Comparison of Multiple Agent-based Organisations for Satellite Constellations (TechSat21)

Thomas Schetter[†], Mark Campbell[†], Derek Surka[‡]

[†]University of Washington
Department of Aeronautics and Astronautics, Box 352400
Seattle, WA 98195-2400
thomas_schetter@hotmail.com, mcamp@aa.washington.edu

[‡]Princeton Satellite Systems
33 Witherspoon St.
Princeton NJ, 08542
dmsurka@psatellite.com

Abstract

Multiple, agent-based satellite systems are envisioned because they are capable of higher performance, lower cost, better fault tolerance, reconfigurability and upgradability. This paper compares several organisational types of agents for autonomous satellite constellations. The focus mission is TechSat21, an Air Force mission to explore the benefits of a distributed approach to satellite systems for space based radar. Required spacecraft functions as software agents and multi-agent based organisations for the TechSat21 mission are described. Agent-based simulations of TechSat21 scenarios show the autonomous operation of three organisations of increasing autonomy.

Introduction

The use of agent-based software architectures represents a new technique in the area of space applications. The individual spacecraft and/or its sub-components are now seen as agents, that is, as individual, independent autonomous entities. This differs from the traditional approach both in the way they act together (i.e. the organisational structure) as well as in the "intelligence" they have (functional distribution). In this discussion, the spacecraft is considered as an agent, referred to a spacecraft-level agent, and the constellation of satellites as the agent-based organisation.

In terms of functional and organisation criteria, there are numerous possibilities for the design of multi-agent (MA) based organisation for applications such as multiple satellite space systems. Parameters such as communication, performance and reliability of an MA-based organisation depend strongly on its design and the task to fulfill. The purpose of this paper is to compare four very distinct approaches to multiple satellite autonomy, each with an increasing level of intelligence. The approach here is to evaluate different MA-based organisations using agent-based and satellite dynamic simulations. The TeamAgent^{TM} multi-agent simulation environment is used in this study as one approach to enabling agent-based multi-satellite systems to fulfill their complex mission objectives. It should be noted that there only a subset of the full satellite subsystems are integrated in this comparison; thus, while particular values for parameters such as communication and computation may be in error, the trends can still be utilized.

The focus mission is TechSat21, an Air Force mission being designed to explore the benefits of a distributed approach to satellites. The initial demonstration will be a space based distributed radar (**RA**dio **D**etection **A**nd **R**anging) (Das *et al.* 1998). The ability to perform a space based radar mission, which historically has required very large, high-power, satellites, is seen as an extreme test of this concept. The concept takes advantage of the DSS by using a sparse aperture array for radar imaging, which allows improved resolution.The satellites will fly in close formation in order to achieve the required performance. For ease and simplicity of the simulation in this work, a constellation of eight satellites was chosen, where four satellites are placed in each ellipse (cluster).

TeamAgent Software Architecture

The TeamAgent toolbox for MATLAB from *Princeton Satellite Systems (PSS)* (PSS 1999) is a simulation environment for multi-agent systems (MAS), especially tailored for the spacecraft domain. It is built upon the ObjectAgent software for single satellites. In TeamAgent, **agents** represent the software and **remote terminals** connect the agents with the hardware.

The basic building blocks within TeamAgent to create agents/remote terminals and pass information are *message centers*, *skills*, *messages* and *tasks*. Message centers (MC) have two primary functions: 1) to register and validate agents and, 2) to pass messages be-

---

tween agents and other message centers.

Skills are the basic building blocks within TeamAgent to create agents and remote terminals. Agents are created in TeamAgent by assigning them a set of skills. Skills are basic software functions pertinent to how the agent performs. These are usually in the form of MATLAB `.m` files. Generally, each skill corresponds to one basic function, has inputs and outputs, and triggers one or more actions. Messages are used to pass information between agents. Tasks are special data structures in TeamAgent that describe (a) particular actions to be taken by the agent and/or (b) messages to be sent to another agent. Messages and tasks for agents and remote terminals are passed identically in TeamAgent via message centers.

Multi-Agent Organisations for Satellites

The AI and MAS community uses many terms and definitions, many of which overlap. This creates a very complex area on which to build. In addition, portions of these technologies are not applicable to multiple satellite systems with mission goals. Therefore, the following section first identifies four levels of spacecraft-level agents, and outlines four organizational architectures of these spacecraft-level agents. Next, required functional agents (lower-level agents) are defined, followed by the implementation of each in TeamAgent/SIMULINK.

Spacecraft-level agents and Organisations

In the following treatment of agent-based organisations, the spacecraft is considered as an agent and the constellation of satellites is seen as the agent-based organisation. The spacecraft is therefore referred to as a **spacecraft-level agent**. In order to narrow the scope of study, the spacecraft-level agents are defined based on the level of capable intelligence. The level of capable intelligence describes the sum of the functionalities that the spacecraft is capable of carrying out. Four levels of capable intelligence I_1-I_4 have been identified, where I_1 denotes the highest level of intelligence and I_4 the lowest level. Figure 1 (top) illustrates the identified spacecraft-level agents.

The spacecraft-level agent I_4 represents the most "unintelligent" agent. It can only receive commands and tasks from other spacecraft-level agents in the organisation or from the ground and execute them. This type of intelligence is similar to what is being flown on most spacecraft today. The spacecraft-level agent I_3 has local planning functionalities on board. This type of intelligence is similar to the Deep Space 1 (DS-1) mission (Muscettola *et al.* 1998). The spacecraft-level agent I_2 adds a capability to interact with other spacecraft-level agents in the organisation. This usually requires that the spacecraft-level agent has at least partial knowledge of the full agent-based organisation, i.e. other spacecraft-level agents. The spacecraft-level agent I_1 represents the most "intelligent" agent. The primary difference between I_1 and the other spacecraft-level agents is that it is capable of monitoring all spacecraft-level agents in the organisation and planning for the organisation as a whole. This requires planning capabilities on the cluster level as well as having full knowledge of all other spacecraft-level agents in the organisation.

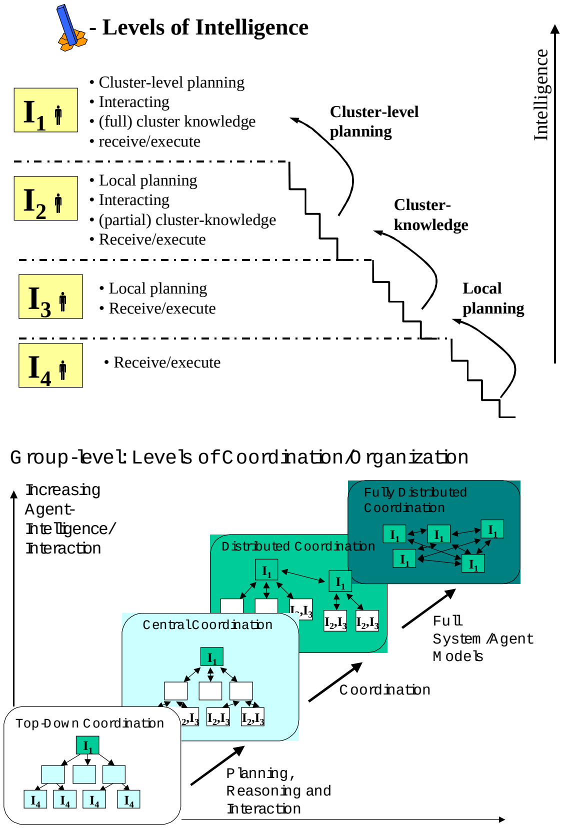


Figure 1: Top: identification of spacecraft-level agents based on levels of capable intelligence I_1-I_4. Bottom: coordination architectures for coordination of multiple spacecraft-level agents.

In order to develop a coherent working community within the cluster such that all of the necessary capabilities can be achieved, the organization must be designed very carefully. Again in order to narrow the scope of the study, four organisational levels are defined.

Figure 1 (bottom) shows a summary of the four possible coordination options mentioned above for a spacecraft-level agent team, as a **function of individual, capable spacecraft-level agent intelligence**. The top-down coordination architecture includes only one single (highly intelligent) spacecraft-level agent I_1 and the other spacecraft are (non-intelligent) I_4 agents. The centralized coordination architecture requires at least local planning and possibly interaction capabilities from each spacecraft. Thus, spacecraft-level agents I_3 or I_2 are required instead. The distributed coordination architecture consists of several parallel hierarchical decision-making structures, each "commanded" by an intelligent spacecraft-level agent I_1. Note that the different spacecraft-level agents I_1 can interact with each other as well as their lower level I_2 or I_3 spacecraft-level agents. In the case of a fully distributed coordination architecture, each spacecraft in the organisation represents a spacecraft-level agent I_1, resulting in a totally "flat" organisation.

Implementation

In order to evaluate the spacecraft-level agents and organisations, lower-level functional agents are defined and implemented in the TeamAgent software, see (Schetter 1999) for more details. Table 1 shows the implemented skills for these agents. Shown also are the AI-based tools used within the skills.

All decision-making skills are implemented using Fuzzy control, (Nguyen *et al.* 1995)). Although the *Cornwell Metric* (Kong *et al.* 1999) is a more accurate approach to cluster positioning, the `PlanReconfigSkill` uses a simplified approach to calculate new optimal cluster positions, namely equal distances between satellites in each elliptical trajectory. A contract net bidding mechanism is used to implement the cluster assignment (`PlanAssignSkill`) and task allocation (`TaskAllocSkill`). The trajectory planner `PlanFFSkill` is implemented with a Linear Program (LP), (Schetter 1999). The `OrbitManSkill` uses a dynamic simulation of the relative spacecraft movements (linearized Hill's equations) in combination with closed loop servo LQR control for orbit maneuvering.

The SIMULINK (Mathworks 1999) environment was chosen to simulate the TechSat21 spacecraft dynamics, while the agents were simulated in TeamAgent. Figure 2 shows the resulting combined dynamic/agent simulation on the mission-level in SIMULINK. Using SIMULINK's capabilites to create hierarchical block structures, the simulation is implemented as a "top-down" approach. From the top "mission level", one can move down through different levels to see more detail. The spacecraft level contains blocks for both hardware components and software agents, represented by special SIMULINK S-functions. These S-functions help build an interface between TeamAgent agent based software and the SIMULINK simulation.

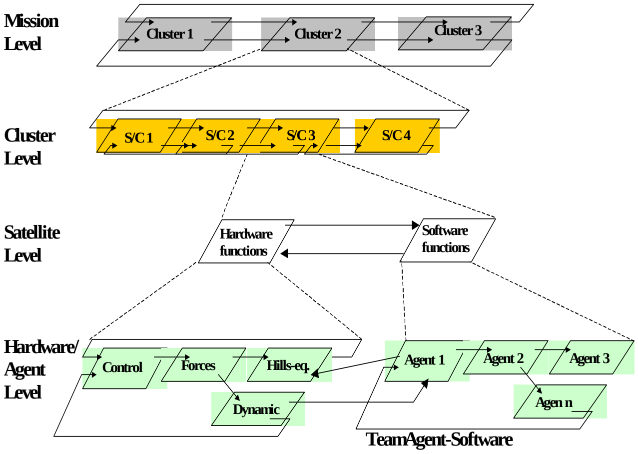


Figure 2: Hierarchical levels of the simulation in SIMULINK

MA Organizations for TechSat21

In order to simplify the comparison and be able to objectively evaluate MA-based organisations for the TechSat21 mission, several comparisons have been identified that can evaluate functional and/or organisational parameters. The organisational comparison presented are listed in the first column of Table 2 along with the evaluation tools, organisational description, and evaluation parameter.

For each comparison studied, two different types of MA-based organisations are compared, and only one organisational component is changed. Figure 3 shows these pictorially, where each block I_i represents an individual spacecraft-level agent. **Comparison 1** evaluates an agent based top down coordination architecture with a traditional architecture using one cluster of four satellites. **Comparison 2** evaluates distributing planning by comparing a top down coordination architecture with a centralised coordination architecture on two clusters of four satellites each. **Comparison 3** evaluates distributing of high-level planning by comparing a centralised coordination architecture with a distributed coordination architecture on two clusters of four satellites each. Note that an additional partially active spacecraft-level agent I_1 is used in the centralised case (which behaves like an I_3 agent) in order to keep the same reliability.

Evaluation parameters can also be numerous in comparing organisations and functional components. For this work, a small subset of important parameters are

Table 1: Examples of implemented software agents in TeamAgent.

Skill	**description**	**Tool**	**usage**			
			I_1	I_2	I_3	I_4
`SensingSkill`	retrieving s/c state	-	x	x	x	x
`CollAvoidSkill`	collision avoidance	Fuzzy Control	x			
`DecMakFailSkill`	health monitoring	Fuzzy Control	x			
`PlanReconfigSkill`	cluster reconfiguration	Cornwell Metric	x			
`PlanAssignSkill`	cluster assignment	Contract net	x			
`PlanFFSkill`	trajectory planning	Linear Program	x	x	x	
`OrbitManSkill`	orbit maneuvering	LQR	x	x	x	x

Table 2: Evaluation metric showing evaluation criteria, tools used, organisational parameters (including the number of spacecraft-level agents) and evaluation parameters.

	Comparison	**Evaluation Tool**	**Organisation**						**Evaluation Parameters**
			Coord.	G	I_1	I_2	I_3	I_4	
1.	Agent-based vs.	SIM #1	T		1			3	c, p, w, r
	traditional	Reliab estimate	$trad$	1				4	
2.	Top down vs.	SIM #2	T		1			6	c, p, w
	Centralised		C		1		6		
3.	Centralised vs.	SIM #3	C		1		5,1_{PA}		c, p, w
	Distributed		D		2		5		

Legend:

P	passive I_1-level agent	PA	partially active I_1-level agent	SIM	Simulation
c	Communication	w	Computation	r	Reliability
p	Performance	$trad$	Traditional coord.	T	Top down coord.
C	Centralised coord.	D	Distributed coord.		

chosen, but could easily be expanded in future studies. Evaluation parameters to be considered are

Communication (C) includes any two of the three sub-parameters communication time (i.e. time required for the communication in seconds) assuming the same data rate, the data rate (i.e. transmitted bytes per second), the total communication effort (i.e. total number of bytes to be transmitted). The required power for the communication is also used, which is related to the above parameters through

$$P = \frac{16 K T_s d S^2 c^2}{L_a L_l D_t^2 D_r^2 \eta^2 f^2} \tag{1}$$

where L_a is the transmission path loss, L_l is the line loss, K is the Boltzmann's constant, T_s is the systems noise temperature, c the velocity of light, $D_{t,r}$ is the diameter of the transmitter or receiver antenna, d is the data rate, S the distance between transmitter and receiver, η the efficiency (factor between 0 and 1) and f the carrier frequency. Table 3 shows the specific parameters that were chosen.

Computation (W) is the current CPU workload and the total workload, or a summation of the CPU usage over all time. The current workload can be thought of as the percentage of the maximum computational rate (comp/sec), or percentage of CPU time dedicated to tasks. In the simulation, the current workload of an individual spacecraft-level agent I_i for the time interval t_s is calculated by

$$\frac{\mathrm{CPU(I_i)}}{t_s} \cdot 100\%, \tag{2}$$

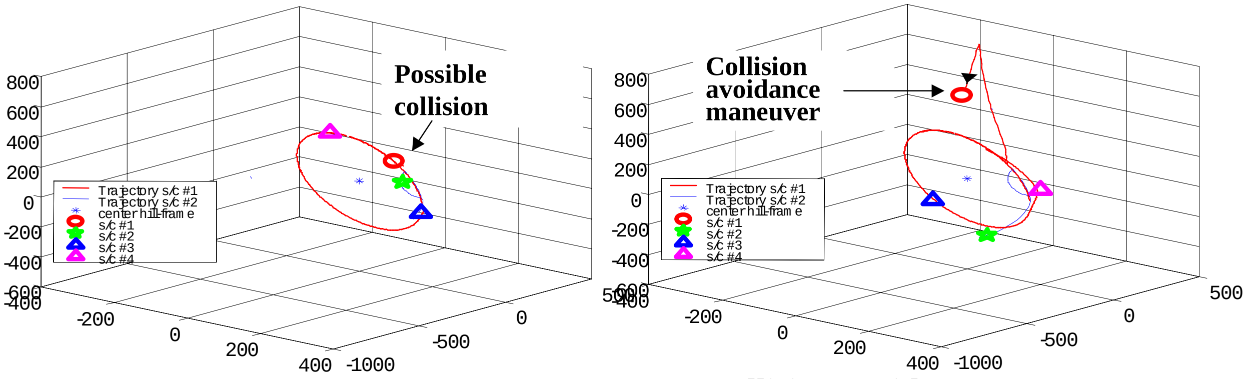
where $\mathrm{CPU}(I_i)$ denotes the CPU usage of the spacecraft-level agent I_i for the time interval t_s. The current workload of the entire organisation for the time interval t_s is found by taking the series or parallel CPU usage into account. Note that the required computation for the ground station is not factored into the parameter.

Performance (P) is the on-board fuel consumption and time required to perform the simulation. Because the time for communication has the greatest impact on the execution time, only the communication time is factored into this parameter.

Reliability (R) is the reliability of the satellites and software architecture on the mission level. Note that the software reliability is assumed to be 100%.

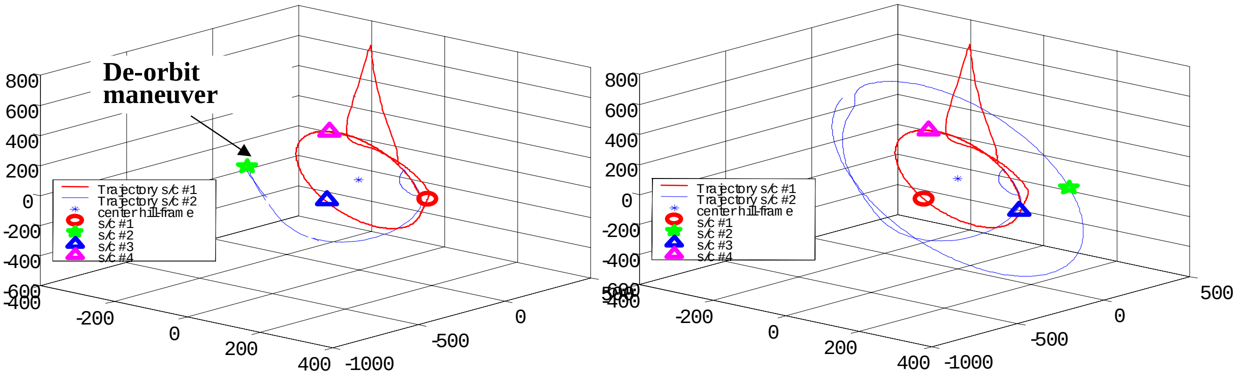
Table 3: Selected communication link parameter for the TechSat21 mission.

Parameter	downlink (d/l)	crosslink(c/l)
Carrier frequency f	2 GHz	60 GHz
Transmission path loss L_a	0.3	1
Transmitter line loss L_l	0.1	0.1
System noise T_s	550 K	1800
Efficiency η	1	1



A) Close to collision

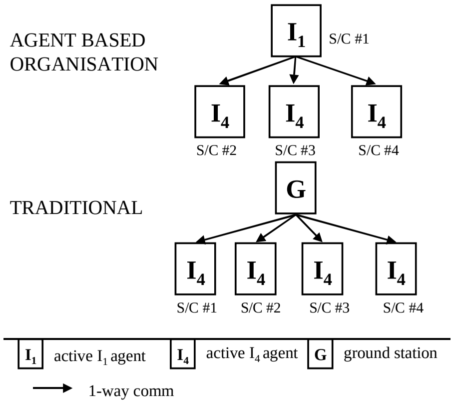
B) Collision avoidance



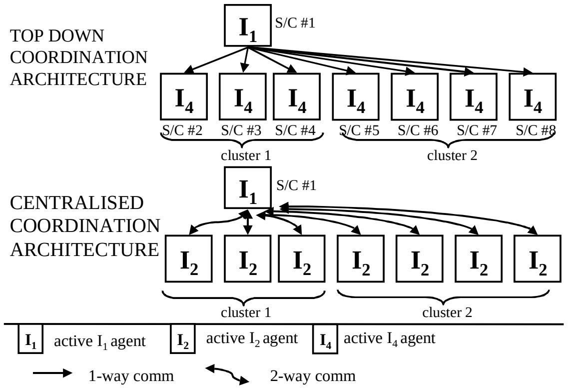
C) De-orbit S/C #2

D) Final Situation

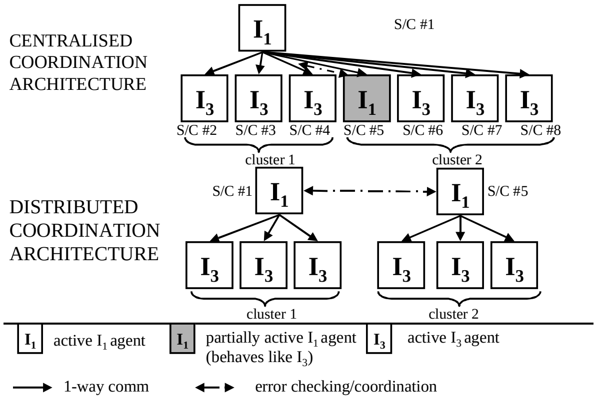
Figure 4: A 3D animation of the simulation scenario 1 (collision avoidance with a cluster reconfiguration). Shown are four snapshots of the scenario for the agent based organisation: A) Close to collision, B) Collision avoidance, C) De-orbit of S/C #2, and D) Final situation.



Comparison 1.



Comparison 2.



Comparison 3.

Figure 3: Three comparisons used to evaluate multi-agent systems for autonomous spacecraft clusters.

Comparison 1

Simulation Simulation scenario 1 simulates the case when one satellite loses GPS contact and begins to move towards a second satellite. If no maneuvers occur, the satellites will collide within 0.2 orbits, or 20 minutes. The necessary steps that the cluster must perform are detecting the collision within a threshold of 50 m, and avoiding it with a maneuver. Figure 4 A,B show the scenario and the required maneuver. For the traditional approach, the decision-making, initialization of the collision avoidance maneuver and cluster reconfiguration must be performed from the ground station since there is no interaction between the satellites. For non geosynchronous orbits, the spacecraft only have contact with earth ground stations in its field of view which, for this simulation, is assumed to be 10 min. The number of contacts per orbit obviously depends on the number of available ground stations.

Figure 4 shows the simulation scenario for the agent-based approach with the different spacecraft marked according to the legend shown. Shown are cases A) just before the collision avoidance maneuver (step 1); B) the collision avoidance maneuver (step 2); C) the de-orbit operation of spacecraft #2 (step 3); and D) the final situation after the cluster reconfiguration and de-orbit operation of the failed spacecraft #2. Shown also are the trajectories for spacecraft #1 and #2. As can be seen, the resulting bang-bang control for the collision avoidance maneuver (B) results in a large trajectory change for spacecraft #1 (represented by the circle symbol).

For the traditional approach, the satellites will obviously collide if there is no ground station contact during the 0.2 orbits (20 min) after failure. If the failure occurs during ground station contact, then the maneuver and de-orbit look identical to the results in Figure 4, but the total time may differ.

Results The agent based approach is much better in terms of power and reliability (for a fixed cost), but not computation. The increased communication distance to the earth for the traditional approach results in a significantly higher power requirement. Using Equation (1) and Table 3, the normalized, relative increase in the required power P_{REL} for the d/l (traditional approach) compared to c/l (agent-based approach) can be calculated as

$$P_{REL} = \frac{P_{TRAD}}{P_{AGENT}} = 0.0133 \cdot \frac{S_{d/l}^2}{S_{c/l}^2} = 0.0133 \cdot \frac{700^2}{0.5^2} \approx 2.6 \cdot 10^4 \quad (3)$$

where $S_{d/l}$ is the distance for communication with the earth (i.e. 700 km altitude), $S_{c/l}$ the distance for inter satellite communication (i.e. 500 m). This assumes a constant data rate of $d_A = d_T = 4$ kbps.

Figure 5 on the top shows a plot of the relative distance between spacecraft #1 and spacecraft #2 for both the agent based and traditional simulations. As can be seen from the plot, the time from the failure occurrence to the point when spacecraft #2 and spacecraft #1 are within 50 m is less than $\frac{1}{5}$ of an orbit (20 min). This 50 m threshold is the critical distance for activating the collision avoidance maneuver. If the above events occur during the time interval when the spacecraft are out of contact with a ground station, the traditional approach would result in a collision between spacecraft #1 and #2.

For the traditional approach, the number of ground stations determines the probability for detecting collisions between the spacecraft and thus, is directly related to the reliability. Figure 5 on the bottom graphs the reliability for the case of the agent-based approach and the traditional approach, where it was assumed that the ground station can be symmetrically distributed. Usually this assumption cannot be realized to due political and geographical constraints, or is very expensive (Wertz and Larson 1992). The cost is directly proportional to the number of ground stations used. If the assumptions about reliability are in error, as shown by the large uncertainty bounds, the overwhelming evidence of Figure 5 is that the agent based approach increases reliability and decreases costs.

The simulation shows that the agent-based organisation has a disadvantage in terms of computation because tasks usually done by the ground station now must be performed on board the spacecraft.

Assuming a constant data rate, the performance in terms of communication time is the same for the two cases. However, the generation of new command sequences for collision avoidance and cluster reconfiguration maneuver most likely will take more time on the ground (traditional approach) than on board the spacecraft (agent based approach).

Table 4 shows the results of comparison 1 (agent-based vs. traditional approach) for the simulation scenario 1.

Comparison 2

Simulation Simulation scenario #2 simulates the case of a cluster reconfiguration where spacecraft #3 in cluster 1 and spacecraft #7 in cluster 2 are de-orbited and the two clusters are reconfigured. Figure 6 shows the simulation for cluster 1 (cluster 2 is similar). Shown are the nominal working of the cluster in A, and the situation after reconfiguration of cluster 1 in B: spacecraft #3 is de-orbited and spacecraft #1,2 and 4 in cluster 1 have maneuvered to new positions within the cluster

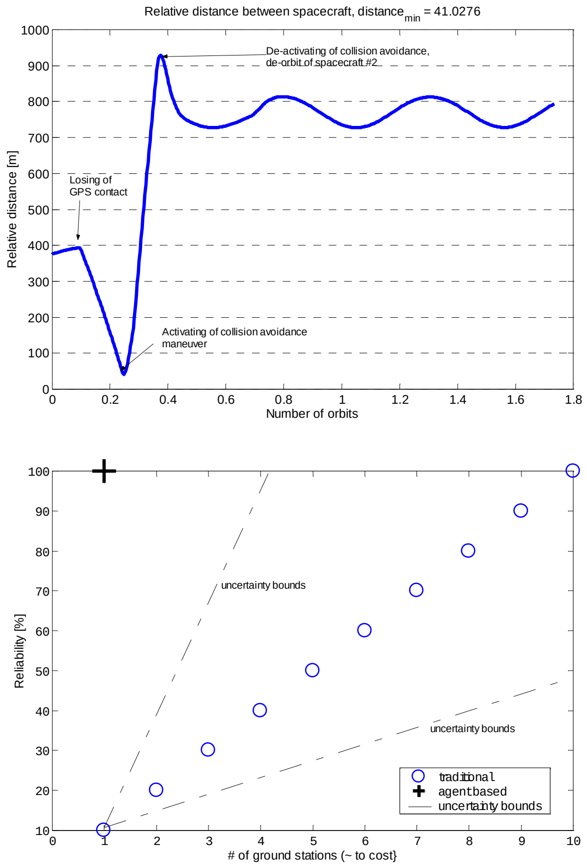
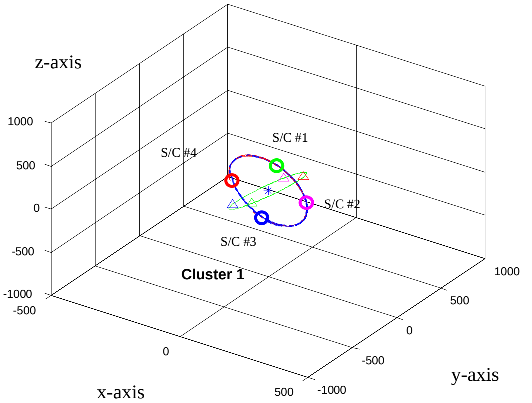
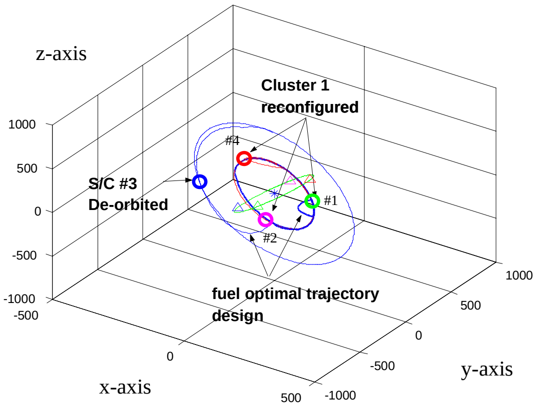


Figure 5: Top: Relative distance between spacecraft #1 and #2 for the simulation scenario #1. Bottom: Reliability for agent based ('+') and traditional approach ('o') for the simulation scenario 1 as a function of number of ground stations and cost.

Results The total communication effort as well as the communication between the individual spacecraft-level agents I_1 and the lower level agents, is decreased due to onboard autonomy of the lower level agents. This results in a decrease in the data rate, power and amount of data to be transmitted. Figure 7 shows the average data rate over small time intervals for the communication link between the active spacecraft-level agent I_1 and the lower level spacecraft agents during the simulation scenario 2. As can be seen, the mean data rate is reduced by a factor four for the centralised coordination (84 bps) compared to the top down coordination (322 bps) due to a significantly smaller amount of total transmitted bytes for the centralised case (0.08 Mbytes after 1.8 orbits) compared to the top down case (0.33 Mbytes after 1.8 orbits).



A) Situation before cluster reconfiguration



B) De-orbit S/C #3 and cluster reconfiguration

Figure 6: A 3D animation of the simulation scenario 2. A) Situation before cluster reconfiguration and B) De-orbit of spacecraft #3 and reconfiguration of spacecraft #1,2 and 4 in cluster 1. Note that a similar maneuver is performed for cluster 2.

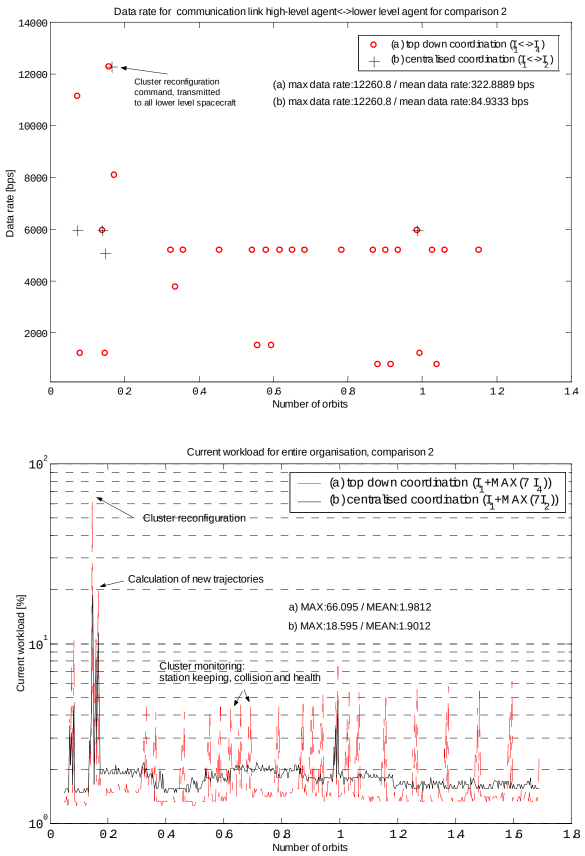


Figure 7: Top: data rate during simulation scenario 2 between the active spacecraft-level agent I_1 and lower-level spacecraft agents. Bottom: Current workload for top down coordination and (b) centralised coordination for simulation scenario 2.

Performing lower level decision-making and planning within all spacecraft-level agents (in parallel) results in a better distribution of the computational workload across the cluster, as well as a savings in time. The downside of the centralised coordination architecture is, however, that the total computational workload increases. Figure 7 on the bottom shows the current workload for simulation scenario 2. The peak loads corresponding to cluster reconfiguration events are decreased for the centralized case (18.5% max) compared to the top down case (66.0% max). The mean value is slightly increased for the centralised coordination (1.98%) compared to the top down coordination (1.9%).

Assuming that the data rate for the c/l in both cases are sized based on the communication requirement (mean data rate) for the centralised case, or 84 bps, the relative performance disadvantage in terms of communication time $\triangle t$ for the top down case is calculated as $\triangle t = \frac{(0.33-0.08)\mathrm{Mbytes}}{84bps} \approx 400$ min.

Table 4 shows the results of comparison 2 (top down coordination vs. centralised coordination architecture) for the simulation scenario 2.

Comparison 3

Simulation Simulation scenario 3 is identical with simulation scenario 2.

Results Figure 8 on the top shows the data rate during the simulation scenario 3 for the communication link between one active spacecraft-level agent I_1 and the lower level spacecraft agents. Both the maximum and mean (average) data rate is decreased nearly by a factor 1/2 for the distributed case (max. 5.9 kbps and mean 169 bps) compared to the centralised case (max.

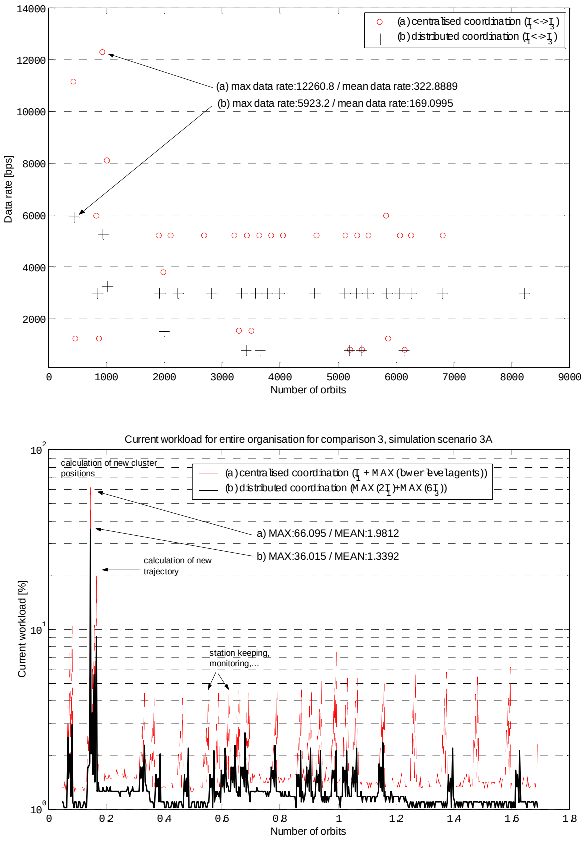


Figure 8: Top: data rate during simulation 3 between active spacecraft-level agent I_1 and lower-level spacecraft agents. Bottom: current computational workload for the entire organisation during simulation scenario 3.

12 kps and mean 333 bps) due to a significantly smaller amount of total transmitted bytes for the distributed case (0.15 Mbytes after 1.8 orbits) compared to the centralised case (0.30 Mbytes after 1.8 orbits). The total communication effort for the two cases is nearly equal.

Figure 8 shows on the bottom the current computational workload for the entire organisation for the centralised and distributed cases. Both the peak loads and the mean values of the computational workload are significantly decreased for the distributed case (max. 36.0% and mean 1.3%) compared to the centralised case (max. 66% and mean 1.9%). Notice also that the mean workload of the I_1 spacecraft-level agents is decreased for the distributed case (0.95%) compared to the centralised case (1.25%). Thus, the distributed case results in a decreased workload requirement for each high-level agent I_1 because the high-level planning and decision-making task are done in parallel by two active spacecraft-level agents I_1 instead of one.

Assuming that the data rate for the c/l in both cases are sized based on the communication requirement (mean data rate) for the centralised case , 169 bps, the relative difference in the communication time $\triangle t$ is calculated as $\triangle t = \frac{(300-150)\mathrm{kbytes}}{169 bps} \approx 118$ min.

Table 4 shows the results of comparison 3 (distributed vs. centralised coordination architecture).

Conclusions and Future Work

The use of multi-agent systems for autonomous satellite clusters has been explored. Using distinct functions for satellite clusters such as linear programming for minimum fuel maneuvers and fuzzy logic for decision making, four types of organizations were simulated for TechSat21, a future distributed space based radar mission. The four organizational types each increase in intelligence and autonomy: Traditional (with a ground station and no intelligence), Top-down, Centralized, and Distributed. Table 4 shows the summary of the numerical results.

Although only several simulations were compared, the following conclusions can be made. Autonomous clusters (using multi-agent systems) are enabling for LEO missions with satellite in close proximity with quick reaction times. Traditional approaches cannot achieve good reliability at a good cost. Each level of autonomy adds more computational workload; thus, the tradeoff is usually added reliability, performance for increased computational effort. Computational effort, however, is usually quite cheap, but adds more uncertainty as well. Distributed systems decrease the maximum workload, but increase the total workload and communications slightly. Reliability is also usually increased. Although much more comparison work needs to be done, a good organization appears to be partially distributed with a small percentage of highly intelligent satellites for redundancy. Future work will focus on exactly how to define this organization, and how to implement it on small lab testbeds and TechSat21.

Acknowledgments

This work is supported under an United States Air Force SBIR contract with Princeton Satellite Systems, Contract Number F29601-99-C-0098.

Bibliography

Hill, G. W. 1978. Researches in the lunar theory. *American Journal of Mathematics, Vol 1, No. 1*, pages p. 5–26.

Table 4: Numerical results of comparison 1 -3.

		Computation (workload organisation)			**Communication**			**Perfor-mance**		**Relia-bility**
		current (max) [%]	current (mean) [%]	total^0 [%]	data rate [kbps]	total bytes [Mbytes]	power ratio [-]	fuel [m/s]	time [min]	succes. probab. [%]
Cp. 1	trad	1.4	0.47	1.6			$2.6 \cdot 10^4$		*	10^+
	agent	33.3	1.69	2.38			1		*	100
Cp. 2	Top down	66.0	1.98	4.76	322	0.60	4		+400	
	Centralised	18.5	1.90	5.77	84	0.14	1			
Cp. 3	Centralised	66.0	1.98	3.51	333	0.60	2		+118	
	Distributed	36	1.33	4.04	169	0.6	1			

Legend:

trad	Traditional approach
agent	Agent-based approach
0	After 1.8 orbits
+	Value depends on number of ground station. Shown value is for 1 ground station.
*	No numeric value can be given, but most likely agent based approach has advantage.

PSS 1999. Princeton Satellite Systems Inc. web page. http://www.psatellite.com.

Kong, E. M.; Tollefson, M. V.; Skinner, J. M.; Rosenstock, J. C. 1999. Techsat 21 Cluster Design Using AI Approaches and the Cornwell Metric. In Proceedings of the AIAA Space Technology Conference, Albuquerque NM. AIAA Paper AIAA-99-4635.

Campbell, M. E.; Schetter, T. 2000. Formation Flying Mission for the UW Dawgstar Satellite. In Proceedings of the IEEE Aerospace Conference, Big Sky Montana.

Schetter, T. 1999. Autonomous Control for Multiple Satellite Clusters, Master's Thesis, University of Washington, Department of Aeronautics and Astronautics.

Das, A.; Cobb, R.; Stallard, M. 1998. A Revolutionary Concept in Distributed Space Based Sensing. In Proceedings of the AIAA Defense and Civil Space Programs Conference & Exhibit, pages p. 1–6, Huntsville, AL.

Surka, D. M. 1999. The ObjectAgent System Architecture, 3-Day Workshop at the Air Force Research Lab, Albuquerque NM, July, 1999.

The Mathworks Inc. 1997. *SIMULINK User's Guide.* The Mathworks, Inc., 24 Prime Park Way, Natick, MA 01760-1500.

Nguyen, H. T.; Sugeno, M.; Tong, R.; and Yager, R. R. 1995. *Theoretical Aspects of Fuzzy Control.* John Wiley Sons, Inc.

Wertz, J. R.; Larson, W. J. 1992. *Space Mission Analysis and Design.* Kluwer Academic Publishers.

Muscettola, N.; Nayak, P. P.; Pell, B.; Williams, B. C. 1998 Remote Agent: To Boldly Go Where No AI System Has Hone Before. In Proceedings of the 15th International Joint Conference on Artificial Intelligence NAGOYA, Aichi, Japan.