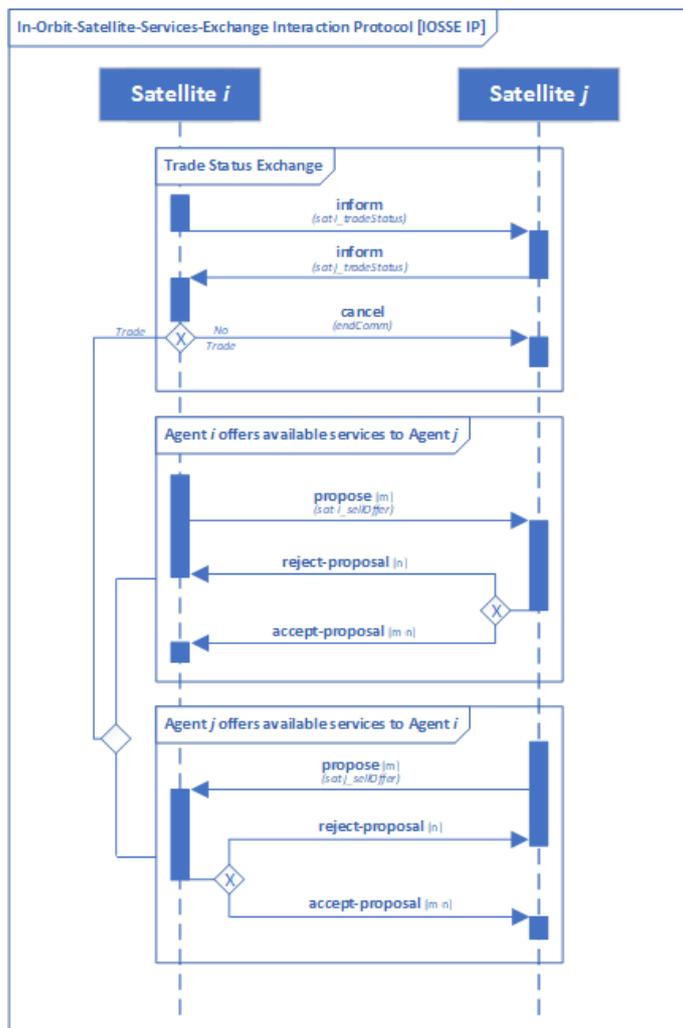


Figura 5.21 – Protocolo de Interacción IOSSE: Intercambio de servicios satelitales en órbita

A continuación, se incluyen ejemplos de mensajes FIPA-ACL por cada tipo de CA incluido en el IP-IOSSE.

La figura 5.22 muestra un mensaje de tipo *inform* enviado por el AgentSat *i* y el mismo mensaje respondido por el AgentSat *j*. Se observa en el primer mensaje (izquierda) que el satélite *i* informa que su estado de intercambio es [*on*, *off*], lo que significa que la venta de servicios está activada (primer elemento del vector, *on*) y la compra de servicios está deshabilitada (segundo elemento del vector, *off*). La respuesta del satélite *j* se observa en el mensaje de la derecha y su estado de intercambio es exactamente el contrario, [*off*, *on*]: habilitado para la compra y deshabilitado para la venta. Por lo tanto, aquí se da un *match* de estados y por ende la comunicación seguirá adelante.

La figura 5.23 presenta mensajes de tipo *propose* para la oferta de servicios y la figura 5.24 muestra los mensajes del tipo *accept-proposal* y *reject-proposal*.

Agent i → j	Agent j → i
<pre>(inform :sender (agent-identifier :name sat00033) :receiver (agent-identifier :name sat00730) :language Prolog :ontology in-orbit-trade :protocol iosse :reply-with trade-status-00001111 :content "trade_status (sat00033, [on, off])")</pre>	<pre>(inform :sender (agent-identifier :name sat00730) :receiver (agent-identifier :name sat00033) :language Prolog :ontology in-orbit-trade :protocol iosse :in-reply-to trade-status-00001111 :content "trade_status (sat00730, [off, on])")</pre>

Figura 5.22 – Ejemplo de mensajes FIPA-ACL para el intercambio de estados de intercambio

Agent i → j	
<pre>(propose :sender (agent-identifier :name sat00033) :receiver (agent-identifier :name sat00730) :language fipa-sl :ontology messaging-trade :protocol iosse :reply-with bid00000736 :content "((action (agent-identifier :name sat00033) (sell-messaging :price-message ... :svc-avail-from ... :svc-avail-to ... :est-time2ground ... :max-msg-size ... :...)))")</pre>	Oferta de Servicio de mensajería
<pre>(propose :sender (agent-identifier :name sat00033) :receiver (agent-identifier :name sat00730) :language fipa-sl :ontology instr-time-trade :protocol iosse :reply-with bid00000737 :content "((action (agent-identifier :name sat00033) (sell-instr-time :price-image ... :svc-avail-from ... :svc-avail-to ... :instr-type ... :instr-bands ... :instr-wavelength ... :instr-geo-res ... :instr-rad-res ... :instr-swath ... :instr-pointing ... :image-raw-size ... :...)))")</pre>	Oferta de Tiempo de Instrumento
<pre>(propose :sender (agent-identifier :name sat00033) :receiver (agent-identifier :name sat00730) :language fipa-sl :ontology downlink-trade :protocol iosse :reply-with bid00000738 :content "((action (agent-identifier :name sat00033) (sell-downlink :price-mbps ... :svc-avail-from ... :svc-avail-to ... :max-dlk-size: ... :contacts [...] :...)))")</pre>	Oferta de Servicio de Bajada

Figura 5.23 – Ejemplos de mensajes FIPA-ACL: AgentSat i ofreciendo servicios satelitales en órbita

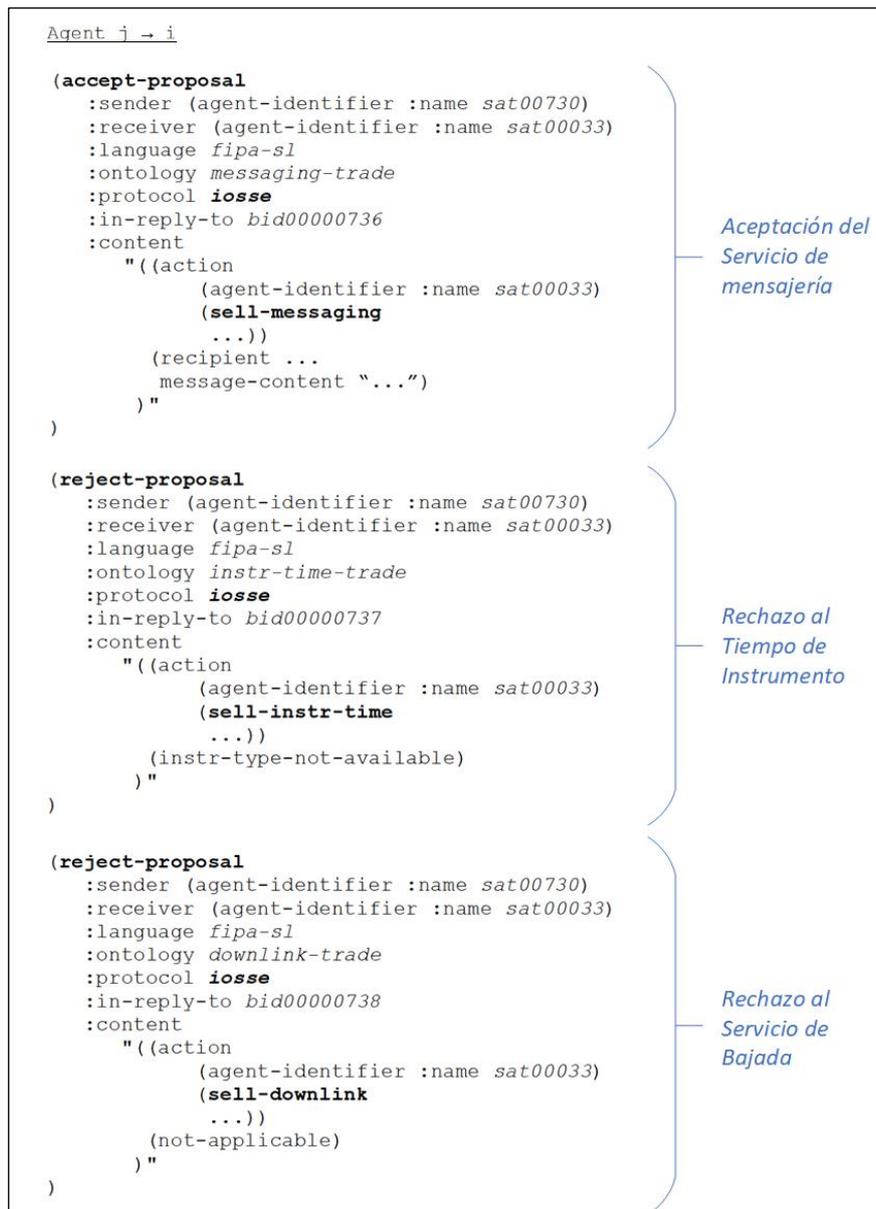


Figura 5.24 – Ejemplos de mensajes FIPA-ACL: AgentSat j aceptando y rechazando servicios ofrecidos

5.9.2 Protocolo de Interacción para Solicitud y Oferta de Servicios (2 agentes)

Un agente en el SSEP utilizará el protocolo de interacción *FIPA-Request* (FIPA, 2002f) cuando desee comprar servicios satelitales y un *FIPA-Propose* (FIPA, 2002e) cuando desee ofrecerlos. Ambos se presentan en la figura 5.25 y han sido adaptados para la aplicación en cuestión.

El agente que inicia el protocolo para la compra de servicios (lado izquierdo de la figura) emplea un CA del tipo *request* para enviar la solicitud (*ServiceRequest*). El agente vendedor (participante) recibe el pedido y decide si lo rechaza mediante un *refuse* o la acepta con un *agree*. En caso que el pedido haya sido aceptado, el agente participante da curso a la orden de servicio y, una vez finalizada, deberá informar al agente

comprador acerca del estado/resultado de la misma: (a) no se pudo ejecutar correctamente y envía un *failure* o (b) la orden se ejecutó correctamente, notificándole con un *inform*. En este último caso puede comunicarse sólo el estado exitoso o también dar mayores detalles acerca de la misma, dependerá del caso.

En el caso de una oferta de servicios, el agente que inicia el protocolo es el agente vendedor. En esta situación, el flujo es idéntico al caso previo con la única diferencia siendo los tipos de CA que son transmitidos. Para enviar la oferta de servicios (*ServiceOffer*) se utiliza un *propose*; para rechazar la oferta se envía un *reject-proposal* y para aceptarla, un *accept-proposal*.

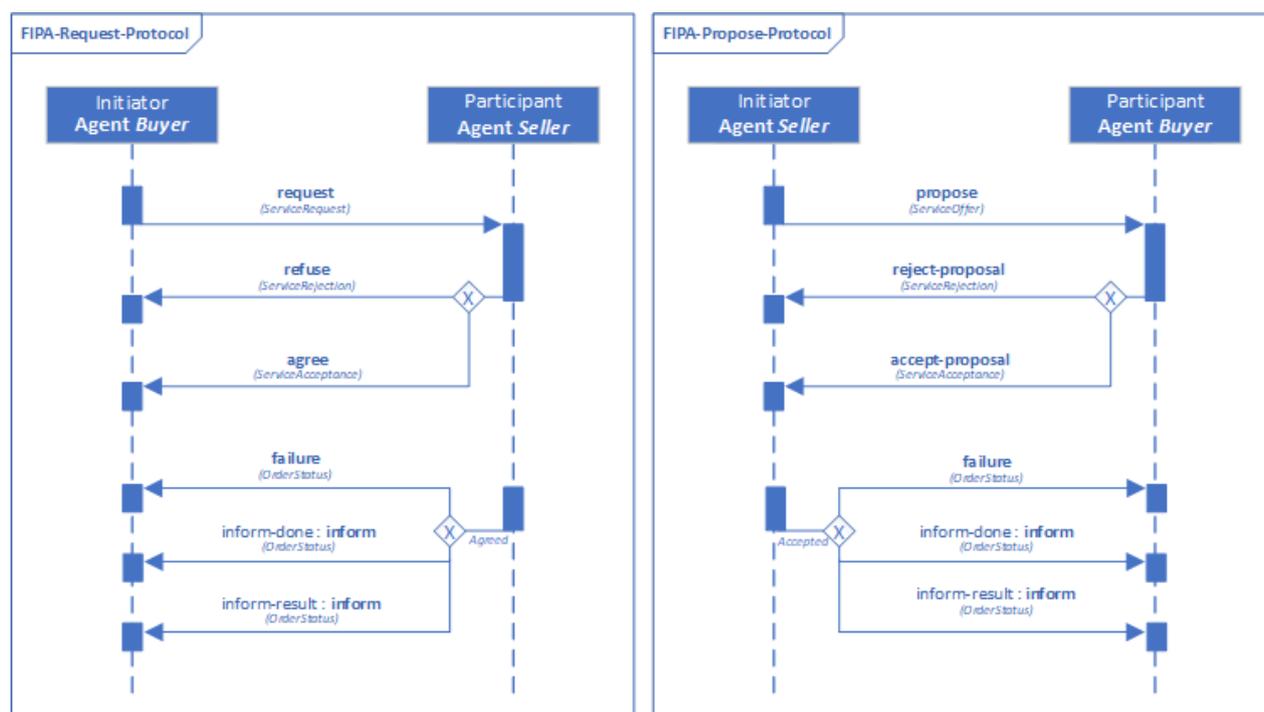


Figura 5.25 – Protocolos FIPA-Request y FIPA-Propose: Solicitud y Oferta de servicio en el SSEP (adaptados de [FIPA, 2002f; 2002e])

5.9.3 Protocolo de Interacción para Subasta inversa de Servicios (n agentes)

Un agente del SSEP utilizará el protocolo de interacción *ContractNet* (FIPA, 2002d) cuando desee comprar servicios satelitales en la modalidad de subasta inversa, esto es, enviando una orden de compra a un conjunto de agentes y esperando que los potenciales vendedores del servicio realicen su mejor oferta. El criterio de adjudicación de la orden será definido por el agente solicitante y dicha información será especificada en la convocatoria (*Call for Proposals -cfp*). Normalmente, el factor determinante será el tiempo de entrega, el precio, la calificación del proveedor, etc., pero bien podrá ser cualquier otra característica. El protocolo se presenta en la figura 5.26 y ha sido adaptado para la aplicación en cuestión.

El agente que inicia el protocolo emplea un CA del tipo *cfp* para enviar la solicitud de compra (*ServiceRequest*), en modo subasta, a un conjunto de potenciales vendedores (*m*). Aquellos agentes que respondan al llamado en tiempo (*n*), podrán rechazarlo (*ServiceRejection*) enviando un *refuse* o, por el contrario, enviar su propuesta (*ServiceOffer*) respondiendo con un *propose*. Una vez recibidas todas las propuestas (*j*), el agente comprador decidirá cuales rechaza y cuales acepta (*l*) mediante un CA del tipo *reject-proposal* o *accept-proposal*. El acuerdo de intercambio de servicio puede involucrar a uno o varios proveedores y esto será decisión del comprador. Una vez efectuado el acuerdo y la adjudicación, los agentes proveedores darán curso a la orden y, una vez finalizada, deberán informar al comprador acerca del estado/resultado de la misma, al igual que se ha descrito en el inciso anterior.

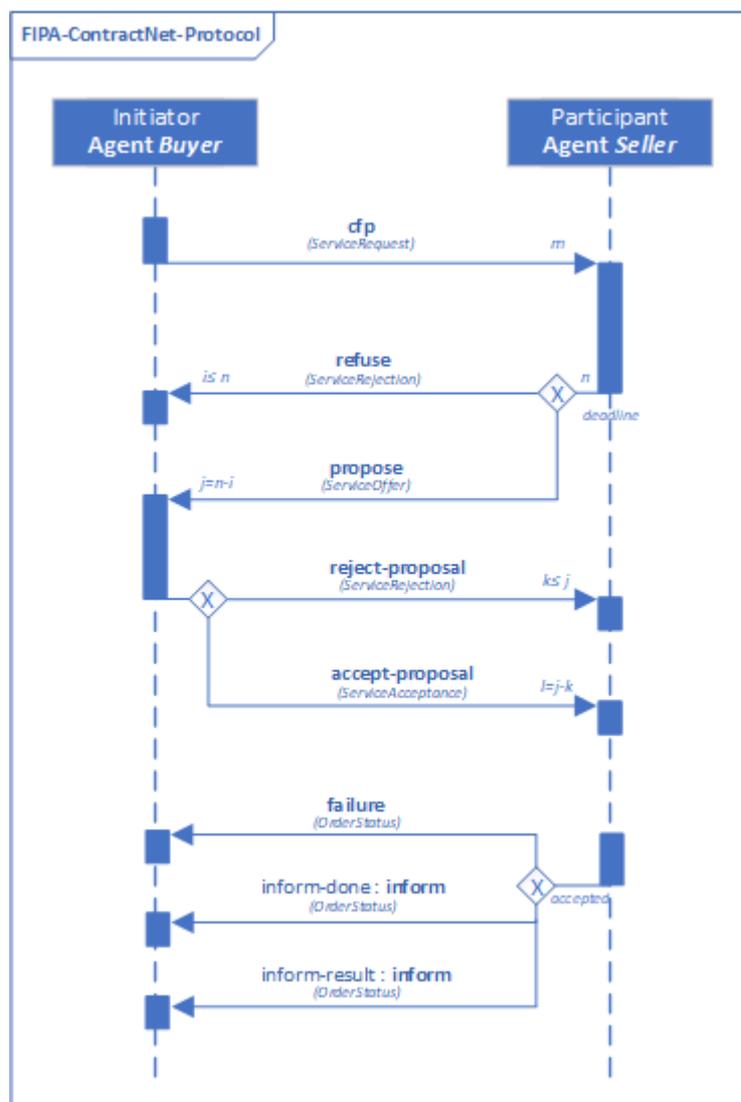


Figura 5.26 – Protocolo FIPA-ContractNet: Llamado a propuestas para Solicitud de servicio en el SSEP (adaptado de [FIPA, 2002d])

Capítulo 6

VALIDACIÓN DE LA ARQUITECTURA

En este capítulo se presentan dos casos de aplicación con el propósito de brindar una validación conceptual de la arquitectura. El primer de ellos trata sobre la gestión de una emergencia que ha sido detectada por un satélite en órbita (sección 6.1) y, el segundo, está relacionado con el uso del sistema para dar soporte a una emergencia detectada en tierra (sección 6.2).

6.1 Caso de Aplicación 1: Gestión de una Emergencia detectada en Órbita

6.1.1 Presentación del caso

En esta sección se describe la dinámica del sistema para el caso de una emergencia detectada en la superficie de la tierra por parte de un satélite suscripto al SatSES. El resumen conceptual del caso se presenta en la figura 6.1.

Los supuestos del caso se detallan a continuación:

- Todos los elementos presentados en la figura 6.1 se encuentran suscriptos al SatSES, a saber:
 - Los 2 satélites: SAT A, SAT B.
 - Los Centros de Operación de Misión (MOC A y B), los cuales poseen Estaciones Terrenas propias (ET MOC A y ET MOC B, respectivamente).
- El satélite A es comandado y monitoreado desde el MOC A y el satélite B lo es desde el MOC B.
- Los satélites A y B cuentan con recursos disponibles para llevar a cabo actividades que no son propias de la misión.
- El satélite A detecta, en vuelo, un evento clasificado como “Emergencia”.

- El satélite A cuenta con instrumentos y procesadores de imágenes que permiten detectar incendios a partir de las imágenes adquiridas.
- Las instituciones que se encuentran suscriptas a la zona geográfica donde se detectó el evento, valoran más la información de la emergencia que la imagen relacionada.

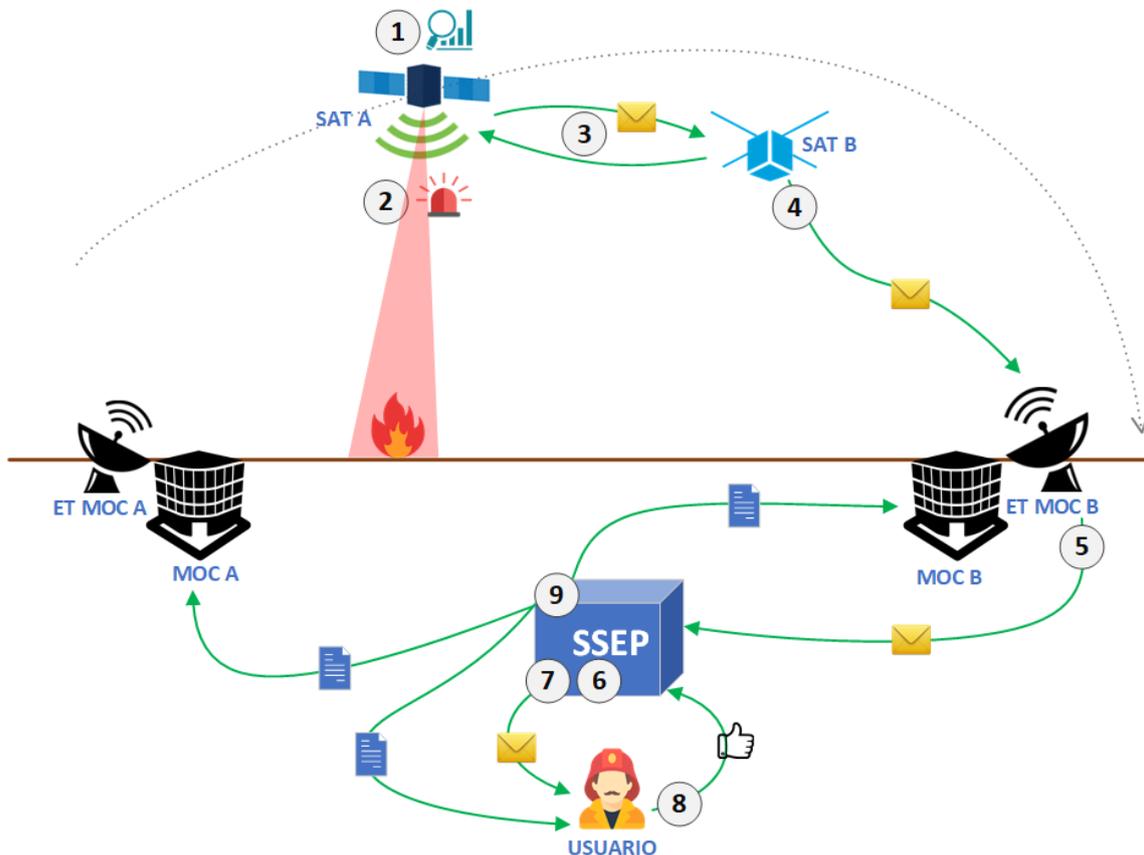


Figura 6.1 – Esquema: Gestión de una Emergencia detectada en órbita

6.1.2 Descripción detallada del caso

Paso 1 → [SAT A] Planificación a bordo de recursos ociosos

Los subpasos se detallan a continuación y se representan en las figuras 6.2, 6.3 y 6.4.

1.1) La unidad “Platform Data & Services Interface” del AgentSat a bordo del Satélite A consulta periódicamente por el estado del satélite y las actividades planificadas por la misión. Esto lo realiza mediante consultas (SatStatusQuery) al subsistema CDH, quien responde enviando la telemetría actual y los telecomandos Time-Tagged activos.

1.2) A partir de los datos recibidos, el Sat-Planner, a través del “Current State Calculator”, procesa los datos recibidos de la plataforma y obtiene el estado energético actual del satélite y el estado actual de cada uno de los servicios habilitados para el AgentSat (por el MOC, a través del TC EnabledServices). En

este caso en particular, se encuentra habilitado el servicio de comunicaciones, el instrumento de adquisición y el instrumento de procesamiento de imágenes. Toda esta información se condensa en el producto llamado “SatCurrentState”.

- 1.3) Partiendo del estado actual del satélite, la unidad “Future State Estimator” computa la capacidad ociosa del satélite para el próximo ciclo. Para ello, tiene en cuenta el consumo de energía que demandarán las actividades planificadas (TT-TC) y el consumo de energía requerido para las tareas operativas y rutinarias del satélite. El cálculo de la capacidad ociosa también considera la existencia de órdenes que aún se encuentren activas dentro del universo del AgentSat (ActiveOrders), ya que éstas consumen servicios y potencia eléctrica. Se supone para este caso particular, que no existe ninguna orden del AgentSat activa. La salida de este proceso es el producto llamado “SatFutureState”.
- 1.4) La unidad “Mission Utility Maximizer” (MUM) recibe la información de que el satélite cuenta con capacidad ociosa en los tres servicios habilitados: comunicaciones, instrumento y procesamiento de imágenes. Por lo tanto, intentará maximizar el uso de los recursos ofreciendo servicios a otras misiones, de forma tal de incrementar la utilidad de la misión. En este escenario particular, la opción de comprar servicios ha sido deshabilitada desde tierra. Luego de evaluar las tendencias del mercado relativas a la oferta y demanda de servicios espaciales (a partir de la información provista por el sat-Learner), el MUM arma un plan, computa su utilidad y el mismo satisface el criterio de aceptación. El plan elaborado se muestra en la tabla 6.1.

Paso	Acción	Precondición	Parámetros Acción
1	<i>Adquisición</i>	<i>Ninguna</i>	<i>Instrumento = A</i> <i>Coordenadas = X,Y,Z</i>
2	<i>Procesamiento</i>	<i>Imagen disponible en Mass Memory</i>	<i>Librería = Detección Incendios</i>
3.1	<i>Contratar Servicio Mensajería en Órbita</i>	1) <i>Patrón de Incendio Detectado</i> 2) <i>Satélite con servicio disponible</i> 3) $TaPlat[sat] < TaPlat[ETA]$	<i>Destinatario = Satélite en órbita</i> <i>Com = ISL Payload</i>
3.2	<i>Bajada de Información a ET</i>	1) <i>Patrón de Incendio Detectado</i> 2) $TaETA \leq umbralDef$ 3) <i>Sin satélite con servicio disponible</i>	<i>Destinatario = ET MOC A</i> <i>Com = Transponder Satélite</i>
3.3	<i>Liberar Recursos</i>	<i>Patrón de Incendio No Detectado y Fin de Ciclo</i>	-
Tipo de Plan: EXE (Ejecución sin necesidad de aprobación de MOC)			

Tabla 6.1 – Esquema del Plan elaborado para el caso de aplicación

La zona geográfica elegida surge de las altas valoraciones que otorgan determinados usuarios suscriptos al SSEP, quienes requieren información satelital sobre zonas agrestes en las que son frecuentes los incendios. Estos usuarios han generado una orden de compra de información satelital (no imágenes crudas) relativa a incendios en el SSEP por un periodo de 2 meses (tiempo que dura la época de incendios en la zona) con el objetivo de prevenir y mitigar este tipo de emergencias ambientales. Puntualmente, las regiones indicadas se corresponden con determinadas localidades de la Provincia de Córdoba dentro de la República Argentina y los usuarios finales son los organismos civiles y estatales responsables de gestionar este tipo de emergencias.

- 1.5) Considerando que el AgentSat ha sido configurado con un nivel de autonomía alto (Ejecutor), el MUM envía el plan al sat-Controller. Por último, el planificador ejecuta la última acción de cada ciclo, que es actualizar el estado del servicio. El valor final del modo del servicio de intercambio queda configurado en: ON-S (Habilitado sólo para la venta).

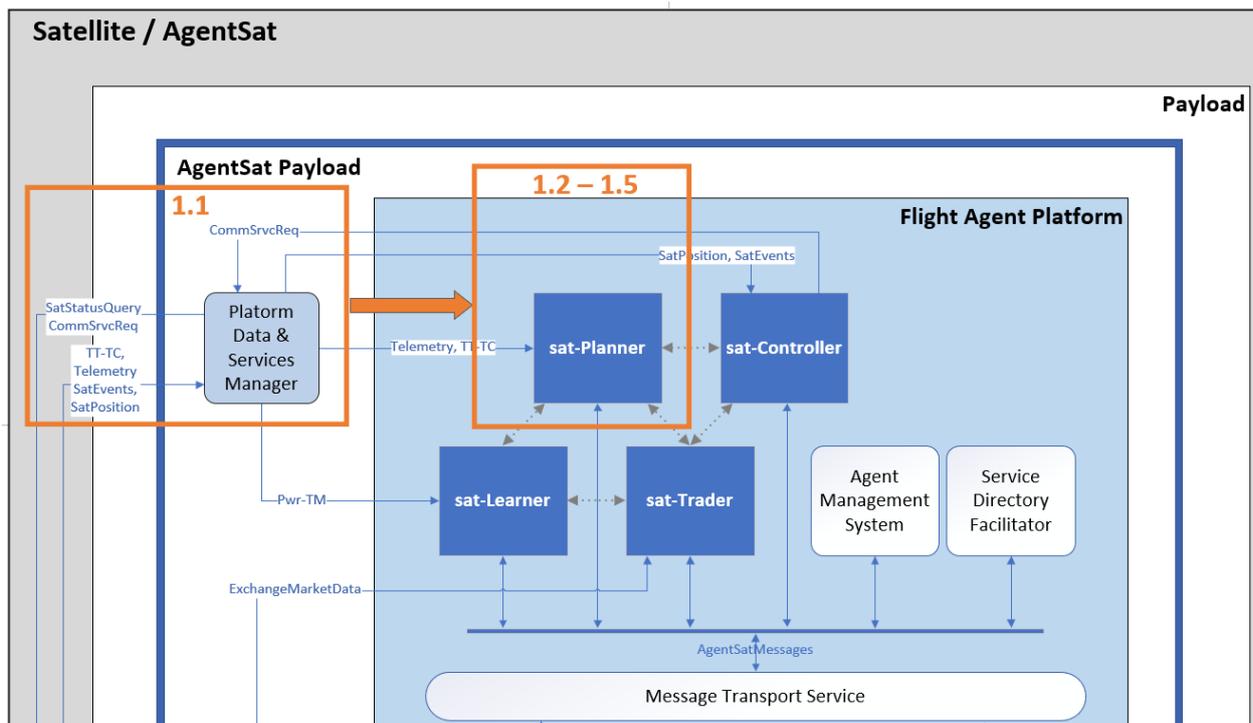


Figura 6.2 – Extracto de la Arquitectura del AgentSat resaltando las cajas activadas durante el Paso 1

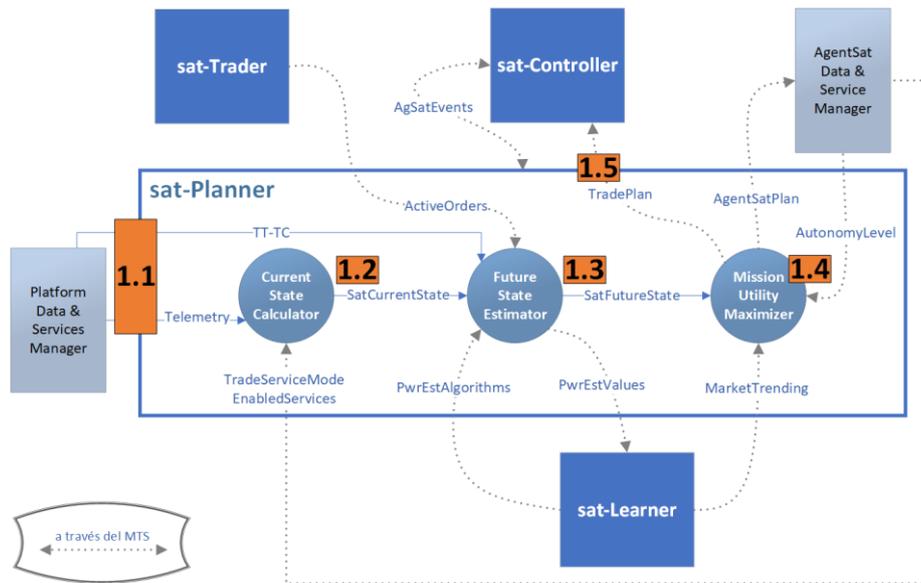


Figura 6.3 – Asociación entre los subpasos del Paso 1 y las unidades internas del sat-Planner

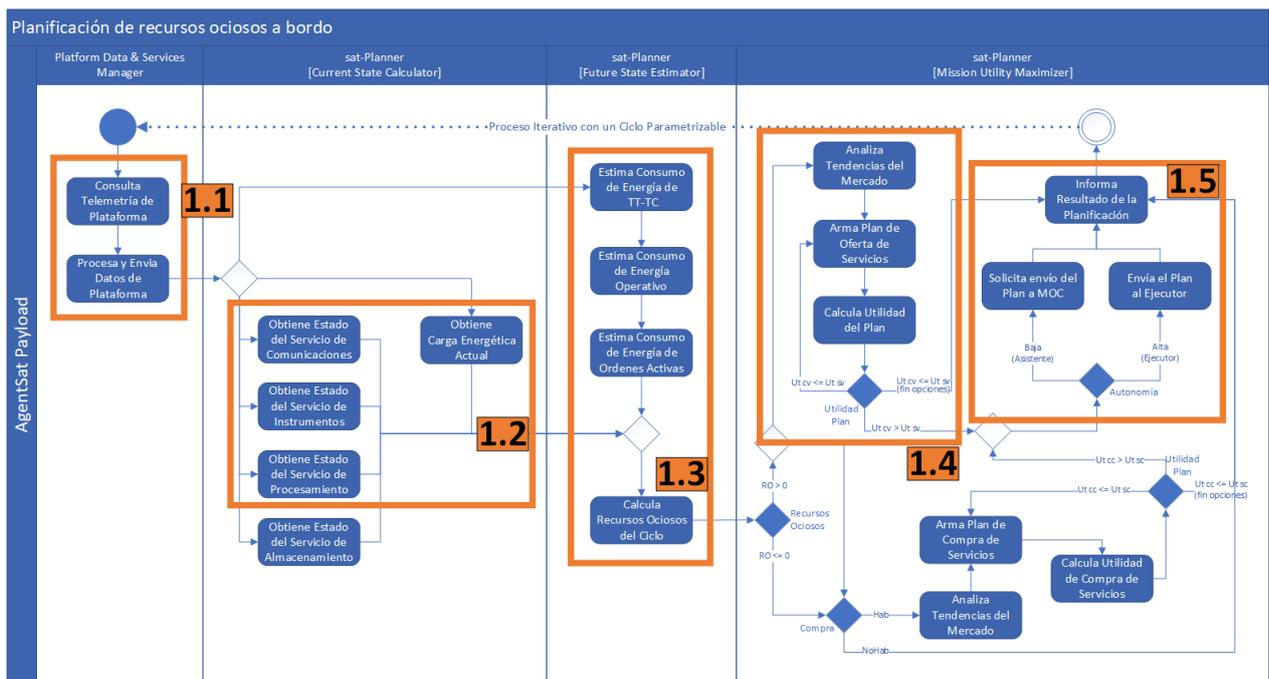


Figura 6.4 – Diagrama de Actividad: Planificación a bordo de Recursos Ociosos

Paso 2 → [SAT A] Ejecución del Plan parte 1 de 2: Adquisición, Procesamiento y Detección

Los subpasos se detallan a continuación y se representan en las figuras 6.5.

2.1) Con el nuevo plan armado, el Ejecutor del plan (sat-Controller) recibe el mensaje del sat-Planner y toma el control. El ejecutor interpreta la información del plan y la prepara para iniciar su ejecución.

- 2.2) La primera acción del plan es adquirir sobre la zona correspondiente. Para ello, el sat-Controller envía la orden al Instrumento correspondiente (a través del AgentSat Data & Service Manager) una vez que el satélite está próximo a la zona de adquisición.
- 2.3) Una vez realizada la adquisición se envía la orden al instrumento de procesamiento de imágenes a bordo para que procese la imagen utilizando las librerías específicas para detección de focos de incendio. Finalizado el procesamiento, se detectaron algunos puntos en la imagen que fueron caracterizados como focos de incendios, por lo cual el próximo paso es enviar la información al SSEP.
- 2.4) A partir de la información de posición actual del satélite y del cómputo del tiempo hasta la estación terrena propia (ET A), el ejecutor del plan concluye que es conveniente buscar servicios de mensajería en órbita para que la información llegue más rápidamente a tierra. Por lo tanto, emite una orden de intercambio (TradeOrder) dirigida al sat-Trader, quien será el responsable de gestionarla.

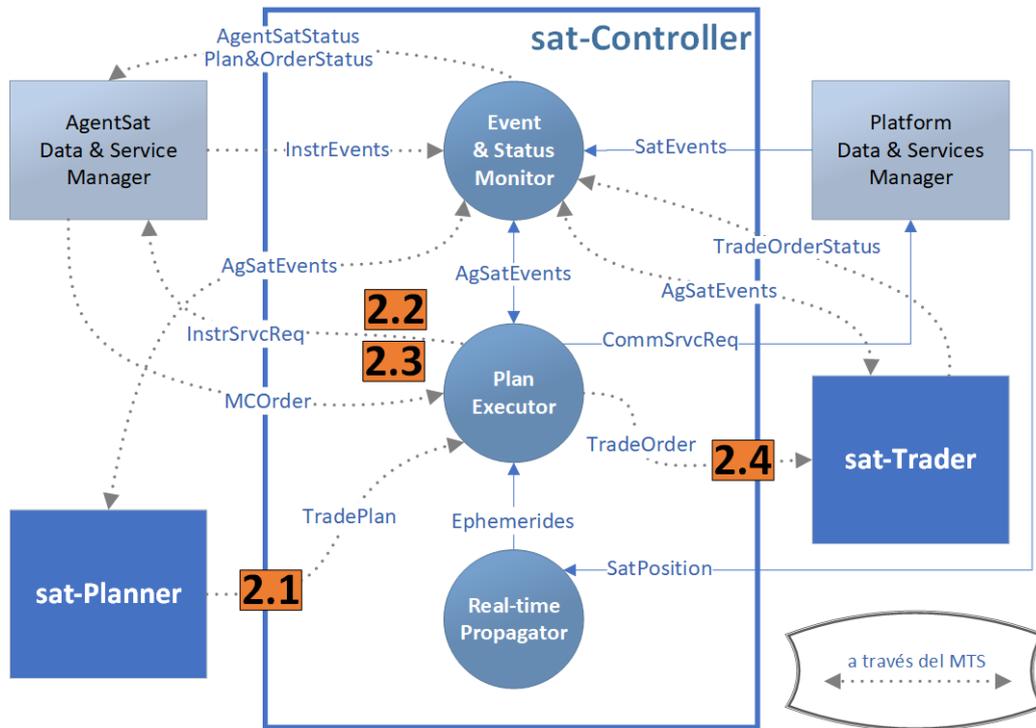


Figura 6.5 – Asociación entre los subpasos del Paso 2 y las unidades internas del sat-Controller

Paso 3 → [SAT A] *Ejecución del Plan parte 2 de 2: Contratación de Servicio de Mensajería en Órbita*

Los subpasos se detallan a continuación y se representan en las figuras 6.6, 6.7 y 6.8.

- 3.1) El gestor de órdenes del sat-Trader recibe la orden de intercambio (TradeOrder), la procesa y la da de alta en su base de datos con estado “Activa”. A continuación, notifica acerca de la nueva orden al Broker.
- 3.2) El Broker crea todos los mensajes que serán intercambiados durante el protocolo de intercambio de servicios en órbita, a partir de la nueva orden recibida. Luego, su flujo de ejecución queda en pausa hasta que se establezca el enlace con algún satélite de órbita cercana.
- 3.3) El satélite B, que orbita en las cercanías del satélite A, está actualmente sólo ofreciendo servicios de mensajería a tierra. Se establece la comunicación entre ambos satélites, se ejecuta el Protocolo IOSSE (figura 6.9) y se cierra el acuerdo de servicio. Toda la información relativa a la emergencia detectada es transmitida por el satélite A en el cuerpo del mensaje durante la ejecución del protocolo de Interacción.
- 3.4) Finalmente, el gestor de órdenes actualiza el estado de la orden en la base de datos (“Active-TRD”) y al mismo tiempo, notifica el sat-Controller.

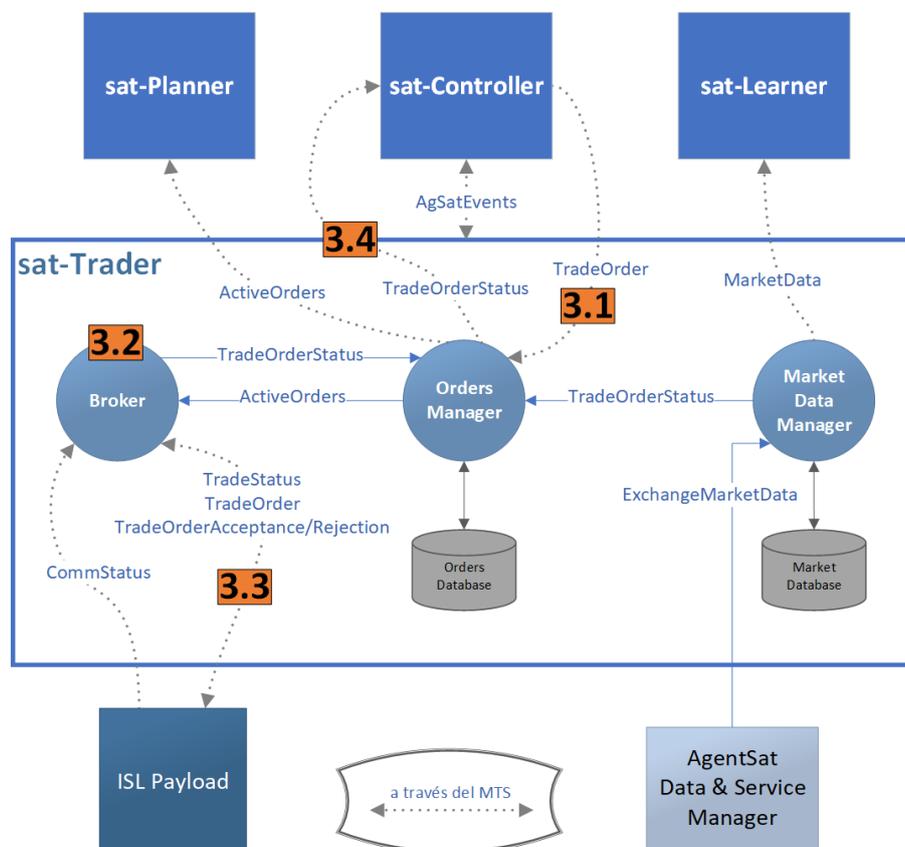


Figura 6.6 – Asociación entre los subpasos del Paso 3 y las unidades internas del sat-Trader

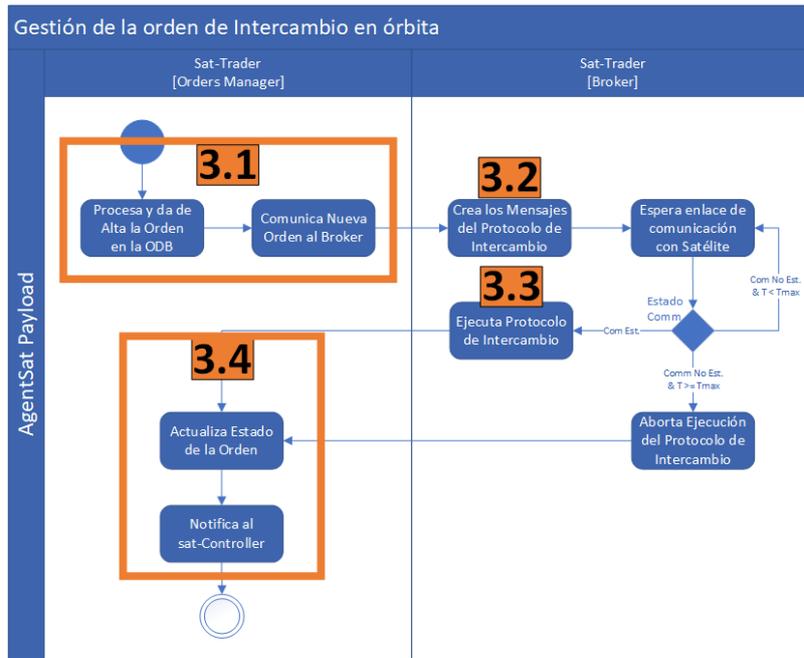


Figura 6.7 – Diagrama de Actividad: Gestión de Intercambio de Servicios en Órbita.

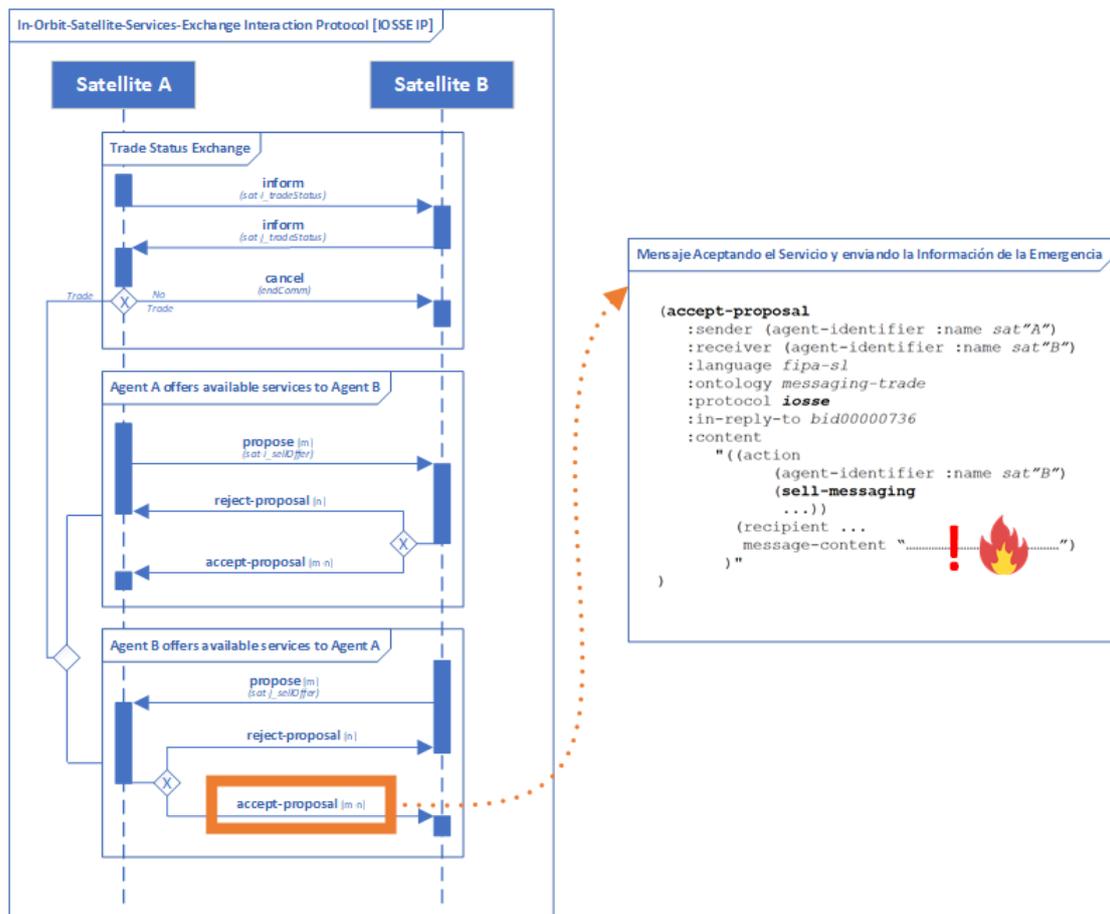


Figura 6.8 – Protocolo de Interacción y Mensaje con la información del Incendio Detectado.

Paso 4 → [SAT B] Ejecución del servicio de mensajería a Tierra

Los subpasos se detallan a continuación y se representan en las figuras 6.9 y 6.10.

- 4.1) El protocolo de interacción en órbita entre el satélite A y B finaliza con la aceptación del servicio de mensajería por parte del satélite A y el cual deberá ejecutar el satélite B. Una vez recibida la aceptación (y los detalles de la misma) por parte del Broker, este notifica acerca del nuevo estado de la orden al Gestor de Órdenes.
- 4.2) El Gestor de Órdenes actualiza el estado de la orden en la Orders DB, que pasa a estar “Active-EXE”, y notifica al Ejecutor del Plan para que retome el control de ejecución del plan.
- 4.3) El Ejecutor del Plan envía una solicitud a CDH para que descargue el mensaje a tierra una vez que el satélite establezca contacto con la estación terrena.
- 4.4) Se establece el contacto con la ET, se descarga el mensaje a tierra y CDH notifica al Ejecutor del Plan del nuevo estado de la orden.
- 4.5) El ejecutor del Plan reenvía el nuevo estado de la orden al Gestor de Órdenes
- 4.6) El Gestor de Órdenes actualiza el estado de la orden y esta pasa a “Finished”.

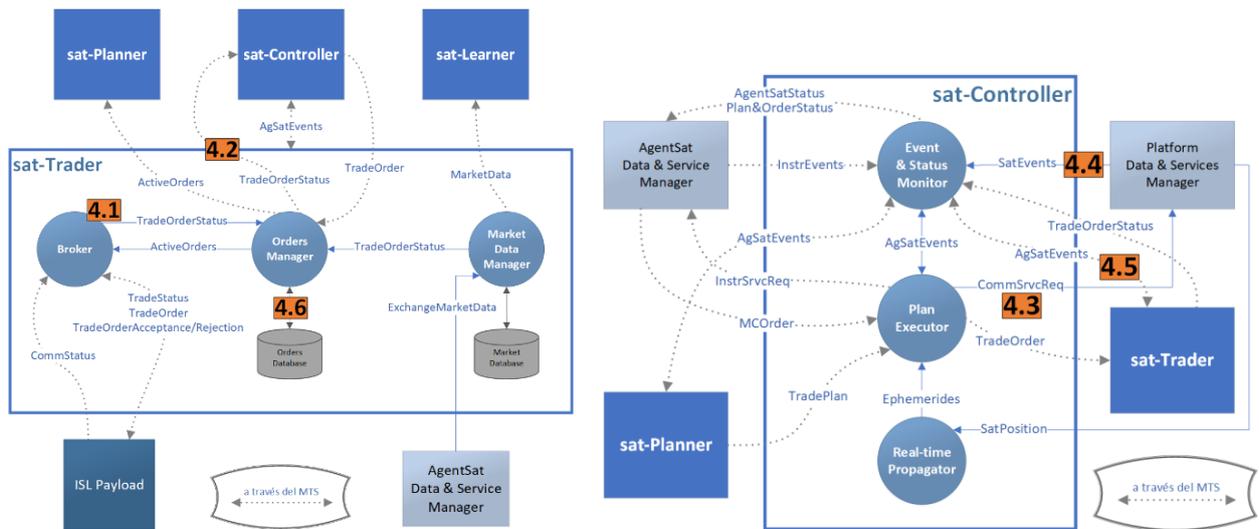


Figura 6.9 – Asociación entre los subpasos del Paso 4 y las unidades del sat-Trader y sat-Controller

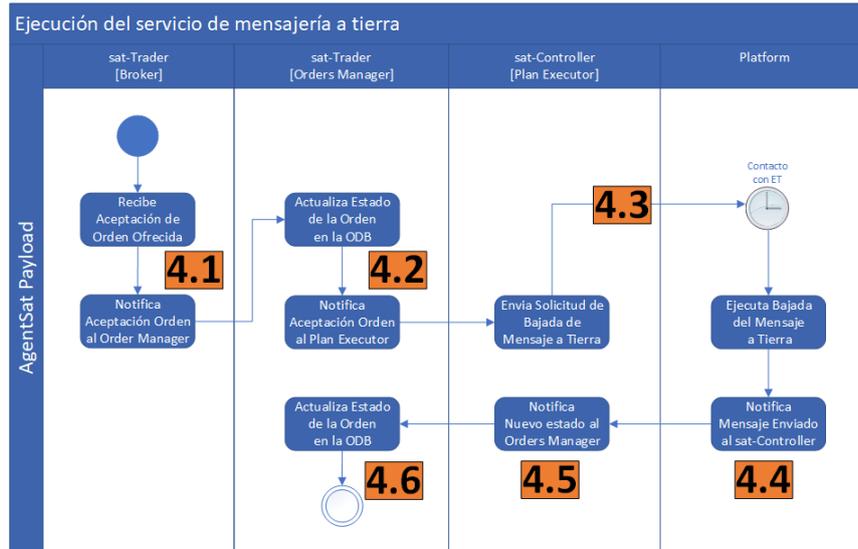


Figura 6.10 – Diagrama de Actividad: Ejecución del Servicio de Mensajería a Tierra.

Paso 5 → [ET B] Recepción y Transmisión del Mensaje al SSEP

Los subpasos se detallan a continuación y se representan en la figura 6.11.

- 5.1) La ET B recibe el mensaje a través de su sistema de antena. Todo el flujo de datos es recepcionado por el Data Acquisition Unit (DAU). Hasta aquí el flujo sería el mismo que el de los datos de ciencia propios de la misión.
- 5.2) El DAU identifica la recepción de un mensaje generado por el AgentSat y lo retransmite hacia la extensión de la Estación Terrena.
- 5.3) El Extension Message Handler (XMH) recibe el mensaje y lo transmite al SSEP a través de Internet.

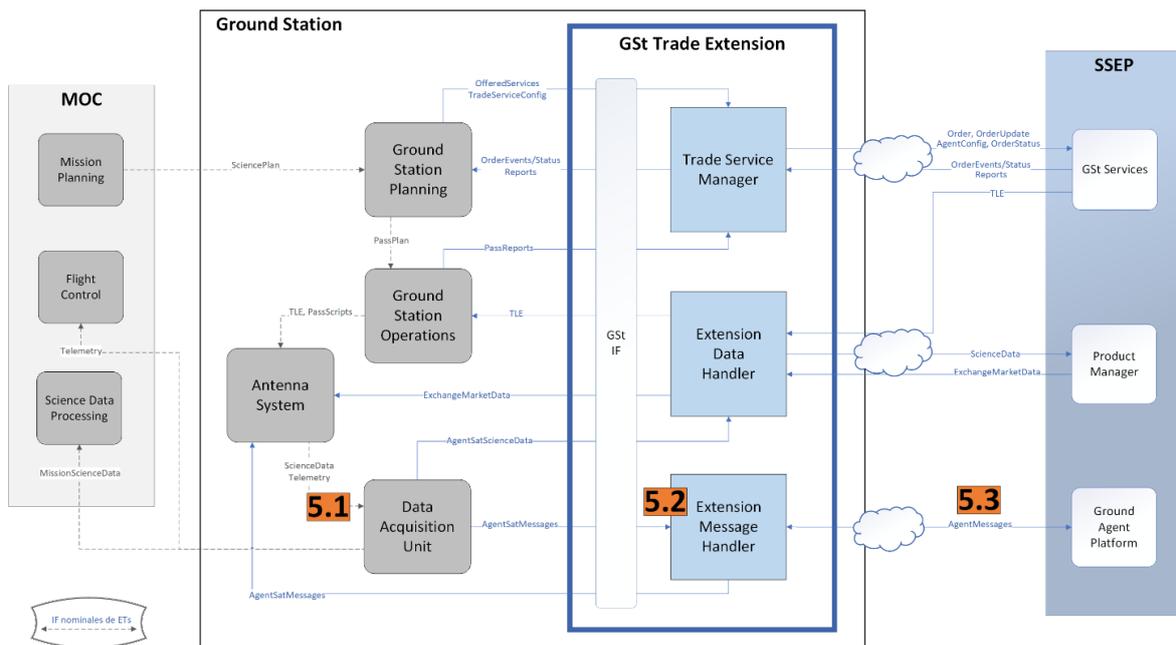


Figura 6.11 – Unidades de la ET involucradas en la recepción y transmisión del mensaje a la Plataforma

Paso 6 → [SSEP] *Recepción de Mensaje en el SSEP y envío al Agente del Destinatario*

Los subpasos se detallan a continuación y se representan en las figuras 6.12 y 6.14.

- 6.1) El mensaje ingresa el SSEP a través del MTS y es recepcionado por el Trade Manager.
- 6.2) El Trade Manager procesa el mensaje, lo da de alta en la base de datos de Transacciones y actualiza su estado como “Recibido en SSEP”.
- 6.3) El Trade Manager identifica los destinatarios, envía el mensaje a cada uno de ellos y actualiza el estado del mensaje a “Enviado a Destinatario”. En este caso, los destinatarios son los usuarios que están suscriptos a la zona del incendio detectado. Ellos son los clientes de la información de origen espacial. El mensaje es enviado al representante del usuario en la Agent Platform, es decir, al usr-Trader.
- 6.4) Asimismo, el Trade Manager notifica al resto de los participantes de la nueva orden de servicio ingresada al SSEP. El TM notifica a los siguientes actores:
 - *MOC B*: es el representante en el SSEP del proveedor del servicio de mensajería satelital. Su cliente es el MOC A.
 - *MOC A*: es el representante en el SSEP del proveedor del servicio de información espacial. A su vez, es el cliente del MOC B.

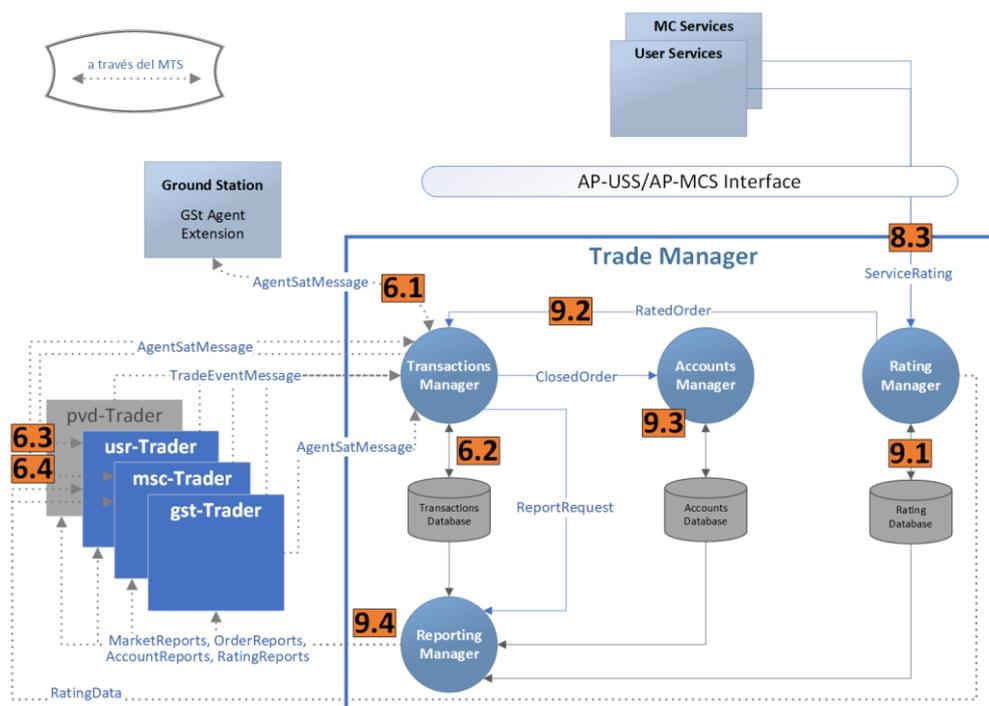


Figura 6.12 – Asociación entre los subpasos del Paso 6, 8 y 9, y las unidades internas del Trade Manager

Paso 7 → [SSEP] *Gestión del mensaje por el Agente del Usuario*

Los subpasos se detallan a continuación y se representan en las figuras 6.13 y 6.14.

- 7.1) El Broker del usr-Trader recibe el mensaje, lo decodifica y se lo envía al usuario a través del User Services.
- 7.2) El Orders Manager actualiza la orden vinculada al mensaje en la User Orders DB.

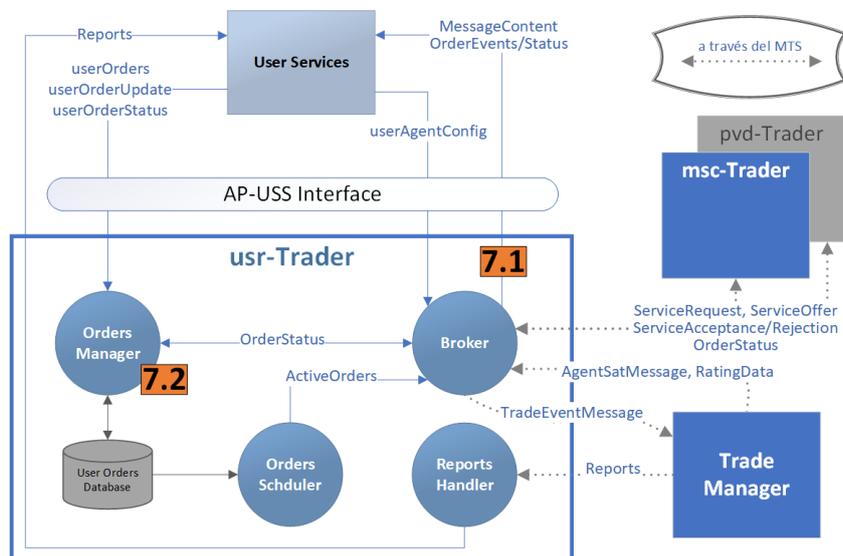


Figura 6.13 – Asociación entre los subpasos del Paso 7 y las unidades internas del usr-Trader

Paso 8 → [Usuario] *Recepción del Mensaje, Gestión de la Información y Calificación del Servicio*

Los subpasos se detallan a continuación y se representan en las figuras 6.12 y 6.14.

- 8.1) El usuario recibe el mensaje y analiza su información para definir las acciones a las que se dará curso.
- 8.2) El usuario ejecuta todas las acciones correspondientes al incendio detectado en la provincia de Córdoba, para prevenirla y/o mitigarla.
- 8.3) Al concluir con la gestión de la emergencia, el usuario interactúa finalmente con el SSEP para calificar el servicio adquirido. La calificación se relaciona con los parámetros que el usuario configuró en la orden de suscripción (veracidad de la información, coordenadas geográficas, tipo de evento detectado, etc.). La información provista por el satélite A resultó ser correcta y acorde a lo solicitado por lo que fue calificada positivamente. El servicio efectivamente brindó información sobre la ubicación e intensidad de un foco de incendio en una zona agreste remota, que las organizaciones civiles en tierra no habían detectado aún.

Paso 9 → [SSEP] *Liquidación del Servicio realizado y notificación a los Participantes*

Los subpasos se detallan a continuación y se representan en las figuras 6.12 y 6.14.

- 9.1) El Rating Manager (RTM) procesa las calificaciones recibidas y actualiza la base de datos. En este caso particular el RTM recibió dos calificaciones:

- *Calificación del Usuario*: fue positiva, lo cual mejoró la posición del Satélite A en el ranking de “proveedores de información de origen espacial”.
- *Calificación del MOC A*: fue positiva, mejorando la posición del Satélite B en el ranking de “proveedores de mensajería espacial”.

9.2) Una vez obtenidas las calificaciones, el Transactions Manager (TRM) actualiza la base de datos de transacciones y actualiza el estado de la orden (Servicio Calificado).

9.3) El Account Manager efectúa el cierre de la orden de servicio y a seguidamente actualiza los estados de cuenta de cada uno de los participantes de la transacción.

9.4) El último paso del proceso es ejecutado por el Reporting Manager, quien elabora un informe que resume las transacciones del servicio llevado a cabo, y lo envía a todos los involucrados en el mismo.

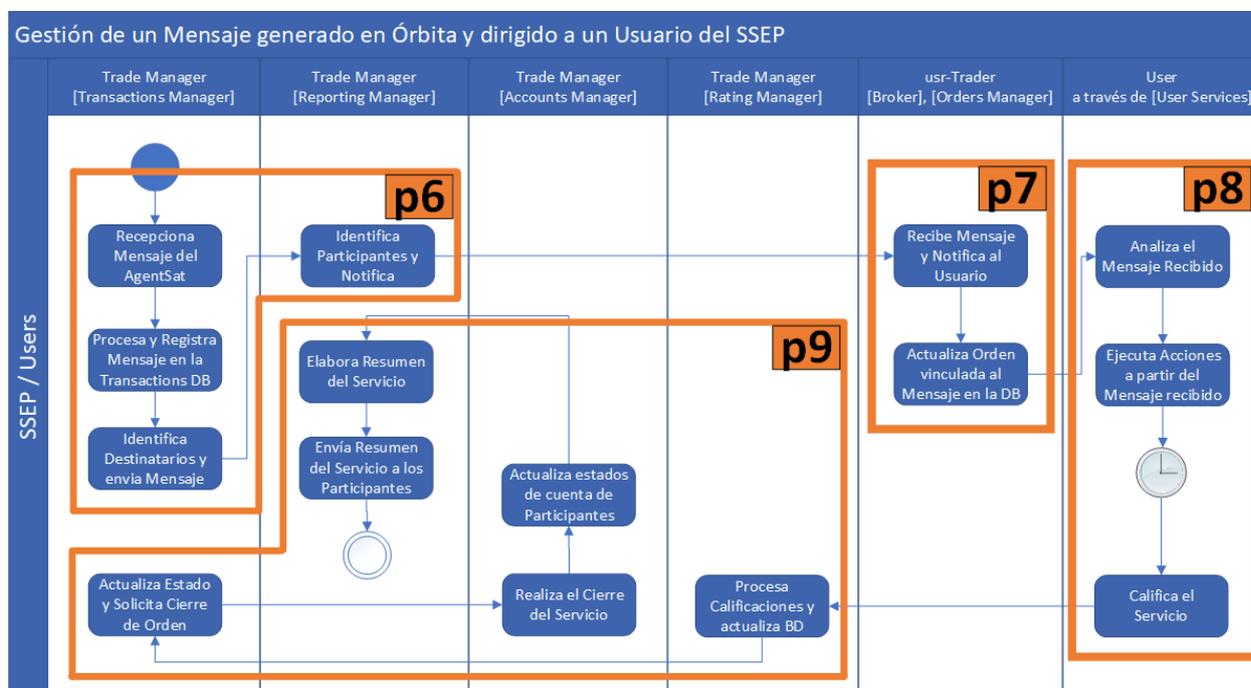


Figura 6.14 – Diagrama de Actividad: Gestión de un mensaje generado en órbita dirigido a un Usuario.

6.2 Caso de Aplicación 2: Soporte a una Emergencia detectada en Tierra

6.2.1 Presentación del caso

En esta sección se describe la dinámica del sistema cuando este es utilizado para dar soporte a una emergencia detectada en tierra. El resumen conceptual del caso se presenta en la figura 6.15.

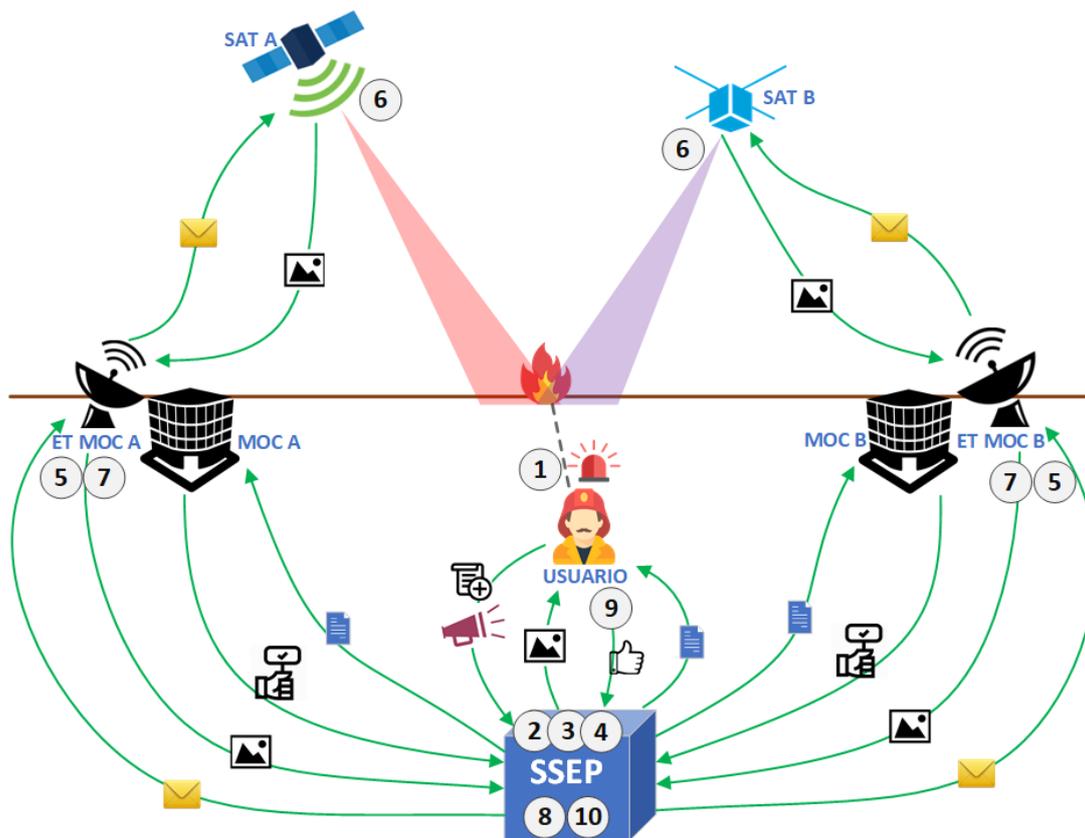


Figura 6.15 – Esquema: Soporte a una Emergencia detectada en Tierra

6.2.2 Descripción detallada del caso

Paso 1 → [Usuario] Detección de un Incendio en tierra

Los subpasos se detallan a continuación.

- 1.1) Personal de defensa civil y bomberos de la provincia de Córdoba detecta un incendio forestal de grandes proporciones que se está extendiendo con rapidez.
- 1.2) Luego de un análisis de situación, el equipo decide generar una solicitud de servicio a través del SSEP para adquirir, por el lapso de una semana, dos tipos de imágenes satelitales (X e Y) en tiempo real sobre la zona del evento. Con el soporte de las imágenes, se espera realizar un análisis más preciso de la dinámica del incendio y sofocarlo más rápidamente. El equipo solicitará imágenes de instrumentos A, en la frecuencia J y con una resolución de K.

Paso 2 → [SSEP] Alta de Pedido en el SSEP para dar soporte a la gestión de la Emergencia

Los subpasos se detallan a continuación y se representan en las figuras 6.16.

- 2.1) El usuario del SSEP (Bomberos y Defensa Civil) da de alta una nueva solicitud de servicio en la plataforma. La orden incluye los requerimientos técnicos (las características de los tipos de imágenes, de los instrumentos, las frecuencias del instrumento, resolución, coordenadas geográficas, frecuencia de adquisición, etc.), el plazo de vigencia de la orden y el precio ofrecido por cada una de las imágenes. La orden será del tipo subasta inversa y se emitirán dos solicitudes de servicio (una por cada tipo de imagen).
- 2.2) El Orders Manager da de alta la nueva orden en la base de datos.
- 2.3) Dado que la orden ha sido emitida para ejecutarse en tiempo real, el Orders Scheduler envía la orden al Broker para que sea gestionada al instante

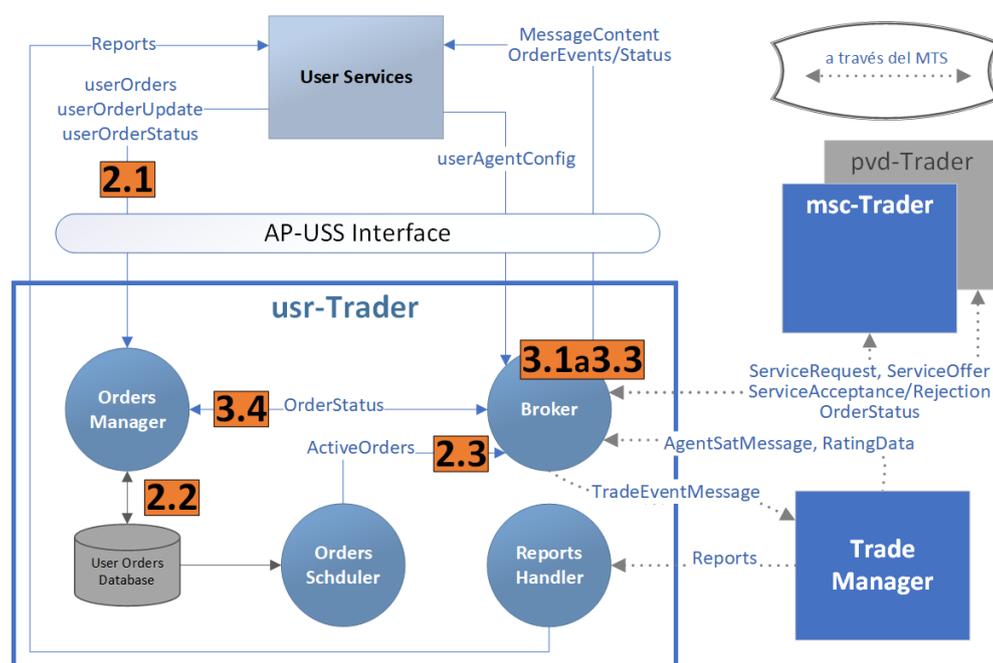


Figura 6.16 – Asociación entre los subpasos del Paso 2 y las unidades internas del usr-Trader

Paso 3 → [SSEP] Contratación del Servicio de Imágenes Satélites mediante Subasta inversa

Los subpasos se detallan a continuación y se representan en las figuras 6.16.

- 3.1) Una vez que el Broker del usuario recibe la orden de tiempo real, realiza una consulta al Agent Management System (AMS) para obtener los ID de los agentes que ofrecen el servicio de provisión de imágenes satelitales del tipo X y del tipo Y.
- 3.2) El Broker realiza una consulta al Trade Manager para conocer el rating de cada uno de los agentes que ha arrojado la búsqueda. Aquellos agentes que tienen una calificación menor a un determinado umbral, son descartados como destinatarios del mensaje.
- 3.3) El Broker crea todos los mensajes que serán intercambiados durante la comunicación y a continuación ejecuta ambos protocolos FIPA-ContractNet. Las dos solicitudes de servicios son enviadas a los agentes

preseleccionados. Ambos protocolos de interacción finalizan con resultados positivos: se han conseguido dos proveedores que cumplirán con las condiciones del pedido. El MOC A proveerá las imágenes satelitales del tipo X y el MOC B las imágenes del tipo Y.

3.4) El gestor de órdenes actualiza el estado de la orden en la base de datos (“Active-TRD”)

Paso 4 → [SSEP] *Gestión de la Orden por parte de los Proveedores del Servicio (MOC A y B)*

Los subpasos se detallan a continuación y se representan en las figuras 6.17 y 6.18.

- 4.1) Luego de haber aceptado la provisión de servicio al Usuario, el Broker del MOC A/B envía los detalles técnicos de la solicitud al Orders Manager, y este último actualiza la orden en su base de datos y su estado (“Active-TRD”).
- 4.2) El Broker envía al Trade Manager el mensaje dirigido para el AgentSat A/B.
- 4.3) El Trade Manager envía el mensaje a las ETs correspondientes (definidas en los detalles de la orden) y registra la salida del mensaje del SSEP.

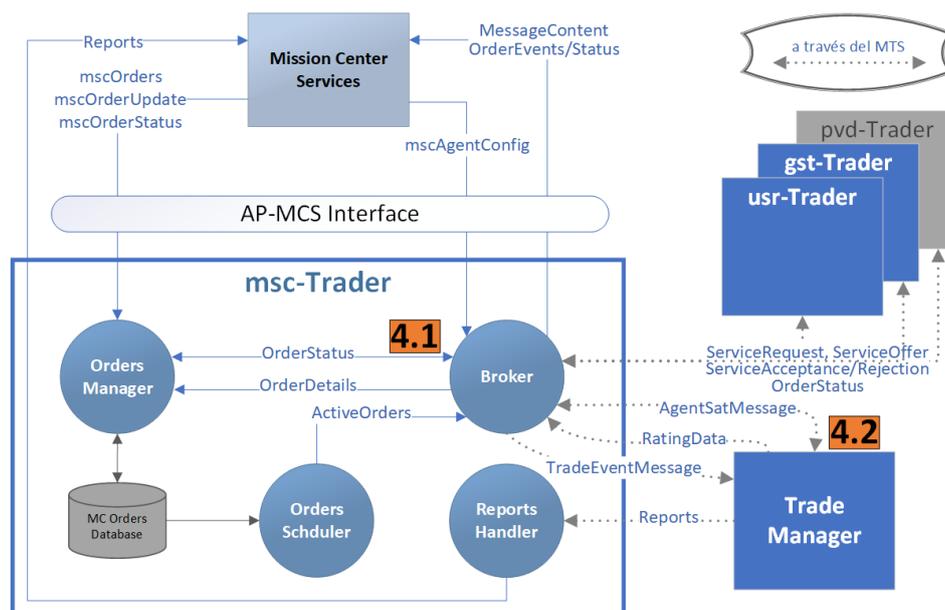


Figura 6.17 – Asociación entre los subpasos del Paso 4 y las unidades internas del msc-Trader

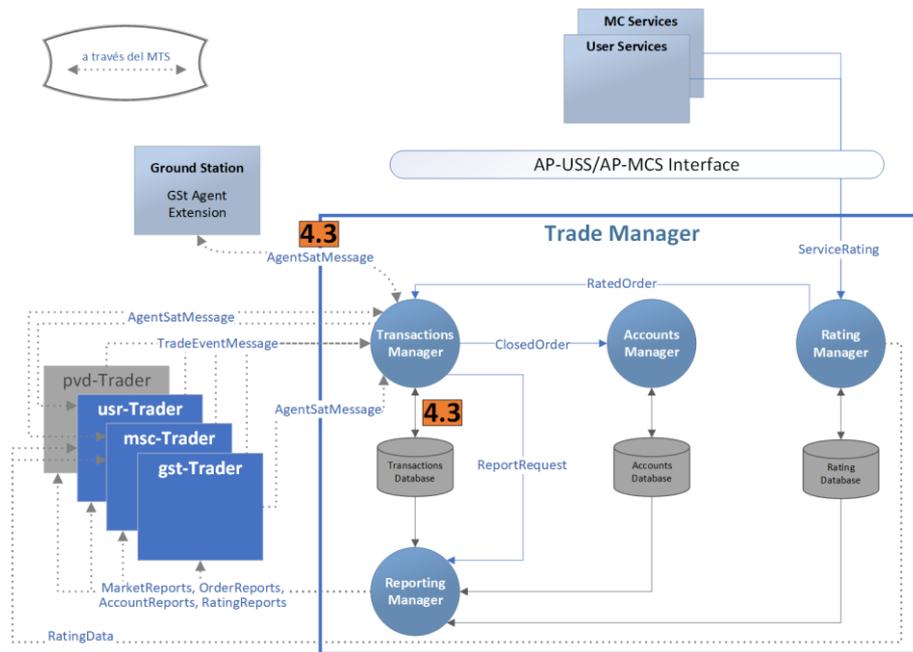


Figura 6.18 – Asociación entre los subpasos del Paso 4 y las unidades internas del Trade Manager

Paso 5 → [ET] Las ETs A y B suben los Mensajes a los AgentSat A y B

Los subpasos se detallan a continuación y se representan en las figuras 6.19.

- 5.1) La unidad Extension Message Handler (XMH) es la encargada de recepcionar, en la ET, el mensaje dirigido al AgentSat. Ella forma parte de la extensión de la arquitectura de la estación para hacerla compatible con el SSEP.
- 5.2) El AgentSatMessage es transmitido al satélite durante la pasada, luego de que el script del mensaje haya sido validado por el sistema de operaciones de la ET.

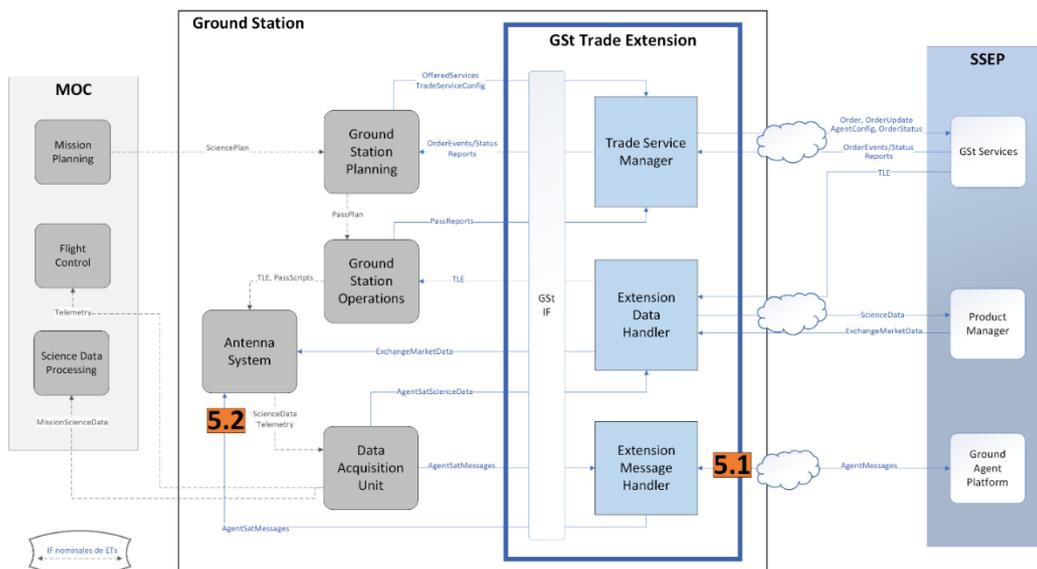


Figura 6.19 – Asociación entre los subpasos del Paso 5 y las unidades internas del GSt Agent Extension

Paso 6 → [SAT A y B] Los AgentSats A y B gestionan las Órdenes de Servicio generadas por sus MOCs

Los subpasos se detallan a continuación y se representan en las figuras 6.20 y 6.21.

- 6.1) El mensaje para el AgentSat es recibido por el sistema de comunicaciones del satélite, gestionado por el subsistema CDH y recepcionado por la unidad Agent Data & Service Manager (ADSM).
- 6.2) El ADSM transfiere el mensaje al sat-Controller quien se encarga de la ejecución de los planes, tanto los generados a bordo como los generados en tierra. En este caso en particular, el plan fue generado en tierra por el MOC que opera el satélite. Dado que la autonomía del satélite está configurada como BAJA, el estado de las actividades del AgentSat es completamente conocido desde tierra, por lo cual, esta nueva orden es ejecutado de inmediato ya que no puede entrar en conflicto con ninguna otra.
- 6.3) El sat-Controller envía la orden de adquisición al instrumento (a través del ADSM) y espera a que la misma haya sido generada.
- 6.4) Una vez que la imagen ha sido generada, la misma es almacenada en la memoria masiva hasta que se establezca el contacto con la estación. El sat-Controller envía una orden de descarga a tierra.
- 6.5) Al establecerse el enlace con la ET, se descarga la imagen a tierra haciendo uso del subsistema de comunicaciones del satélite.

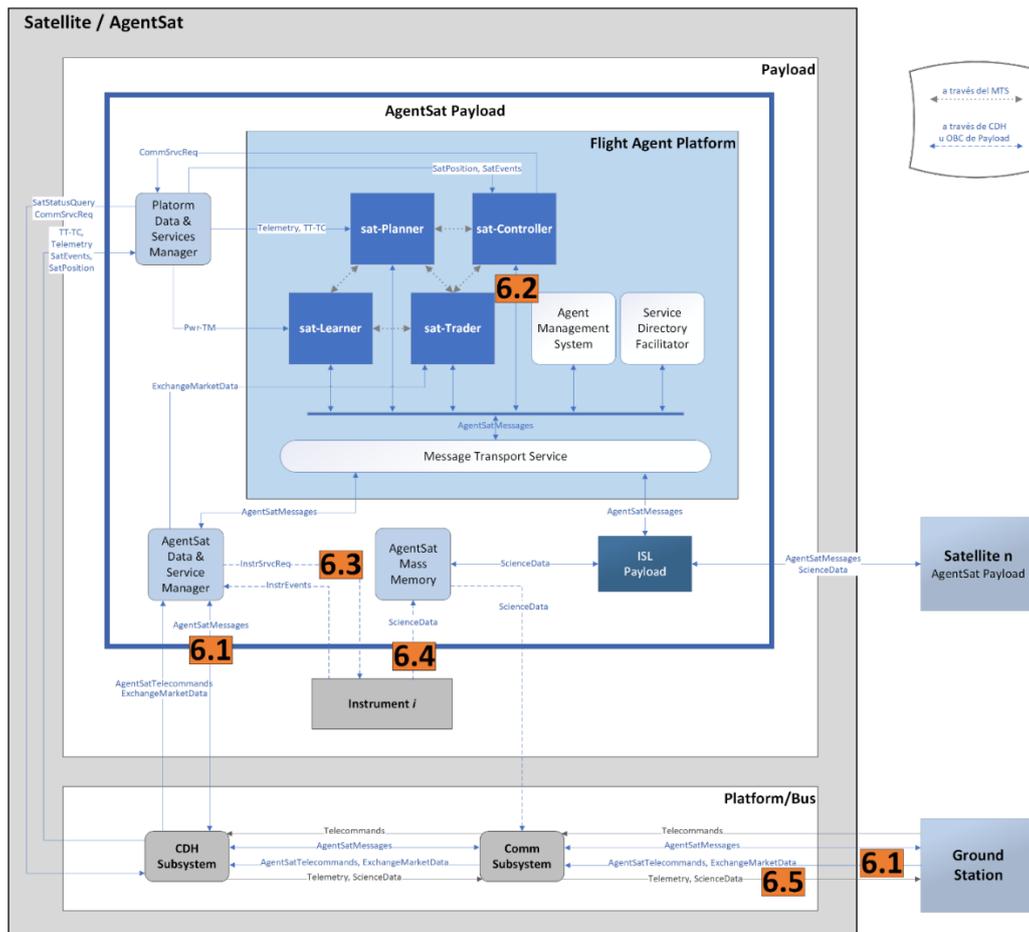


Figura 6.20 – Asociación entre los subpasos del Paso 6 y las unidades internas del AgentSat

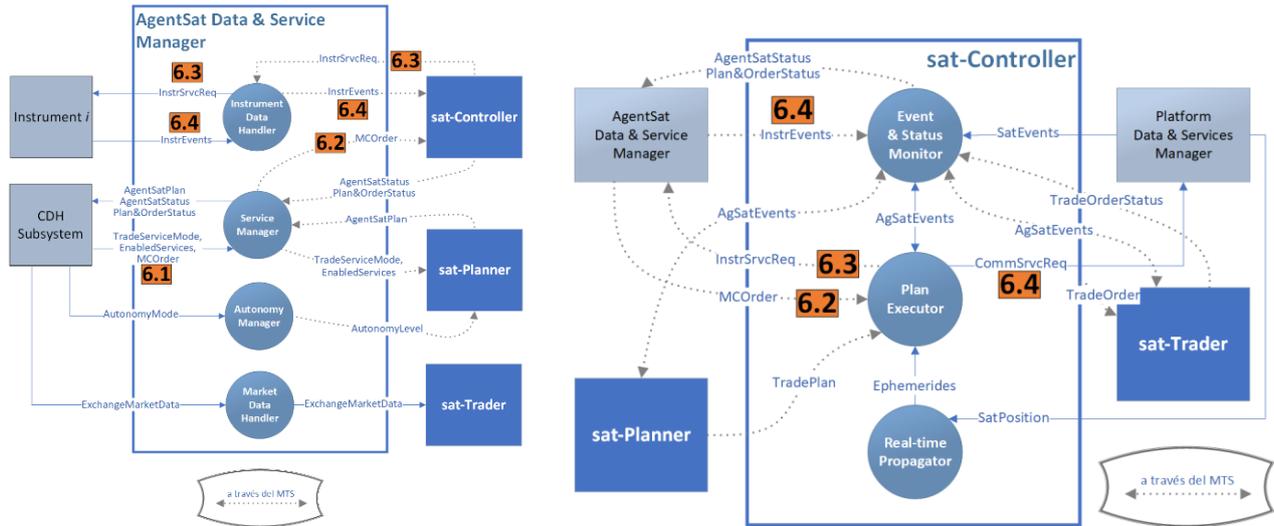


Figura 6.21 – Asociación entre los subpasos del Paso 6 y las unidades internas del AgentSat Data & Service Manager y el sat-Controller

Paso 7 → [ET A y B] Las ET A y B reciben las imágenes de los AgentSat A y B y las suben al SSEP

Los subpasos se detallan a continuación y se representan en las figuras 6.22.

- 7.1) La imagen es descargada en la estación y del sistema de antena pasa al Data Acquisition Unit.
- 7.2) La imagen es recepcionada en la extensión AgentSat de la ET por la unidad Extension Data Handler (XDH), quien al mismo tiempo es la encargada de enviarla al SSEP.

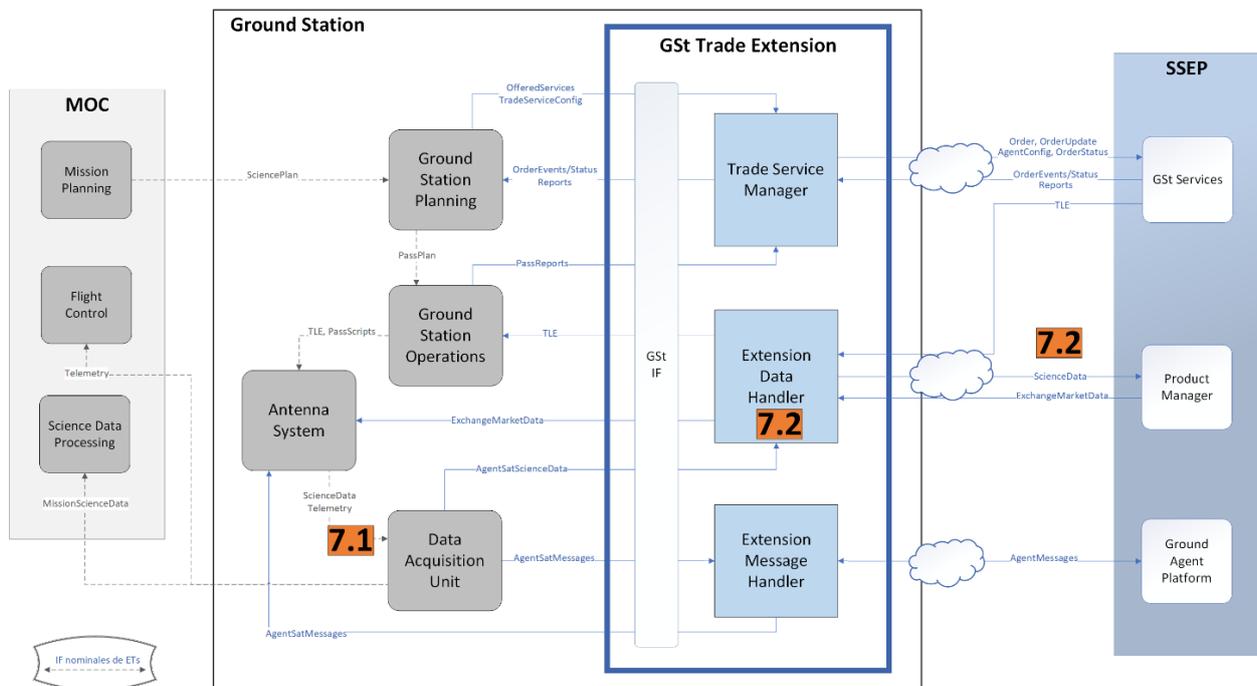


Figura 6.22 – Asociación entre los subpasos del Paso 7 y las unidades internas del GSt Agent Extension

Paso 8 → [SSEP] *Recepción de las Imágenes en el SSEP y puesta a disposición del Usuario*

Los subpasos se detallan a continuación y se representan en las figuras 6.23.

- 8.1) Las imágenes obtenidas por los satélites A y B ingresan al SSEP por la unidad Image Manager (IMM) dentro del subsistema Product Manager (PRM).
- 8.2) El IM genera los permisos para que el Usuario pueda acceder a las imágenes desde el Data & Messages Manager (DMM), y desde aquí hacer una descarga local.
- 8.3) La notificación de que la imagen ha sido recibida en el SSEP y los cambios de estados correspondientes a la orden, se implementa mediante el envío de mensajes desde el Trade Manager hacia los agentes representantes de la transacción, a saber: msc-Trader A, msc-Trader B y usr-Trader.

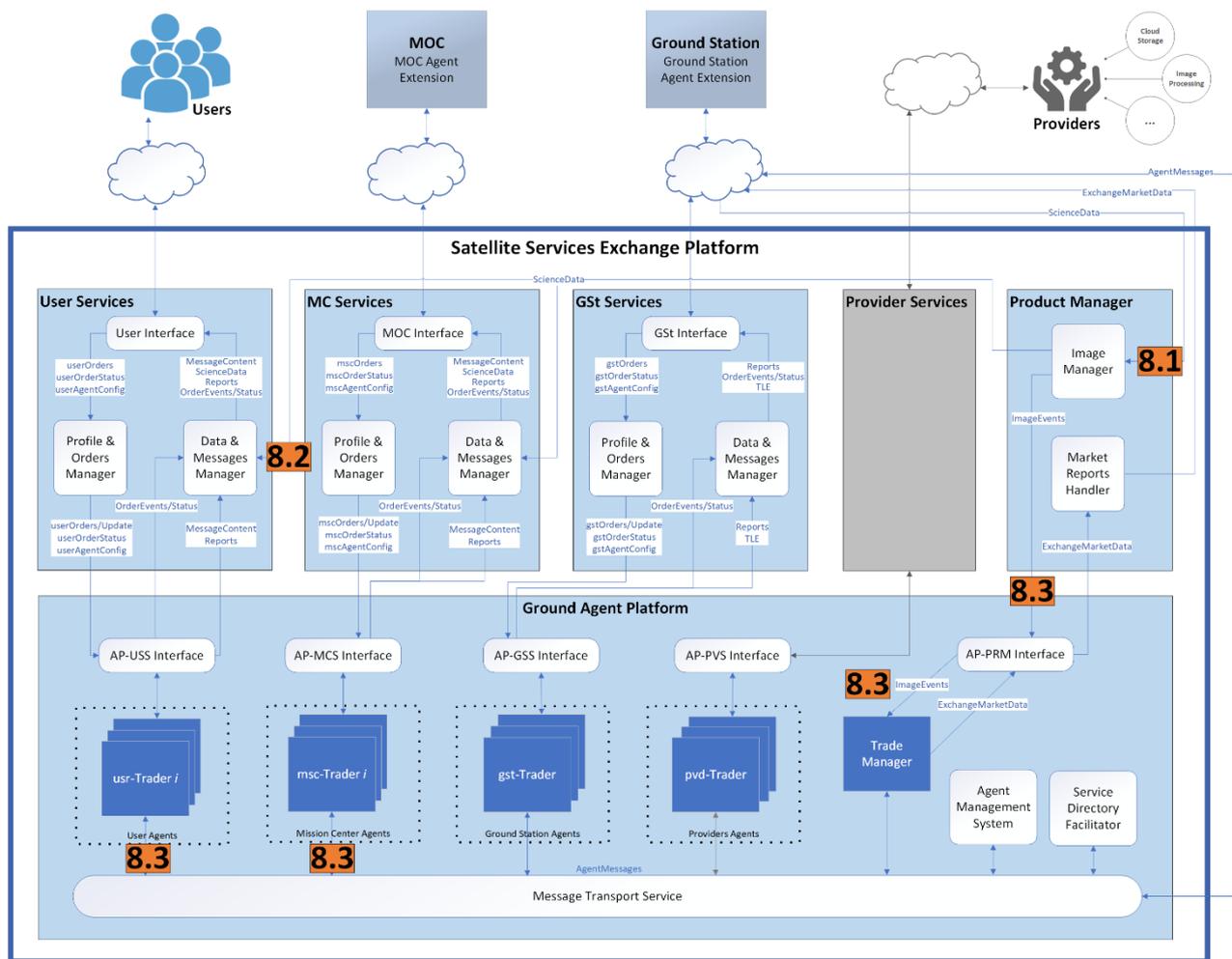


Figura 6.23 – Asociación entre los subpasos del Paso 8 y las unidades internas del SSEP

Paso 9 → [Usuario] *Recepción de las Imágenes, Gestión del Producto y Calificación del Servicio*

Los subpasos se detallan a continuación y se representan en las figuras 6.24.

- 9.1) El usuario recibe las imágenes y analiza su información para definir las acciones a las que se dará curso.
- 9.2) El usuario ejecuta todas las acciones correspondientes al incendio detectado en la provincia de Córdoba, para prevenirla y/o mitigarla.
- 9.3) Una vez finalizada la gestión de la emergencia, el usuario interactúa finalmente con el SSEP para calificar el servicio/producto adquirido. La calificación se relaciona con los parámetros que el usuario configuró en la orden de suscripción (calidad del producto, rapidez del servicio, etc.). Las imágenes provistas por los satélites A y B cumplieron con los requerimientos, por lo que fueron calificadas positivamente. Los productos recibidos fueron un gran soporte para gestionar y mitigar la emergencia.

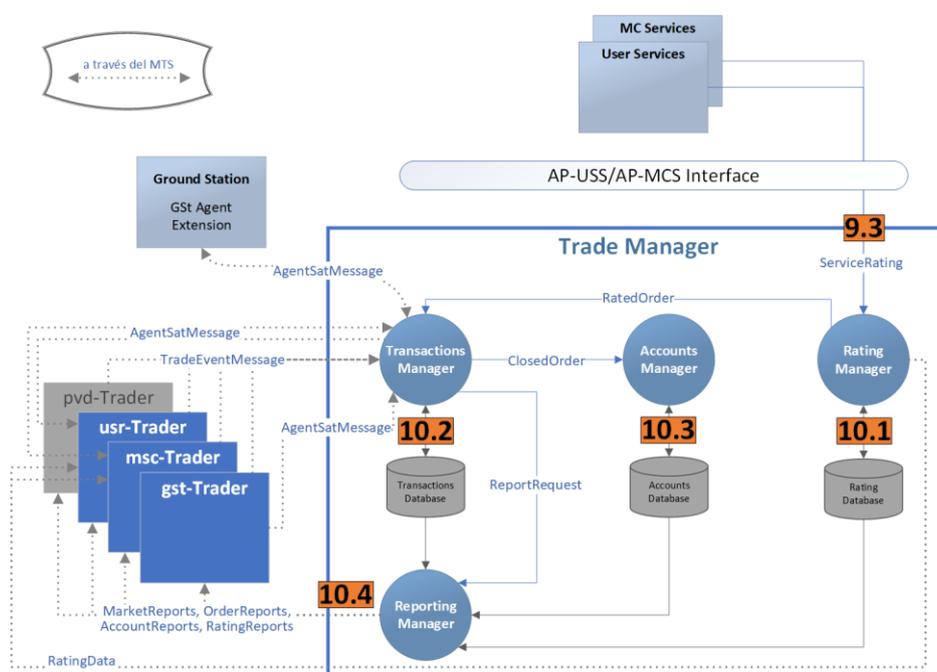


Figura 6.24 – Asociación entre los subpasos del Paso 9 y las unidades internas del SSEP

Paso 10 → [SSEP] *Liquidación del Servicio realizado y notificación a los Participantes*

Los subpasos se detallan a continuación y se representan en las figuras 6.26.

- 10.1) El Rating Manager (RTM) procesa las calificaciones recibidas y actualiza la base de datos. En este caso particular el RTM recibió dos calificaciones:
 - *Calificación por parte del Usuario hacia el MOC A:* fue positiva, lo cual mejoró la posición del Satélite A en el ranking de “proveedores de imágenes satelitales”.
 - *Calificación por parte del Usuario hacia el MOC B:* fue positiva, lo cual mejoró la posición del Satélite B en el ranking de “proveedores de imágenes satelitales”.

- 10.2) Una vez obtenidas las calificaciones, el Transactions Manager actualiza la base de datos de transacciones y actualiza el estado de la orden (Servicio Calificado).
- 10.3) El Account Manager efectúa el cierre de la orden de servicio y a seguidamente actualiza los estados de cuenta de cada uno de los participantes de la transacción.
- 10.4) El último paso del proceso es ejecutado por el Reporting Manager, quien elabora un informe que resume las transacciones del servicio llevado a cabo, y lo envía a todos los involucrados en el mismo.

Capítulo 7

CONCLUSIONES Y TRABAJO FUTURO

7.1 Conclusiones

El objetivo de la tesis era elaborar el diseño de una arquitectura satelital segmentada basada en sistemas multiagente para la gestión de emergencias. En el capítulo 1 se mencionó acerca de las bondades de los sistemas satelitales federados para el caso de la aplicación en cuestión, lo que motivó que finalmente el diseño se elabore tomando este tipo de SED como base. Considerando el concepto de sistema que se había ideado, las implicancias de esta adaptación fueron despreciables ya que las arquitecturas segmentadas pueden considerarse como una instancia de las federaciones, es decir, como un caso más restrictivo en el que los segmentos son operados o administrados por la misma organización.

En el capítulo 2 se presentó el marco teórico que introducía y describía con alto grado de detalle todos los conceptos o elementos necesarios para una correcta comprensión de la tesis. A continuación, en el capítulo 3 se tomaron aquellos tópicos considerados como el núcleo de la tesis (arquitecturas segmentadas, sistemas federados y sistemas multiagente) para realizar una exhaustiva revisión bibliográfica. El objetivo fue buscar antecedentes concretos que permitieran detectar las mejores líneas de investigación, a partir de una primera idea inicial. El resultado del análisis se presentó en el capítulo 4 y de él surgieron ajustes en la aproximación al diseño (ya mencionados). También se llegó a la conclusión de que no existían antecedentes concretos que sean comparables con la tesis, aunque sí se detectaron algunos elementos en común en dos trabajos de los relevados.

Una vez analizada la bibliografía, ajustado el tipo de SED a utilizar en el diseño y delimitado el alcance de la tesis, se fue desarrollando y documentando la arquitectura en el capítulo 5. Y, en la medida que se la instanciaba para los casos de aplicación, se la iba refinando. Es decir, los casos de aplicación y el diseño se fueron elaborando en paralelo y ambos procesos se retroalimentaban entre sí. Luego de varias iteraciones,

finalmente se llegó a la versión final del sistema y se describieron los dos casos de validación en el capítulo 6, los más representativos a criterio del autor.

En conclusión, los resultados de la tesis son los siguientes:

- El sistema incluye todos los elementos y *stakeholders* de una misión satelital tipo, y algunos más, con el propósito de obtener una correcta visión sistémica de un eventual mercado de servicios satelitales unificado. También incluye el diseño interno de los agentes para estudiar su interacción, tanto entre los agentes de vuelo como de tierra. La arquitectura propuesta se validó conceptualmente con los casos de aplicación, a partir de los cuales se detalló el concepto de operación y se pudo ejemplificar su dinámica.
- El sistema incluye satélites con un nivel de autonomía configurable (AgentSats) y una arquitectura relativamente genérica, de forma tal que podría servir como base para otras aplicaciones.
- El sistema propuesto incorpora un novedoso elemento (el SSEP) que, en caso que fuese implementado, podría imaginarse una industria espacial más colaborativa, más responsiva, más eficiente, más efectiva en costo y más accesible al público general.
- La arquitectura elaborada es un aporte concreto a la literatura especializada en sistemas espaciales distribuidos y en sistemas multiagente, pudiendo actuar como catalizador de futuras investigaciones.

7.2 Futuras Líneas de Trabajo e Investigación

A partir de la experiencia en el desarrollo de la presente tesis y la contribución realizada en términos de arquitecturas SSF, se prevén las siguientes líneas de trabajo:

- Implementación de la arquitectura a través de algún framework multiagente existente.
- Desarrollo de una plataforma de simulación ad-hoc para el sistema propuesto, de forma tal que permita modelar y evaluar la dinámica del SatSES y, a futuro, otras instancias de SEDs.
- Actualización de la arquitectura tomando como dada la existencia de conectividad de Internet en órbita. Esto permitiría diseñar protocolos de interacción en órbita más complejos, incrementar las capacidades del segmento de vuelo y reducir una buena porción del procesamiento que se da en tierra, optimizando el sistema en su generalidad.

Con respecto a las líneas de investigación, se proponen:

- Profundización de la arquitectura del AgentSat, en particular de su capa de aprendizaje, ya que podría resultar de interés para un sinnúmero de aplicaciones, además del SatSES.
- Diseño de nuevas arquitecturas basadas en el SatSES orientadas a otro tipo de misiones que no sean de observación de la tierra, como, por ejemplo, de exploración planetaria.

Referencias

- Akhtyamov, R., Lluch, I., Matevosyan, H., Knoll, D., Pica, U., Lisi, M. y Golkar, A. (2016). An Implementation of Software Defined Radios for Federated Aerospace Networks: Informing Satellite Implementations Using an Inter-Balloon Communications Experiment. *Acta Astronautica*, 123, 470–478.
- Alfriend, K., Vadali, S. R., Gurfil, P., How, J., y Breger, L. (2010). *Spacecraft formation flying: Dynamics, control and navigation*. Burlington, MA, EEUU: Elsevier.
- Amazon (2019). *Cloud computing with AWS*. Recuperado de <https://aws.amazon.com/what-is-aws/?nc2=h ql le> [2019, 30 de junio].
- Atzori, L., Iera, A., y Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54(15), 2787-2805.
- Bernard, D. E., Gamble, E. B., Rouquette, N. F., Smith, B., Tung, Y. W., Muscettola, N., ... y Nayal, P. (2000). Remote Agent Experiment DS1 Technology Validation Report. *Ames Research Center and JPL*.
- Bhasin, K., y Hayden, J. L. (2001, Mar). Space Internet architectures and technologies for NASA enterprises. En *2001 IEEE Aerospace Conference Proceedings 2*, 2/931-2/941. Big Sky, MT, EEUU.
- Bonabeau, E., Dorigo, M., y Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems* (No. 1). New York, NY, EEUU: Oxford University Press.
- Bonnet, J., Gleizes, M. P., Kaddoum, E., Rainjonneau, S., y Flandin, G. (2015, Sep). Multi-satellite mission planning using a self-adaptive multi-agent system. En *2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems* (pp. 11-20). IEEE.
- Bonnet, G., y Tessier, C. (2007, May). Collaboration among a satellite swarm. En *Proceedings of the 6th International Joint Conference on Autonomous agents and multiagent systems*. Honolulu, HI, EEUU.

- Bratman, M. E., Israel, D. J., y Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational intelligence*, 4(3), 349-355.
- Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1), 14-23.
- Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47(1-3), 139-159.
- Brown, O., y Eremenko, P. (2006a). Fractionated space architectures: A vision for responsive space. En *4th Responsive Space Conference*. Los Angeles, CA, EEUU.
- Brown, O., y Eremenko, P. (2006b). The value proposition for fractionated space architectures. En *Space 2006*. San José, CA, EEUU.
- Brown, O., Eremenko, P., y Collopy, P. (2009, Sep). Value-centric design methodologies for fractionated spacecraft: Progress summary from Phase I of the DARPA System F6 Program. En *AIAA Space 2009 Conference & Exposition* (p. 6540). Pasadena, CA, EEUU.
- Chien, S., Doubleday, J., Thompson, D. R., Wagstaff, K. L., Bellardo, J., Francis, C., ... y Piug-Suari, J. (2017). Onboard Autonomy on the Intelligent Payload EXperiment CubeSat Mission. *Journal of Aerospace Information Systems*, 14(6), 307-315.
- Chu, J., Guo, J., y Gill, E. (2014). Functional, physical, and organizational architectures of a fractionated space infrastructure for long-term earth observation missions. *IEEE Aerospace and Electronic Systems Magazine*, 29(12), 6-17.
- Chu, J., Guo, J., y Gill, E. (2016). Decentralized autonomous planning of cluster reconfiguration for fractionated spacecraft. *Acta Astronautica*, 123, 397-408.
- Colby, M., Yliniemi, L., y Tumer, K. (2016). Autonomous multiagent space exploration with high-level human feedback. *Journal of Aerospace Information Systems*, 13(8), 301-315.
- CONAE (2010). *Plan Espacial Nacional Argentina en el espacio 2004-2015*. Recuperado de <http://www.cetam.fadu.uba.ar/wp-content/uploads/2015/01/Actualizacion-Plan-Espacial-2010-2015.pdf> [2019, 30 de junio].
- CONAE (2019). *Serie SARE*. Recuperado de <http://www.conae.gov.ar/index.php/espanol/misiones-satelitales/sare/objetivos> [2019, 30 de junio].
- Corbin, B. A. (2015). *The Value Proposition of Distributed Satellite Systems for Space Science Missions* (Tesis de Doctorado). Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. Cambridge, MA, EEUU.

- Das, K. S., Gonsalves, P., Krikorian, R., y Truszkowski, W. (1999, Jun). Multi-agent planning and scheduling environment for enhanced spacecraft autonomy. En *Proceedings from the Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space (SP-440)*. Simposio llevado a cabo en Noordwijk, Países Bajos.
- De Selding, P. (2015). Virgin, Qualcomm invest in OneWeb satellite Internet venture. *Spacenews*, Ene 2015. Recuperado de: <https://spacenews.com/virgin-qualcomm-invest-in-global-satellite-internet-plan/> [2019, 30 de junio].
- Dignum, V., y Padget, J. (2013). Multiagent organizations. En Weiss, G. (Ed.), *Multiagent systems* (2da. ed.) (cap. 2, pp. 51-98). Cambridge, MA, EEUU: MIT Press.
- Dubey, A., Emfinger, W., Gokhale, A., Karsai, G., Otte, W. R., Parsons, J., ... y Bose, P. (2012, Mar). A software platform for fractionated spacecraft. En *2012 IEEE Aerospace Conference* (pp. 1-20). IEEE.
- D'Amico, S., Pavone, M., Saraf, S., Alhussien, A., Al-Saud, T., Buchman, S., ... y Farhat, C. (2015, Jun). Miniaturized Autonomous Distributed Space System for Future Science and Exploration. En *International Workshop on Satellite Constellations and Formation Flying*. Workshop llevado a cabo en Delft, Países Bajos.
- D'Amico, S. (2016, Ago). *Satellite Formation Flying for Space Exploration - Simone D'Amico (SETI Talks 2016)* (9m 40s) [Video]. Recuperado de <https://www.youtube.com/watch?v=quVFqnEQogQ>.
- D'Errico, M. (Ed.) (2013). *Distributed space missions for earth system monitoring*. Nueva York, EEUU: Microcosm Press y Springer.
- Escoubet, C. P., Fehringer, M., y Goldstein, M. (2001). Introduction the Cluster mission. En *Annales Geophysicae*, 19, 1197-1200.
- Ferber, J. (1996). Reactive distributed artificial intelligence. En O'Hare G. M., y Jennings, N. R. (Eds.), *Foundations of distributed artificial intelligence*, cap. 11, pp. 287-317. New York, NY, EEUU: John Wiley & Sons.
- Ferster, W. (2013). *DARPA Cancels Formation-flying Satellite Demo*. Recuperado de: <https://spacenews.com/35375darpa-cancels-formation-flying-satellite-demo/> [2019, 30 de junio].
- FIPA (2002a). *FIPA Abstract Architecture Specification* (Document Number SC00001L). Recuperado de: <http://www.fipa.org/specs/fipa00001/index.html> [2019, 30 de junio].
- FIPA (2002b). *FIPA Agent Message Transport Service Specification* (Document Number SC00067F). Recuperado de: <http://www.fipa.org/specs/fipa00067/index.html> [2019, 30 de junio].

- FIPA (2002c). *FIPA Communicative Act Library Specification* (Document Number SC00037J). Recuperado de: <http://www.fipa.org/specs/fipa00037/index.html> [2019, 30 de junio].
- FIPA (2002d). *FIPA Contract Net Interaction Protocol Specification* (Document Number SC00029H). Recuperado de: <http://www.fipa.org/specs/fipa00029/index.html> [2019, 30 de junio].
- FIPA (2002e). *FIPA Propose Interaction Protocol Specification* (Document Number SC00036H). Recuperado de: <http://www.fipa.org/specs/fipa00036/index.html> [2019, 30 de junio].
- FIPA (2002f). *FIPA Request Interaction Protocol Specification* (Document Number SC00026H). Recuperado de: <http://www.fipa.org/specs/fipa00026/index.html> [2019, 30 de junio].
- FIPA (2002g). *FIPA SL Content Language Specification* (Document Number SC00008I). Recuperado de: <http://www.fipa.org/specs/fipa00008/index.html> [2019, 30 de junio].
- FIPA (2004). *FIPA Agent Management Specification* (Document Number SC00023K). Recuperado de: <http://www.fipa.org/specs/fipa00023/index.html> [2019, 30 de junio].
- Fortescue, P., Stark, J., y Swinerd, G. (Eds.). (2003). *Spacecraft systems engineering* (3ra Ed). Chichester, West Sussex, Inglaterra: John Wiley & Sons.
- Frenken, K. (2017). Political economies and environmental futures for the sharing economy. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 375(2095), 20160367.
- Ghaffar, M. A., y Vu, T. T. (2015, Ago). Cloud computing providers for satellite image processing service: A comparative study. En *2015 International Conference on Space Science and Communication (IconSpace)* (pp. 61-64). IEEE.
- Golkar, A. (2013a). Federated Satellite Systems: a vision towards an innovation in space systems design. En *9th IAA Symposium on Small Satellites for Earth Observation*. Berlin, Alemania.
- Golkar, A. (2013b). Architecting federated satellite systems for successful commercial implementation. En *AIAA Space Symposium*. San Diego, CA, EEUU.
- Golkar, A. (2013c). Federated Satellite Systems: A Case Study on Sustainability Enhancement of Space Exploration System Architectures. En *International Astronautical Congress*. Beijing, China.
- Golkar, A., y Lluh, I. (2015). The Federated Satellite Systems paradigm: Concept and business case evaluation. *Acta Astronautica*, 111, 230-248.
- Grogan, P. T., Golkar, A., Shirasaka, S., y de Weck, O. L. (2014, Mar). Multi-stakeholder interactive simulation for federated satellite systems. En *2014 IEEE Aerospace Conference* (pp. 1-15). IEEE.

- Grogan, P. T., Ho, K., Golkar, A., y de Weck, O. L. (2016). Multi-actor value modeling for federated systems. *IEEE Systems Journal*, 12(2), 1193-1202.
- Guttentag, D. (2015). Airbnb: disruptive innovation and the rise of an informal tourism accommodation sector. *Current Issues in Tourism*, 18(12), 1192-1217.
- Haddadi, A., y Sundermeyer, K. (1996). Belief-desire-intention agent architectures. En O'Hare G. M., y Jennings, N. R. (Eds.), *Foundations of distributed artificial intelligence*, cap. 5, pp. 169-185. New York, NY, EEUU: John Wiley & Sons.
- Hoebeke, J., Moerman, I., Dhoedt, B., y Demeester, P. (2004). An overview of Mobile Ad Hoc Networks: applications and challenges. *Journal-Communications Network*, 3(3), 60-66.
- Hofmann-Wellenhof, B., Lichtenegger, H., y Collins, J. (2012). *Global positioning system: theory and practice* (5ta Ed. revisada). New York, NY, EEUU: Springer.
- Kramer, H. J. (2019). *Iridium NEXT (Hosting Payloads on a Communications Constellation)*. Recuperado de <https://earth.esa.int/web/eoportal/satellite-missions/i/iridium-next> [2019, 30 de junio].
- Krishna, V. (2010). *Auction theory* (2da Ed). Burlington, MA, EEUU: Academic Press/Elsevier.
- Kusza, K., y Paluszek, M. (2000, Nov). Intersatellite Links: Lower Layer Protocols for Autonomous Constellations. En *1st Joint Space Internet Workshop*. Greenbelt, MD, EEUU.
- Leitner, J. (2002, Abr). Distributed Space Systems: Mission Concepts, Systems Engineering, and Technology Development. En *Systems Engineering Seminar*, NASA-GSFC. Recuperado de https://ses.gsfc.nasa.gov/ses_data_2002/020402_Leitner_DSS.pdf [2019, 30 de junio].
- Leitner, J. (2010, May). Simulating resource sharing in spacecraft clusters using multi-agent-systems. En *Small Satellite Systems and Services – The 4S Symposium*. Simposio llevado a cabo en Funchal, Madeira, Portugal.
- Ley, W., Wittmann, K., y Hallmann, W. (Eds.). (2009). *Handbook of Space Technology*. John Wiley & Sons.
- Li, J., Chen, Y., Liu, X., y He, R. (2018, May). JADE implemented multi-agent based platform for multiple autonomous satellite system. En *2018 SpaceOps Conference*. Marsella, Francia.
- Lluch, I., y Golkar, A. (2014a). Satellite-to-satellite coverage optimization approach for opportunistic inter-satellite links. En *2014 IEEE Aerospace Conference* (pp. 1-13). IEEE.
- Lluch, I., y Golkar, A. (2014b). Resource balancing analysis of federated satellite systems. En *AIAA Space 2014 Conference and Exposition*. San Diego, CA, EEUU.

- Lluch, I., y Golkar, A. (2015). Design implications for missions participating in federated satellite systems. *Journal of Spacecraft and Rockets*, 52(5), 1361-1374.
- Lluch, I., Grogan, P. T., Pica, U., y Golkar, A. (2015, Mar). Simulating a Proactive ad-hoc Network Protocol for Federated Satellite Systems. En *2015 IEEE Aerospace Conference* (1-16). IEEE.
- Lluch, I., y Golkar, A. (2019). Architecting federations of systems: A framework for capturing synergy. *Systems Engineering*. <https://doi.org/10.1002/sys.21482>.
- Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering* 1(4), 267-284.
- Manchester, Z., y Peck, M. (2011, Ago). Stochastic space exploration with microscale spacecraft. En *AIAA Guidance, Navigation, and Control Conference* (p. 6648). Portland, OR, EEUU.
- Mas, A. (2005). *Agentes Software y Sistemas Multi-agente: conceptos, arquitecturas y aplicaciones*. Madrid, España: Pearson Educación.
- Matevosyan, H., Lluch, I., Moreno, C. A., Lamb, A., Akhtyamov, R., De Angelis, G. ... Camps, A. (2016). Stakeholders Needs: User Needs Analysis and Earth Observation Infrastructure State of the Art Assessment. *Operational Network of Individual Observation Nodes (ONION)*.
- Matevosyan, H., Taylor, C., y Golkar, A. (2015, Ago). Evaluating Virtual Satellite Mission Opportunities. En *AIAA Space 2015 Conference and Exposition*. Pasadena, CA, EEUU.
- Mathieu, C., y Weigel, A. (2005, Oct). Assessing the Flexibility Provided by an On-Orbit Infrastructure of Fractionated Spacecraft. En *Proceedings of the 56th International Astronautical Congress, IAC-05-D3.3.01*. Fukuoka, Japón.
- Mohammed, J (2001). Mission Planning for Formation-Flying Satellite Cluster. En *FLAIRS-01 Proceedings* (pp. 58-62).
- Molette, P., Cougnet, C., Saint-Aubert, P., Young, R. W., y Helas, D. (1984). Technical and economical comparison between a modular geostationary space platform and a cluster of satellites. *Acta Astronautica*, 11(12), 771-784.
- Muscettola, N., Nayak, P. P., Pell, B., y Williams, B. C. (1998). Remote agent: To boldly go where no AI system has gone before. *Artificial Intelligence*, 103(1), 5-47.
- Ntagiou, E., Policella, N., Iacopino, C., Armellin, R., y Donati, A. (2017). Ant Colony Optimization applied to the Planning of a Data Relay Space Mission. En *International Workshop on Planning and Scheduling for Space*.
- Object Management Group [OMG] (2017). *OMG Systems Modeling Language (OMG SysML®)*. Recuperado de <http://www.omg.org/spec/SysML/1.5/> [2019, 30 de junio].

- Odell, J., Parunak, H. V. D., y Bauer, B. (2000, Jul). Extending UML for agents. En *Proceedings of the agent-oriented information systems workshop at the 17th national conference on artificial intelligence* (pp. 3-17).
- O'Neill, M. G., y Weigel, A. L. (2011). Assessing fractionated spacecraft value propositions for earth imaging space missions. *Journal of Spacecraft and Rockets*, 48(6), 974-986.
- Osborne, M. y Rubinstein, A (1994). *A Course in Game Theory*. Cambridge, MA, EEUU: MIT Press.
- Pica, U. y Golkar, A. (2017). Sealed-Bid Reverse Auction Pricing Mechanisms for Federated Satellite Systems. *Systems Engineering*, 20(5), 432-446.
- Poghosyan, A., Lluch, I., Matevosyan, H., Lamb, A., Moreno, C. A., Taylor, C., ... Camps, A. (2016a). Unified Classification for Distributed Satellite Systems. En *4th International Federated and Fractionated Satellite Systems Workshop*. Workshop llevado a cabo en Roma, Italia.
- Poghosyan, A., Lluch, I., Matevosyan, H., Lamb, A., Moreno, C. A., Taylor, C., ... Camps, A. (2016b). Fractionated and Federated Technology State of the Art Survey. Operational Network of Individual Observation Nodes (ONION).
- Poslad, S. (2007). Specifying protocols for multi-agent systems interaction. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 2(4), 15.
- Puschmann, T., y Alt, R. (2016). Sharing economy. *Business & Information Systems Engineering*, 58(1), 93-99.
- Raman, S., Weigel, R., y Lee, T. (2016). The Internet of space (IoS): A future backbone for the Internet of Things?. *IEEE IoT Newsletter*.
- Rayman, M. D., Varghese, P., Lehman, D. H., & Livesay, L. L. (2000). Results from the Deep Space 1 technology validation mission. *Acta Astronautica*, 47(2-9), 475-487.
- Ruiz de Azúa, J. A., Calveras, A., y Camps, A. (2018). Internet of satellites (IoSat): Analysis of network models and routing protocol requirements. *IEEE access*, 6, 20390-20411.
- Russell, S., y Norvig, P. (2010). *Artificial intelligence: a modern approach (3a ed.)*. Upper Saddle River, NJ, EEUU: Prentice Hall.
- Sangiovanni, G., Farè, S., Amigoni, F., y Lavagna, M. (2005, Sep). Multiagent-based scheduling and execution of activities for space systems. En *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space -iSAIRAS-*. Simposio llevado a cabo en Munich, Alemania.

- Schetter, T., Campbell, M., y Surka, D. (2003). Multiple agent-based autonomy for satellite constellations. *Artificial Intelligence*, 145(1), 147-180.
- Schoeberl, M. R. (2002, Jun). The afternoon constellation: a formation of Earth observing systems for the atmosphere and hydrosphere. En *IEEE International Geoscience and Remote Sensing Symposium*, 1, 354-356. IEEE.
- Selva, D., Golkar, A., Korobova, O., Lluch, I., Collopy, P., y de Weck, O. (2017). Distributed earth satellite systems: What is needed to move forward?. *Journal of Aerospace Information Systems*, 14(8), 412-438.
- Sierhuis, M., Clancey, W. J., van Hoof, R. J., Seah, C. H., Scott, M. S., Nado, R. A., ... y Buckley, C. B. (2009, May). NASA's OCA Mirroring System: An application of multiagent systems in Mission Control. En *Proceedings of 8th International Conference on Autonomous Agents and Multiagent Systems -AAMAS-*.
- Skobelev, P. O., Simonova, E. V., Zhilyaev, A. A., y Travin, V. S. (2017). Application of multi-agent technology in the scheduling system of swarm of earth remote sensing satellites. *Procedia Computer Science*, 103, 396-402.
- Sundararajan, A. (2016). *The Sharing Economy: The end of employment and the rise of crowd-based capitalism*. Cambridge, MA, EEUU: MIT Press.
- Sutton, R. y Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA, EEUU: MIT Press.
- Tambe, M., Scerri, P., y Pynadath, D. V. (2002). Adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17(1), 171-228.
- Thanapalan, K. K. T., y Veres, S. M. (2005, Dic). Agent based controller for satellite formation flying. En *2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing* (pp. 385-389). IEEE.
- Truskowski, W., Hinchey, M., Rash, J., y Rouff, C. (2004). NASA's swarm missions: The challenge of building autonomous software. *IT professional*, 6(5), 47-52.
- Turing, A. (1950). Computing machinery and intelligence. *Mind*, 59, 433-460.
- von Maurich, O., y Golkar, A. (2018). Data authentication, integrity and confidentiality mechanisms for federated satellite systems. *Acta Astronautica*, 149, 61-76.
- Weiss, G. (Ed.) (2013). *Multiagent systems* (2da ed.), p. xxxvi. Cambridge, MA, EEUU: MIT Press.
- Wertz, J. R., Everett, D. F., y Puschell, J. J. (Eds.). (2011). *Space mission engineering: the new SMAD*. Hawthorne, CA, EEUU: Microcosm Press.

- Wooldridge, M. (2011). *An introduction to multiagent systems* (2da ed.). Chichester, West Sussex, Inglaterra: Wiley.
- Wooldridge, M., y Jennings, N. R. (1995). Intelligent agents: theory and practice. *The knowledge Engineering Review*, 10(2), 115-152.
- Xu, R., Yuan-Cui, P., Xu, X. F., & Cui, H. T. (2003, Nov). Multi-agent planning system for spacecraft. En *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics*, 4, 1995-1999. IEEE.
- Yliniemi, L., Agogino, A. K., y Tumer, K. (2014). Multirobot coordination for space exploration. *AI Magazine*, 35(4), 61-74.

