

Management of Celestial Systems under Spatial Grasp Model

Peter Simon Sapaty¹

¹ Institute of Mathematical Machines and Systems, National Academy of Sciences, Russia

Correspondence: Peter Simon Sapaty, Institute of Mathematical Machines and Systems, National Academy of Sciences, Russia. E-mail: peter.sapaty@gmail.com

Received: November 2, 2021 Accepted: December 29, 2021 Online Published: December 30, 2021

doi:10.5539/nct.v6n2p10

URL: <https://doi.org/10.5539/nct.v6n2p10>

Abstract

Many governmental agencies and private companies of different countries are now rushing into space around Earth in hope to provide smart communication, industrial, security, and defense solutions. This often involves massive launches of cheap small satellites which are also contributing to the growth of space debris. The current paper discusses how the developed high-level system philosophy and model can effectively organize distributed space-based systems on different stages of their development and growth. The briefed Spatial Grasp Technology, based on parallel pattern-matching of distributed environments with high-level recursive mobile code, can effectively provide any networking protocols and important applications of large satellite constellations, especially those on Low Earth Orbits. The paper contains examples of technology-based solutions for establishing basic communications between satellites, starting from their initial, often chaotic, launches and distributing and collecting data in the growing constellations with even unstable and rapidly changing connections between satellites. It describes how to organize and register networking topologies in case of predictable distances between satellites, and how the fixed networking structures can help in solving complex problems. The latter including those related to the new Space Development Agency multiple-satellite defense-oriented architecture and allowing for effective integration of its continuous earth custody observation and cooperative missile tracking and elimination layers, based on self-spreading mobile intelligence. Earlier versions of the technology, described in many papers, six books including, were prototyped and used in different countries, with the current one quickly implementable too, even in university-based environments.

Keywords: space conquest; satellite constellations; Spatial Grasp Technology; communication protocols; new SDA architecture; transport, custody, and tracking layers

1. Introduction

Different countries are now chaotically rushing into space, in a sort of Klondike-like adventures (Klondike Gold Rush, n.d.; What Was the Klondike Gold Rush?, n.d.) (apology for this funny but friendly comparison), in hope to provide quick and smart communication, climate, industrial, security, and defense solutions (Mohanta, 2021; Space Debris, NASA Headquarters Library, 2021; United Nations Register of Objects Launched into Outer Space, The United Nations Office for Outer Space Affairs; Martin, 2014; Bockel, 2018; Curzi, Modenini, & Tortora, 2020; Venkatesan et al., 2020; Skibba, 2020; Minet, 2020; Siegel, 2021; Jones, 2021; Reiland et al., 2021; Foust, 2016; Space Development Agency Next-Generation Space Architecture; 2019; Messier, 2019; Insinna, 2019; Strout, 2021). This often involves massive launches of cheap and unsafe small satellites in Low Earth Orbits, which are also contributing to the growth of space debris endangering any future space research and conquest. There were 7,389 individual satellites in Space at the end of April, 2021 -- an increase of 27.97% compared to 2020 (Mohanta, 2021). In total, 11,139 satellites have been launched, out of which only these 7,389 are in the Space, while the rest have either been burnt up in the atmosphere or have returned to Earth in the form of debris. According to NASA (Space Debris, NASA Headquarters Library, 2021), there are millions of pieces of junk flying in LEO, which comprises of Space craft, tiny flecks of paint from Space craft, parts of rockets and satellites that are either dead or lost, including objects that are results of explosions in the Space.

But this massive and often competitive penetration into space in an attempt to gain something very quickly, possible cheaply, and ahead of others, is not based on any global culture and planning, say, UN supported. The author of this paper witnessed a similar situation more than half a century ago, but on Earth, having been actively engaged in creation of the first citywide heterogeneous computer networks (actually inventing and writing himself so far unknown communication protocols) combining very different and even incompatible machines

(Bondarenko et al., 1970; Bondarenko et al., 1972; Sapaty, 1973), well before the internet. Having offered at that time how to unite such distributed equipment into a highly capable complexes using a sort of parallel wavelike (even virus-like) model, the author is eager now to use similar but much more advanced and successfully tested in different countries the Spatial Grasp model and resulting networking Technology, in hope to bring certain order into this enormously and chaotically growing multiple satellite and junk mess around the Earth, with its symbolic picture shown in Figure 1.

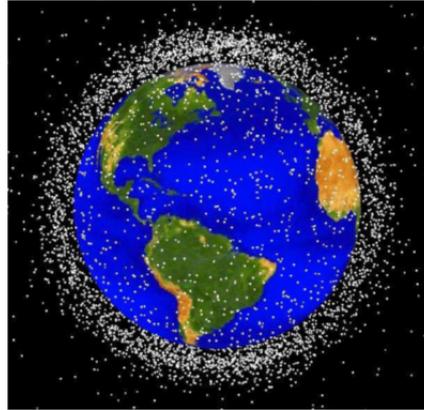


Figure 1. Rapidly growing multiple objects around the Earth

The rest of this paper is organized as follows. Section 2 briefs basics of Spatial Grasp Technology (SGT), including its Spatial Grasp Language (SGL), description of worlds SGT operates with, also organization of the SGL networked interpreter. Section 3 shows how to deal with highly dynamic constellation topologies under SGT, which includes reaching proper satellite nodes, delivery of a package to proper nodes, reaching proper destinations with the return and collection of certain items from them, also introducing special measures for dealing with high unpredictability of topology dynamics. Section 4 is working with stable constellation topologies explaining advantages of them, and shows how to create and fix communication topology in case of stable distances between satellites. In Section 5, examples of solving distributed problems under stable constellation topologies, including collection and return of remote data, and also how to integrate in SGL of cooperative work of the custody and tracking layers of SDA Next-Generation Space Architecture. Such symbiosis allows for a constant overseeing of dangerous objects on Earth with immediate tracing and elimination of the launched hypersonic missiles. Section 6 concludes the paper with mentioning plans for a continuing research and development in this area, where the new book is currently being completed, also possibility of quick implementation of the described latest SGT version for advanced constellations management.

2. Spatial Grasp Technology (SGT)

SGT operates by spatial scenarios self-spreading in physical and virtual worlds while creating, matching, transforming and managing them. The interpreted scenario text is not staying in any fixed point or points in space but rather spreads itself while carrying further its remainder and omitting the utilized parts.

2.1 Spatial Grasp Language

All SGT functionality can be expressed and obtained by the recursive high level Spatial Grasp Language (SGL) in which all spatial scenarios are represented, with its top level syntax following.

```

grasp    →  constant | variable | rule [({ grasp })]
constant  →  information | matter | custom | special | grasp
variable  →  global | heritable | frontal | nodal | environmental
rule      →  type | usage | movement | creation | echoing |
            verification | assignment | advancement | branching |
            transference | exchange | timing | qualifying | grasp

```

2.2 Worlds SGT operates with

SGT allows us to directly operate with the following world representations: *Physical World*, considered as

continuous and infinite; *Virtual World*, which is discrete and consists of nodes and semantic links between them; and *Executive world* consisting of active “doers” with communication possibilities between them. Different kinds of combination of these worlds are also possible within the same formalism, like: *Virtual-Physical World*, *Virtual-Execution World*, *Execution-Physical World*, and also *Virtual-Execution-Physical World* combining all features of the previous cases.

2.3 SGL networked interpreter

Communicating Interpreters of SGL can be in arbitrary number of copies, say, up to millions and billions, which can be effectively integrated with any existing systems and communications, and their dynamic networks can represent powerful spatial engines capable of solving any problems in terrestrial and celestial environments. Such collective engines can simultaneously execute many cooperative or competitive tasks without any central resources or control. The SGL interpreter main components are shown in Figure 2.

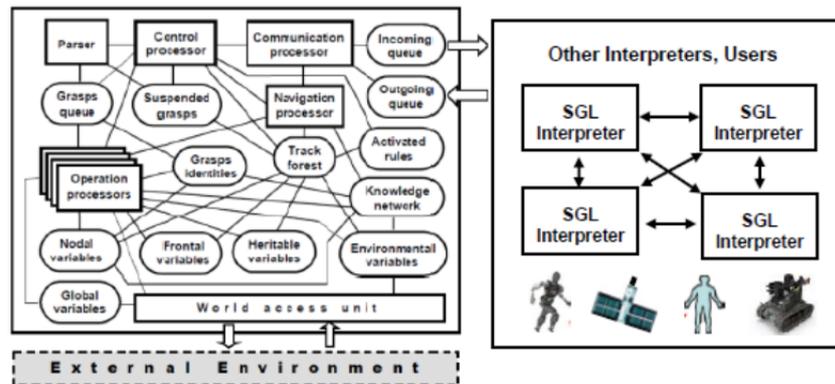


Figure 2. Main components of SGL interpreter and its networked organization

As both backbone and nerve system of the distributed interpreter, its self-optimizing *spatial track system* provides hierarchical command and control as well as remote data and code access. It also supports spatial variables and merges distributed control states for decisions at different organizational levels. The track infrastructure is automatically distributed between active components (humans, robots, computers, smartphones, satellites, etc.) during scenario self-spreading in distributed environments.

Additional data on SGT and SGL can be found in existing publications, including (Sapaty, n.d.; 1999; 2005; 2017; 2018; 2019; 2020; 2021a; 2021b; 2021c; 2021d; 2021e; 2021f; 2021g; 2021h), also more in google.com by “spatial grasp” and amazon.com in books by “sapaty peter”.

3. Dealing with Highly Dynamic Constellation Topologies under SGT

In general, constellations may consist of satellites on different orbits, which may be changing in time their distances to each other, so having no stable network topologies, as symbolically shown in Figure 3. But we will assume here that there are enough of them around the earth, and at each moment of time any satellite can directly communicate with other satellites (at least one of them), which may be very different from the previous communications. So this in principle could enable to propagate through the whole constellation regardless of which satellites are now visible from each other, and also from a ground station.

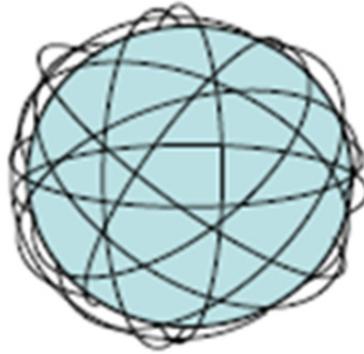


Figure 3. An example of a multiple and complex satellite constellation orbits

We will consider some basic operations on such dynamic networks under SGT, expressed in SGL.

3.1 Reaching Proper Satellite Nodes

Reaching particular satellite node *a* from a ground station (see Figure 4) with a number of such nodes of interest) initially via an arbitrary satellite visible from it at this moment of time, can be accomplished as follows, with subsequent propagation via other satellites (if this starting node is not a itself). The satellites of interest are supposed to be of certain *Type*, and direct communication between them can be possible by some *Depth* threshold distance. The reached satellite is supposed to acknowledge the fact of being reached by a symbolic *output* operation, after which propagation to other satellites will be canceled by *done* statement. The operation *hopfirst* by reaching any new satellite node properly marks it, blocking a repeated entry and therefore any navigation cycling, with could become endless otherwise.

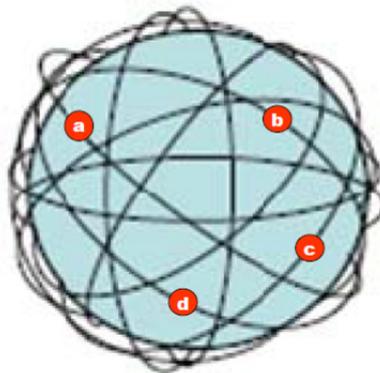


Figure 4. Certain nodes to be reaches in complex satellite constellation

```
frontal (Depth = ..., Type = ...);
hopfirst (any, Type);
repeat (
  if (NAME == a, done (output (ok)));
  hopfirst (Depth, Type))
```

Reaching particular satellite node *a* from another satellite node after initially staying in it, without involvement of ground stations, can be a bit simpler, as follows.

```
frontal (Depth = ..., Type = ...);
repeat (
```

```

if(NAME == a, done(output(ok)));
hopfirst(Depth, Type)

```

Reaching particular group of nodes *Dest* from another satellite node after initially staying in it, where the found nodes may also serve as transits to the other *Dest* nodes, can be as:

```

frontal(Depth = ..., Type = ..., Dest = (a, b, c, d));
repeat(
  if(belong(NAME, Dest), output(ok));
  hopfirst(Depth, Type)
)

```

This can also be accomplished from a ground station getting accidental access to any satellite node, similar to the first case above.

```

frontal(Depth = ..., Type = ..., Dest = (a, b, c, d));
hopfirst(any, Type);
repeat(
  if(belong(NAME, Dest), output(ok));
  hopfirst(Depth, Type)
)

```

We may also use spatial recursion instead of spatial cycling for the previous case, when navigating and covering the system in parallel too, as follows.

```

frontal(
  Depth = ..., Type = ..., Dest = (a, b, c, d),
  Cover = {if(belong(NAME, Dest), output(ok));
           hopfirst(Depth, type); run(Cover)});
hopfirst(any, Type); run(Cover)

```

3.2 Package Delivery to All Proper Destinations

Reaching the set of particular nodes with delivering a package into them all from the ground station, using spatial cycling and taking for simplicity that this package will be attached to their contents (the latter expressed by standard nodal environmental variable *CONTENT*), may be as;

```

frontal(Depth = ..., Type = ..., Pack = ...,
        Dest = (a, b, c, d));
hopfirst(any, Type);
repeat(
  if(belong(NAME, Dest), append(Pack, CONTENT));
  hopfirst(Depth, Type)
)

```

The same by the result, but with the use of spatial recursion:

```

frontal(
  Depth = ..., Type = ..., Dest = (a, b, c, d), Pack = ...;

```

```

    Deliver = {if(belong(NAME, Dest), append(Pack, CONTENT));
              hopfirst(Depth, type); run(Deliver)}};
hopfirst(any, Type); run(Deliver)

```

For the both cases above, instead of CONTENT any other variables could be used in the destination nodes. Also, the delivery may be initially launched from any satellite node rather than ground station.

3.3 Moving to Proper Destinations with Returning and Collecting Packs from Them All

We will consider first picking up a concrete Item in all destinations after reaching them (supposedly from their CONTENT), and then their return, collection, and output from the casual starting satellite node to a ground station. From the very beginning, mobile environmental variable IDENTITY has default value like the starting node name, address (if has it), or just randomly generated value (with which all subsequently reached nodes will be marked by hopfirst, unless this value is changed explicitly, as in the following example for the backward propagations).

```

frontal(Depth = ..., Type = ..., Item = ..., Pack, Start,
        Dest = (a, b, c, d));
hopfirst(any, Type); nodal(Summary), Start = NAME);
sequence(
  repeat(
    if(belong(NAME, Dest),
      free(IDENTITY = NAME;
          Pack = element(CONTENT, Item);
          repeat(if(NAME == Start,
                  blind(append(Summary, Pack)));
                  hopfirst(Depth, Type)))));
    hopfirst(Depth, Type)),
  output(Summary))

```

By output, this packs collection may be delivered to the ground station. The collected packs will also remain in the variable Summary in the starting satellite node. The above scenario combines forward and backward simultaneous spatial processes, where the forward process spreads in parallel from the starting node to reach the needed destinations, and the backward ones immediately start in all reached nodes to deliver their items to the starting node, without waiting the completion of forward process.

Another solution may be with the use of spatial coverage recursion, as follows (for reaching the needed nodes only, but not for returning results which remains spatially repetitive).

```

frontal(
  Depth = ..., Type = ..., Item = ..., Pack, Start, Dest = (a, b, c, d),
  Cover = {(belong(NAME, Dest); IDENTITY = NAME;
            Pack = element(CONTENT, Item);
            repeat(if(NAME == Start, blind(append(Summary, Pack)));
                  hopfirst(Depth, Type))),
            (hopfirst(Depth, Type); apply(Cover))});
hopfirst(any, Type); nodal(Summary); Start = NAME;
sequence(apply(Cover), output(Summary))

```

3.4 Providing Additional Measures for Dealing with Unpredictable Topology Dynamics

We will consider here the possibility of reaching particular nodes from a ground station by allowing nodes to be entered more than once, assuming that with the time passed other nodes may become directly accessible from the same nodes. The additional parameter `Revisit` for rule `hopfirst` shows how many times the nodes can be reentered (i.e. zero without it, as before). The sought and already reached destinations may be ignored when tried to be reentered, as representing the results already reported, for which the mark `Visited` is left in them.

```
frontal(Depth = ..., Type = ..., Dest = (a, b, c, d), Revisit = 3);
nodal(Visited);
hopfirst(Revisit)(any, Type);
repeat(
  if(belong(NAME, Dest),
    done(if(Visited == nil, (Visited = 1; output(ok))));
  hopfirst(Revisit)(Depth, Type))
```

This `hopfirst` option is also using the quality of `hopforth` rule not allowing to reenter the nodes just visited before them. The discussed nodes revisiting variant can also be used in all previous scenario examples for the cases with very high dynamics and unpredictability of spatial distribution of satellite constellations.

4. Working with Stable Constellation Topologies

With established and registered stable links between nodes which are all moving in similar orbits, as in Figure 5, we can describe and solve various problems in satellites constellations much easier and quicker.

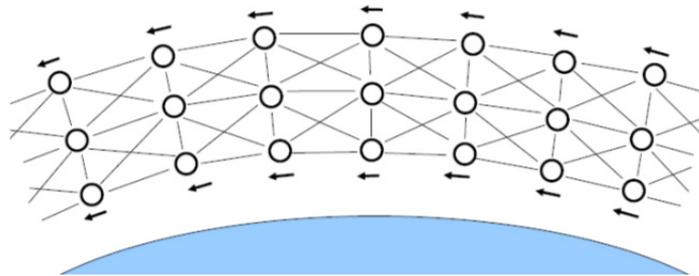


Figure 5. Example of a stable constellation network

4.1 Creating Stable Constellation Network Topology and Its Advantages

Starting from a ground station and linking certain `Type` nodes close enough in space, we may obtain a stable and useful network topology (using `Depth` threshold distance between satellites reflecting their capability of direct communication and official linking with each other). Such network topology can be created in parallel with the constellation navigation, which can also be done in parallel itself (all established links between neighboring satellites are just named as `link`), as follows.

```
frontal(Depth = ..., Type = ...);
hopfirst(any, Type);
repeat(
  free(hop_nodes(Depth, Type); if(no_links(BACK), linkup(link, BACK)));
  hopfirst(Depth, Type))
```

Getting stable topology, we can use the full power of SGT and its basic language SGL for solving any problems on distributed satellite networks, with numerous publications already published in this area, including (Sapaty, n.d.; 1999; 2005; 2017; 2018; 2019; 2020; 2021a; 2021b; 2021c; 2021d; 2021e; 2021f; 2021g; 2021h). Only few of them: finding shortest paths, routing tables, weak and strong components like articulation points and cliques, any

particular structures and components (say, represented as spatial images or patterns), and so on. Which may be very useful for large constellations management -- from creation of celestial version of internet -- to weather prediction -- to global industry -- to security and defense.

For example, an effective and highly parallel SGL scenario finding shortest path trees and full routing tables (as in (Sapaty, 2018; 2019; 2021a), with an example of a network and related routing tables for all its nodes shown in Figure 6. Where the *Dest* vector in each node collects names of all other nodes (as final destinations) and the *Next* vector – the next nodes through which these destinations can be reached via the shortest path to them.

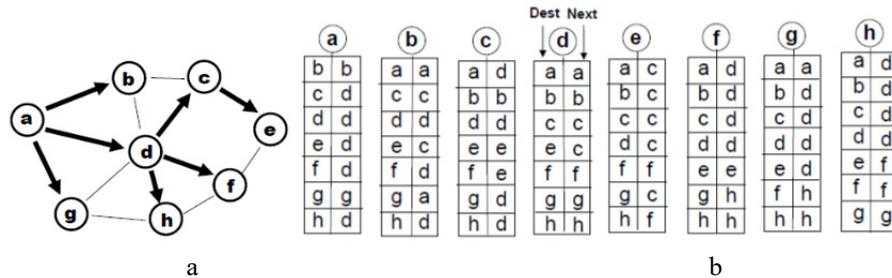


Figure 6. (a) A network with shortest path tree from a node; (b) Full collection of routing tables in all nodes

For example, finding the shortest path from any node, say *g*, to any other node, like *e*, via the routing tables of Figure 6,b can be organized in SGL as follows:

```
hop(g); repeat(hop(link(any), node(element(Next, order(Dest, e)))))
```

This scenario will follow the path $g \rightarrow d \rightarrow c \rightarrow e$ with using the tables of Figure 6 in nodes *g*, *d*, and *c*.

5. Examples of Solving Problems with Stable Constellation Networks

5.1 Collection and Return of Distributed Data

As an example of advantage of such stable network topology, we may show how simple will be picking up certain items in the given destinations and their return, collection, and output at the ground station, using registered paths from the start to destinations in a reverse order. (See for comparison the related previous examples for such a task but for unpredictable and rapidly changeable constellation structures in Section 3.)

```
frontal(Item = ..., Dest = (a, b, c, d));
output(
  hopfirst_node(any);
  repeat(if(belong(NAME, Dest),
    free(element(CONTENT, Item)));
  hopfirst_links(all)))
```

This may be even simpler if we introduce such possibility as direct hops to given nodes by their names, which can be easily achieved with the use of routing tables (see above) providing shorted paths to the needed nodes from the starting satellite. And then these paths after being registered in the network structure are providing echoing and collecting proper items from the reached needed destinations and outputting them altogether in the starting node (see also Figure 7).

```
frontal(Item) = ...; Nodal(Dest) = (a, b, c, d);
output(hop_nodes(Dest); element(CONTENT, Item))
```

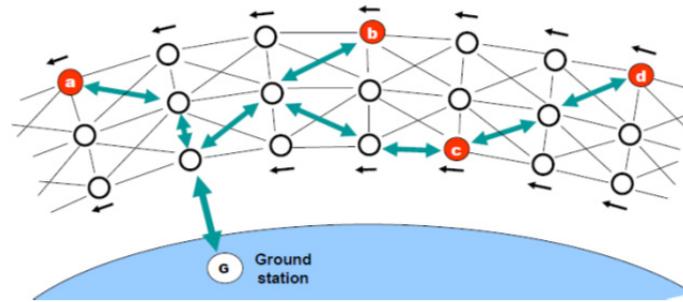


Figure 7. Collection and return of remote data in a stable constellation network

5.2 Integration and Support of the Next-Generation Space Architecture

The recently launched by Space Development Agency the notional architecture (Space Development Agency Next-Generation Space Architecture, 2019; Magnuson, 2019; Messier, 2019; Insinna, 2019; Strout, 2021) is made up of several layers (see also Figure 8):

- 1). A space transport layer is a global mesh network providing data and communications.
- 2). A tracking layer provides tracking, targeting and advanced warning of missile threats.
- 3). A custody layer provides all-weather custody of all identified time-critical targets.
- 4). A deterrence layer provides space situational awareness by detecting and tracking objects.
- 5). A navigation layer provides alternative positioning, navigation and timing services.
- 6). A battle management layer is a command, control and communication network.
- 7). A support layer: includes ground command, control facilities and user terminals.

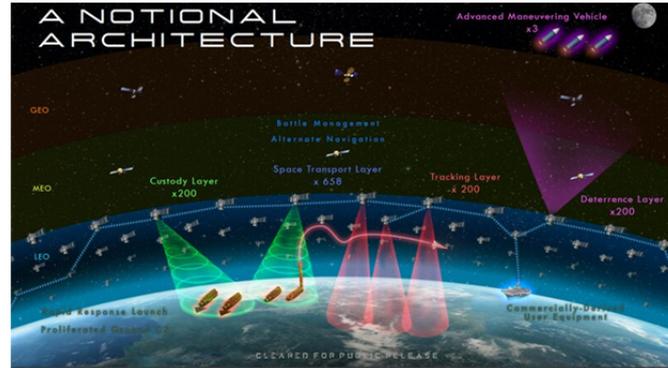


Figure 8. Next-Generation Space Architecture

It plans to fight growing space-based threats, to move quickly on hypersonic defense and track hypersonic threats from space, also arm satellites with lasers to shoot down missiles, and so on. This architecture is oriented on intensive cooperation and collective behavior of many satellites, which can be effectively managed by SGT. We will show here how the tracking and custody layers for this architecture can be implemented with the following integral SGL scenario.

```

frontal(Custody = X_Y, History, Threshold1 = ...,
        Threshold2 = ..., Threshold3 = ...);
min_destination(hop(all); distance(WHERE, Custody));
repeat (
    if(distance(WHERE, Custody) <= Threshold1,
        (update(History, observe(Custody));

```

```

if(belong(glider_launch, History),
  free_repeat(
    loop(if(visibility(glider) >= Threshold2,
      if(and(TYPE = destroyer,
        distance(Object, WHERE) < Threshold3),
        pursue_destroy(glider))));
    max_destination(
      hop(neighbors); visibility(glider))),
  min_destination(
    hop(neighbors); distance(WHERE, Custody)))

```

After fixing a glider launch at the constantly observed custody object, the glider-tracing mobile intelligence is activated in SGL, which will accompany this glider wherever it goes via the satellite network. This will be accomplished via the sensors mounted on satellites which can communicate directly, with elimination of this object if possible, as shown in Figure 9.

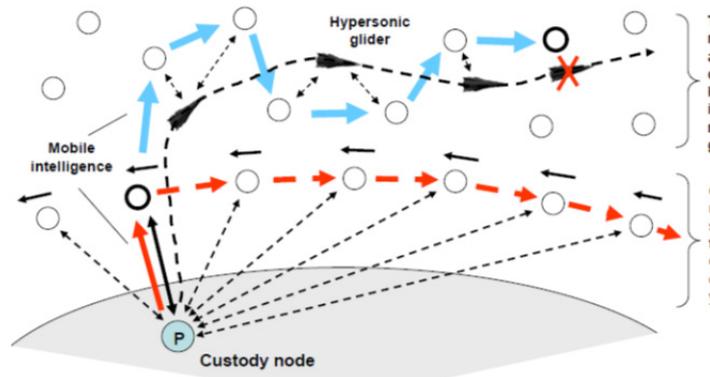


Figure 9. Integration of SDA's custody and tracking layers under SGT

The custody-related SGL scenario will work repeatedly and endlessly, and if a new glider launch is detected in the observed custody, another tracing intelligence will be associated with this object, which will follow it via the satellite networks, and so on. More SGL solutions for both custody observation and glider-tracing can be found in (Sapaty, 2021b; Sapaty, 2021c).

6. Conclusions

This concludes the paper with a continuing research and development in this area, with more international publications planned, the new book being completed too (Sapaty, 2021c). The paper has offered a unified approach toward organization of large satellite constellations, especially on Low Earth Orbits, and on any stages of their development, starting from their accidental launches and gradual growth. Such constellations at the beginning may not have stable communication structures and network topologies similar to the existing terrestrial systems. This approach with wavelike mobile spatial code covering distributed spaces in a spatial pattern-matching mode is ideologically stemming from the organization and implementation of first citywide heterogeneous computer networks half a century ago, well before the internet. It has grown up into a real high level networking technology that can effectively cover the rapidly growing celestial systems -- from elementary communication protocols to high-level holistic solutions of very complex problems, both civil and military. Another planned application is oriented on collective removal of huge space debris which can endanger any further human activities in space (Space Debris, NASA Headquarters Library, 2021; Sapaty, 2021e; 2021f). The described latest SGT version, suitable for the advanced constellations management, can be quickly implemented on any platforms, even within traditional university environments, similar to the previous technology versions in different countries.

References

- Bockel, J. M. (2018). The Future of the Space Industry-General Report. *NATO Parliamentary Assembly-Economic and Security Committee (ESC), France*.
- Bondarenko, A. T., Karpus, V. P., Mikhalevich, S. B., Nikitin, A. I., & Sapaty, P. S. (1972). "Information-computing system ABONENT", *Tech. Report No. B178338, All-Union Scientific and Technical Inform. Centre, Moscow, 1972 (in Russian)*.
- Bondarenko, A. T., Mikhalevich, S. B., Nikitin, A. I., & Sapaty, P. S. (1970). Software of BESM-6 computer for communication with peripheral computers via telephone channels. *Computer Software, 5*.
- Curzi, G., Modenini, D., & Tortora, P. (2020). Large constellations of small satellites: A survey of near future challenges and missions. *Aerospace, 7(9)*, 133. <https://doi.org/10.3390/aerospace7090133>
- Foust, J. (2016). Mega-constellations and mega-debris. *The Space Review*. Retrieved October 10, 2016, from <https://www.thespacereview.com/article/3078/1>
- Insinna, V. (2019). Space agency has an ambitious plan to launch 'hundreds' of small satellites. Can it get off the ground. *DefenceNews. Space*. Retrieved from <https://www.defensenews.com/space/2019/04/10/sda-has-an-ambitious-plan-to-launch-hundreds-of-small-satellites-can-it-get-off-the-ground/>
- Jones, A. (2021). China Is Developing Plans for a 13,000-Satellite Megaconstellation. *Space News*. Retrieved April 21, 2021, from <https://spacenews.com/china-is-developing-plans-for-a-13000-satellite-communications-megaconstellation/>
- Klondike Gold Rush. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Klondike_Gold_Rush
- Magnuson, S. (2019). *Web Exclusive: Details of the Pentagon's New Space Architecture Revealed*. Retrieved from <https://www.nationaldefensemagazine.org/articles/2019/9/19/details-of-the-pentagon-new-spacearchitecture-revealed>
- Martin, G. L. (2014). *NewSpace: The emerging commercial space industry* (No. ARC-E-DAA-TN14888). Retrieved from <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140011156.pdf>
- Messier, D. (2019). *Space Development Agency Seeks Next-Gen Architecture in First RFI*. Retrieved from <http://www.parabolicarc.com/2019/07/07/space-development-agency-issues-rfi/>
- Minet, M. (2020). *The Space Legal Issues with Megs-Constellations*. Retrieved November 3, 2020, from <https://www.spacelegalissues.com/mega-constellations-a-gordian-knot/>
- Mohanta, N. (2021). *How many satellites are orbiting the Earth in 2021?* Geospatial World. Retrieved May 28, 2021, from <https://www.geospatialworld.net/blogs/how-many-satellites-are-orbiting-the-earth-in-2021/>
- Reiland, N., Rosengren, A. J., Malhotra, R., & Bombardelli, C. (2021). Assessing and minimizing collisions in satellite mega-constellations. *Advances in Space Research, 67(11)*, 3755-3774. <https://doi.org/10.1016/j.asr.2021.01.010>
- Sapaty, P. (1999). *Mobile processing in distributed and open environments* (Vol. 10, p. 410). New York: John Wiley & Sons.
- Sapaty, P. (2005). *Ruling distributed dynamic worlds* (Vol. 65, p. 255). New York: John Wiley & Sons. <https://doi.org/10.1002/0471656356>
- Sapaty, P. (2021a). *Symbiosis of Real and Simulated Worlds Under Spatial Grasp Technology* (Vol. 354). Springer Nature. <https://doi.org/10.1007/978-3-030-68341-2>
- Sapaty, P. (2021b). Managing multiple satellite architectures by spatial grasp technology. *Mathematical machines and systems, (1)*, 3-16. <https://doi.org/10.34121/1028-9763-2021-1-3-16>
- Sapaty, P. (2021c). Spatial Grasp as a Model for Space-based Control and Management Systems. *Mathematical machines and systems, (1)*, 135-138. Retrieved from http://www.immsp.kiev.ua/publications/articles/2021/2021_1/Sapaty_book_1_2021.pdf
- Sapaty, P. A. (n.d.). Distributed Processing System: European Patent. № 0389655. *Publ, 10(93)*, 35. European Patent Office.
- Sapaty, P. S. (1973). A Method of organization of an intercomputer dialogue in the radial computer systems. *The Design of Software and Hardware for Automatic Control Systems*.

- Sapaty, P. S. (2017). *Managing Distributed Dynamic Systems with Spatial Grasp Technology* (p. 284). Springer International Publishing. <https://doi.org/10.1007/978-3-319-50461-2>
- Sapaty, P. S. (2018). *Holistic analysis and management of distributed social systems* (p. 234). Springer. <https://doi.org/10.1007/978-3-030-01830-6>
- Sapaty, P. S. (2019). *Complexity in International Security: A Holistic Spatial Approach* (p. 160). Emerald Group Publishing. <https://doi.org/10.1108/9781789737158>
- Sapaty, P. S. (2020). Global network management under spatial grasp paradigm. *Global Journal of Research In Engineering*, 20(5), 58-69. Retrieved from <https://engineeringresearch.org/index.php/GJRE/article/view/2082/2013>
- Sapaty, P. S. (2021d). Spatial Management of Large Constellations of Small Satellites. *Mathematical machines and systems*, (2), 3-14. Retrieved from http://www.immsp.kiev.ua/publications/articles/2021/2021_2/02_21_Sapaty.pdf
- Sapaty, P. S. (2021e). Global Management of Space Debris Removal Under Spatial Grasp Technology. *Acta Scientific Computer Sciences*, 3(7). Retrieved from <https://www.actascientific.com/ASCS/pdf/ASCS-03-0135.pdf>
- Sapaty, P. S. (2021f). Space Debris Removal under Spatial Grasp Technology. *Network and Communication Technologies*, 6(1), 1-16. <https://doi.org/10.5539/nct.v6n1p16>
- Sapaty, P. S. (2021g). Spatial Grasp Model for Management of Dynamic Distributed Systems. *Acta Scientific Computer Sciences*, 3(9). Retrieved from <https://www.actascientific.com/ASCS/pdf/ASCS-03-0170.pdf>
- Sapaty, P. S. (2021h). Spatial Grasp Model for Dynamic Distributed Systems. *Mathematical machines and systems*, (3), 3-21. <https://doi.org/10.34121/1028-9763-2021-4-3-14>
- Siegel, E. (2021). *Astronomy Faces A Mega-Crisis As Satellite Mega-Constellations Loom*. Retrieved Jan 19, 2021, from <https://www.forbes.com/sites/startswithabang/2021/01/19/astronomy-faces-a-mega-crisis-as-satellite-mega-constellations-loom/?sh=30597dca300d>
- Skibba, R. (2020). How satellite mega-constellations will change the way we use space. Retrieved February 26, 2020, from <https://www.technologyreview.com/2020/02/26/905733/satellite-mega-constellations-change-the-way-we-use-space-moon-mars/>
- Space Debris, NASA Headquarters Library. (2021). Retrieved from https://www.nasa.gov/centers/hq/library/find/bibliographies/space_debris
- Space Development Agency Next-Generation Space Architecture. (2019). Retrieved from [https://www.airforcemag.com/PDF/DocumentFile/Documents/2019/SDA_Next_Generation_Space_Architecture_RFI%20\(1\).pdf](https://www.airforcemag.com/PDF/DocumentFile/Documents/2019/SDA_Next_Generation_Space_Architecture_RFI%20(1).pdf)
- Strout, N. (2021). *Space Development Agency approves design for satellites that can track hypersonic weapons*. C4ISRNET, 2021. Retrieved from <https://www.c4isrnet.com/battlefield-tech/space/2021/09/20/space-development-agency-approves-design-for-satellites-that-can-track-hypersonic-weapons/>
- United Nations Register of Objects Launched into Outer Space, The United Nations Office for Outer Space Affairs. (n.d.). Retrieved from <http://www.unoosa.org/oosa/en/spaceobjectregister/index.html>.
- Venkatesan, A., Lowenthal, J., Prem, P., & Vidauri, M. (2020). The impact of satellite constellations on space as an ancestral global commons. *Nature Astronomy*, 4(11), 1043-1048. <https://doi.org/10.1038/s41550-020-01238-3>
- What Was the Klondike Gold Rush? (n.d.). Retrieved from <https://www.nps.gov/klgo/learn/goldrush.htm>

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).