https://orcid.org/0000-0002-4168-7190

UDC 623.764

P.S. SAPATY*

DEVELOPMENT OF SPACE-BASED DISTRIBUTED SYSTEMS UNDER SPATIAL GRASP TECHNOLOGY

*Institute of Mathematical Machines and Systems Problems of the National Academy of Sciences of Ukraine, Kyiv, Ukraine

Анотація. *Наразі багато державних установ і приватних компаній із різних країн світу прямують у космічний простір навколо Землі, сподіваючись знайти ефективні рішення проблем сфер комунікації, промисловості, безпеки та оборони. Такі дії часто передбачають масові запуски невеликих дешевих супутників, що, у свою чергу, призводить до збільшення кількості космічного сміття. У даній статті розглядається, як розроблені філософія та модель системи високого рівня можуть ефективно організувати розподілені космічні системи на різних етапах їх розвитку та розростання. Технологія просторового захоплення, яка була створена в результаті паралельного співставлення розподілених середовищ із високорівневим рекурсивним мобільним кодом, здатна ефективно забезпечувати будь-які мережеві протоколи та важливі засоби застосування сузір'їв великих супутників, насамперед тих, які знаходяться на низьких навколоземних орбітах. У статті наведено ряд прикладів технологічних рішень для встановлення базового зв'язку між супутниками, починаючи від їхніх перших, часто хаотичних, запусків, а також розподілу та збору даних у сузір'ях, які розростаються, навіть із нестабільними та швидко змінними зв'язками між супутниками. Робота описує, як організувати та зареєструвати топології мережі у випадку передбачуваних відстаней між супутниками, і як саме фіксовані мережеві структури можуть допомогти у вирішенні складних проблем. Ці структури, а також ті, які стосуються нової архітектури Агентства космічного розвитку, що складається з багатьох супутників і є орієнтованою на забезпечення безпеки, дозволяє ефективну інтеграцію безперервного спостереження за Землею та пошук спільних рівнів відстеження і ліквідації ракет на основі саморозповсюджуваної мобільної розвідки. Попередні версії технології були описані у багатьох статтях і шести книжках, взяті у розробку і використані в різних країнах світу, тоді як остання версія також може ефективно реалізовуватися навіть в університетських умовах.*

Ключові слова: *завоювання космосу, сузір'я супутників, Технологія просторового захоплення, протоколи зв'язку, нова архітектура Агентства космічного розвитку, рівні транспортування, контролю та відстеження.*

Abstract. *Many governmental agencies and private companies of different countries are now rushing into space around Earth in the hope to provide smart communication, industrial, security and defense solutions. This often involves massive launches of small cheap 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 in low Earth orbits. The article gives some 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 includes those related to the new Space Development Agency's multiple-satellite defense-oriented architecture and allows for effective integration of its continuous Earth custody observation and cooperative missile tracking and elimina-*

ISSN 1028-9763. Математичні машини і системи. 2021. № 4

tion 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.*

DOI: 10.34121/1028-9763-2021-4-3-14

1. Introduction

Different countries are now chaotically rushing into space in the hope to provide quick and smart communication, climate, industrial, security and defense solutions [3–20]. This often involves massive launches of cheap and unsafe small satellites in low Earth orbits, which also contribute 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 [3]. In total, 11,139 satellites were launched, out of which only 7,389 are in 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 [4], there are millions of pieces of junk flying in LEO, which comprises of spacecraft, tiny flecks of paint from spacecraft, parts of rockets and satellites that are either destroyed or lost, including objects that are results of explosions in space.

But this massive and often competitive penetration into space in an attempt to gain something very quickly, possibly cheaply and ahead of others, is not based on any global culture and UN-supported planning. The author of this paper witnessed a similar situation more than half a century ago, but on Earth, having been actively engaged in the creation of the first citywide heterogeneous computer networks (actually inventing and writing himself so far unknown communication protocols) combining different and even incompatible machines [21–23], well before the Internet. Having offered at that time how to unite such distributed equipment into 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 the hope to bring certain order into this enormously and chaotically growing multiple satellite and junk mess around Earth, with its symbolic picture shown in Fig. 1.

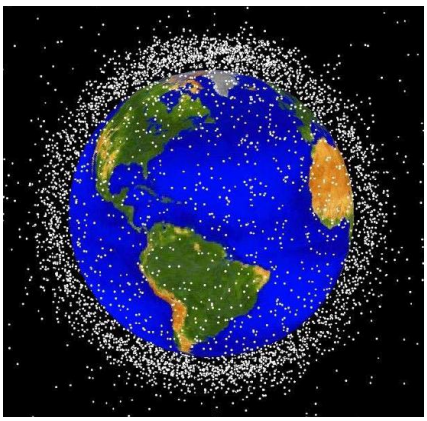

Figure 1 – Rapidly growing multiple objects around Earth

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, as well as 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 describes stable constellation topologies and explains their advantages, 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 the ways of 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 and mentions plans for continuing research and development in this area, as well as the possibility of quick implementation of the described latest SGT version for advanced constellations management.

The aim of the article is to develop a unified approach to the organization of large satellite constellations, especially in low Earth orbits and on any stages of their development, starting from their accidental launches and gradual growth, taking into account that at the beginning such constellations may have no stable communication structures and network topologies which we used to see on Earth.

2. Spatial Grasp Technology (SGT)

SGT operates by spatial scenarios self-spreading in physical and virtual worlds during their creation, matching, transformation and management. 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.

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 following top-level syntax:

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***

Worlds SGT operates with

SGT allows us to directly operate with the following world representations: Physical World which is considered as continuous and infinite; Virtual World which is discrete and consists of nodes and semantic links between them; and Executive world which consists of active «doers» with communication possibilities between them. Different kinds of combinations of these worlds are also possible within the same formalism, for example, Virtual-Physical World, Virtual-Execution World, Execution-Physical World, and also Virtual-Execution-Physical World combining all features of the previous cases.

SGL networked interpreter

Communicating Interpreters of SGL can be in an arbitrary number of copies 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's main components are shown in Fig. 2.

As both backbone and nerve systems 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.

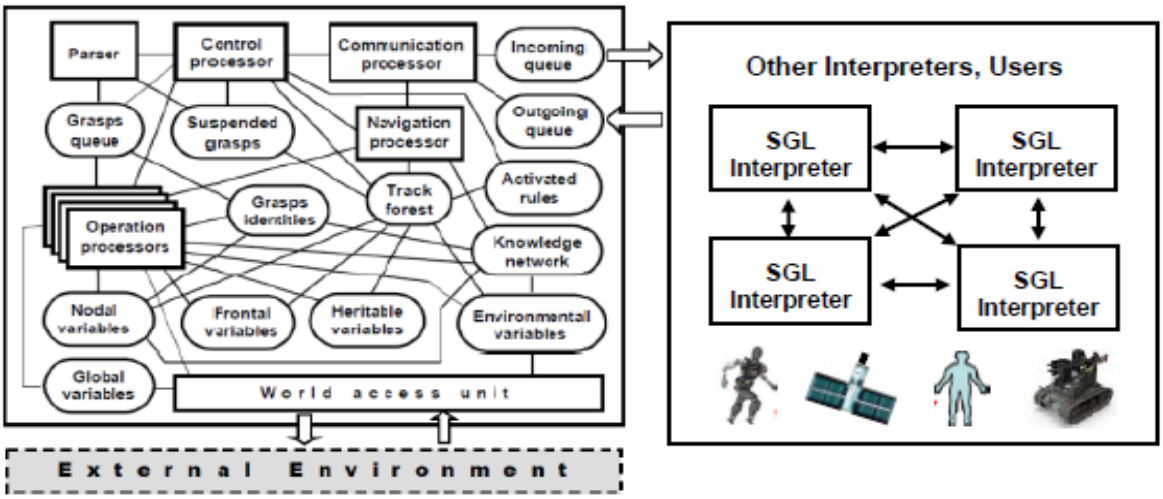

Figure 2 – Main components of the SGL interpreter and its networked organization

Figure 3 – An example of multiple and complex satellite constellation orbits

3. Dealing with highly dynamic constellation topologies under SGT

In general, constellations may consist of satellites in different orbits which over time may change their distances to each other, so having no stable network topologies, as symbolically shown in Fig. 3. But it is still assumed that there are enough of them around Earth, and at any moment any satellite can directly communicate with other satellites (at least one of them), which may differ a lot from the previous communications. So this in principle could enable to propagate through the whole constellation regardless of satellites which can be seen from each other, and also from a ground station.

Some basic operations in such dynamic networks under SGT, expressed in SGL, will be considered.

Reaching proper satellite nodes

Reaching particular satellite node a from a ground station (Fig. 4) with some such nodes of interest initially via an arbitrary satellite visible from it at this moment, can be accomplished as follows, with subsequent propagation via other satellites (if this is not the very starting node). The satellites of interest are supposed to be of a 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 the done statement. The operation starts with reaching any new satellite node and its proper marking, then blocking a repeated entry and therefore any navigation cycling that otherwise could become endless.

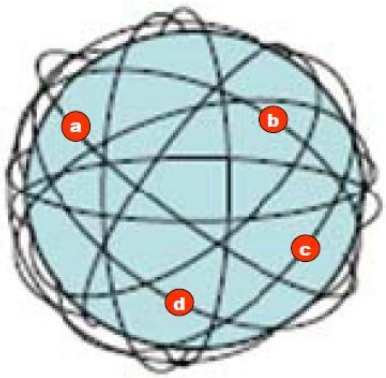

Figure 4 – Certain nodes to be reaches in complex satellite constel-

```
frontal(Depth = …, Type = …);
hopfirst(any, Type);
```

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

Reaching a particular satellite node from another one after initially staying in it, without the 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 a particular group of nodes `Dest` from another satellite node after initially staying in it where the found nodes may also serve as transits to other `Dest` nodes, can be as follows:

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

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

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

Spatial recursion can be also utilized 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)
```

Package delivery to all proper destinations

Reaching the set of particular nodes with delivering a package into them 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 follows:

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

Moving to proper destinations with returning and collecting packs from them all

Let's 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 the ground station. From the very beginning, mobile environmental variable `IDENTITY` has default value like the starting node name, address (if it has it), or just a 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 collection of packs 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 for the completion of the 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 that remain 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))
```

Providing additional measures for dealing with unpredictable topology dynamics

Below there is considered the possibility of reaching particular nodes from the ground station by allowing nodes to be entered more than once, assuming that over time other nodes may become directly accessible from the same nodes. The additional parameter `Revisit` for the rule `hop-first` 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))
```

The `hopfirst` option also uses the quality of the `hopforth` rule not allowing reentering the nodes visited just 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 that are all moving in similar orbits, as in Fig. 5, it is possible to describe and solve various problems in constellations of satellites much easier and quicker.

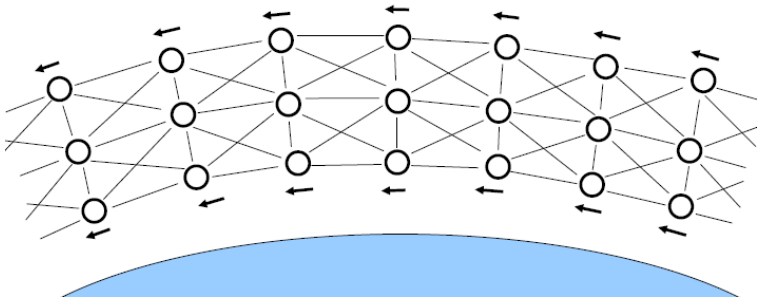

Figure 5 – Example of a stable constellation network

Creation of a stable constellation network topology, its advantages

Starting from the ground station and linking certain `Type` nodes close enough in space, there can be obtained 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 [24–38], for example, for finding the shortest paths, routing tables, weak and strong components like articulation points and cliques, any particular structures and components (represented as spatial images or patterns), etc. It may be very useful for large constellations management, for instance, for the creation of celestial version of the Internet, weather prediction, managing global industry, security and defense.

As an example there can be considered an effective and highly parallel SGL scenario finding the shortest-path trees and full routing tables (as in [24–26]), with an example of a network and related routing tables for all its nodes shown in Fig. 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.

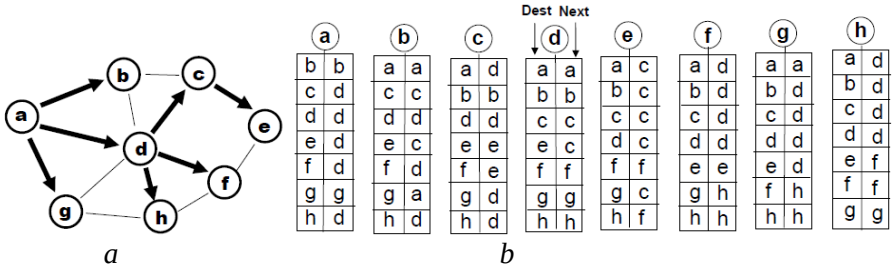

Figure 6 – (a) A network with the shortest path tree from a node;
(b) Full collection of routing tables in all nodes

For example, finding the shortest path from any node, say `g`, to any other node, like `e`, via the routing tables of Fig. 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`→`d`→`c`→`e` with using the tables of Fig. 6 in nodes `g`, `d`, and `c`.

5. Examples of solving problems with stable constellation networks

Collection and return of distributed data

As an example of an advantage of such stable network topology, it can be shown that 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 reverse order can be very simple. (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 possibilities as direct hops to the 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. Then these paths after be-

ing registered in the network structure provide echoing and collecting proper items from the reached needed destinations and outputting them all together in the starting node (Fig. 7).

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

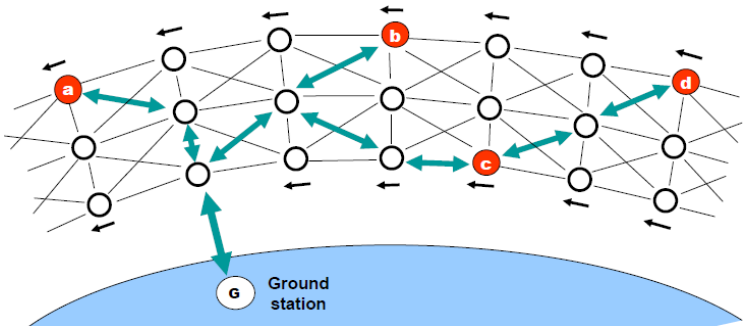

Figure 7 – Collection and return of remote data in a stable constellation network

Integration and support of the next-generation space architecture

Space Development Agency has recently launched a notional architecture [16–20] which is made up of several layers (Fig. 8):

– a space transport layer that is a global mesh network providing data and communications;

– a tracking layer that provides tracking, targeting and advanced warning of missile threats;

– a custody layer that provides all-weather custody of all identified time-critical targets;

– a deterrence layer that provides situational awareness by detecting and tracking objects;

– a navigation layer that provides alternative positioning, navigation and timing services;

– a battle management layer that is a command, control and communication network;

– a support layer that includes ground command, control facilities and user terminals.

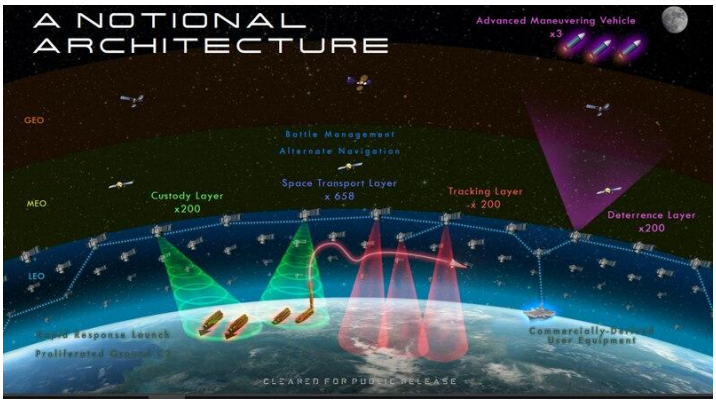

Figure 8 – Next-generation space architecture

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. Below there is shown 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 the elimination of this object if possible, as shown in Fig. 9.

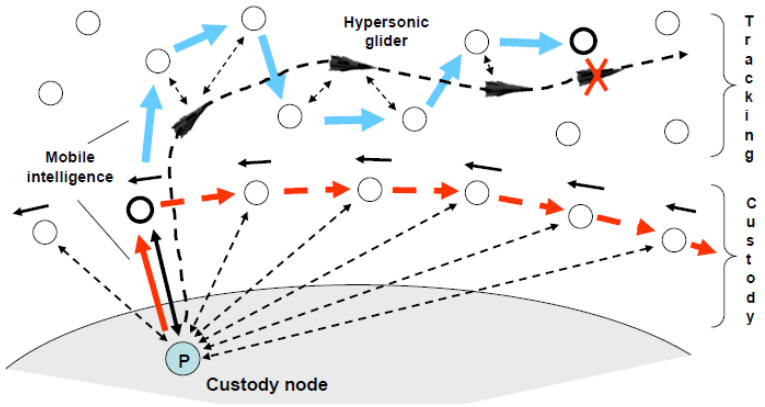

Figure 9 – Integration of SDA's custody and tracking layers under SGT

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 [32, 33].

6. Conclusions

The paper has offered a unified approach toward the organization of large satellite constellations, especially in low Earth orbits, and on any stages of their development, starting from their accidental launches and gradual growth. At the beginning, such constellations may have no stable communication structures and network topologies similar to the existing terrestrial systems. This approach with a 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 [4, 35, 36]. The described latest SGT version, suitable for the advanced constellations management, can be quickly implemented on any platform, even within traditional university environments, similar to the previous technology versions in different countries.

REFERENCES

1. Klondike Gold Rush. URL: https://en.wikipedia.org/wiki/Klondike_Gold_Rush.
2. What Was the Klondike Gold Rush? URL: https://www.nps.gov/klgo/learn/goldrush.htm.
3. Mohanta N. How many satellites are orbiting the Earth in 2021? *Geospatial World*. 2021. N 05/28. URL: https://www.geospatialworld.net/blogs/how-many-satellites-are-orbiting-the-earth-in-2021/.
4. Space Debris, NASA Headquarters Library. URL: https://www.nasa.gov/centers/hq/library/find/bibliographies/ space_debris.
5. United Nations Register of Objects Launched into Outer Space. The United Nations Office for Outer Space Affairs. URL: http://www.unoosa.org/oosa/en/spaceobjectregister/index.html.
6. Martin G. NewSpace: The «Emerging» Commercial Space Industry. 2014. June 30. URL: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140011156.pdf.
7. Bockel J.-M. The Future of the Space Industry. General Report. 2018. November 17. URL: https://www.nato-pa.int/download-file?filename=sites/default/files/2018-12/2018%20-%20THE%20 FUTURE%20OF%20SPACE%20INDUSTRY%20-%20BOCKEL%20REPORT%20-%20173%20ESC%2018%20E%20fin.pdf.
8. Curzi G., Modenini D., Tortora P. Review Large Constellations of Small Satellites: A Survey of Near Future Challenges and Missions. *Aerospace*. 2020. Vol. **7** (9), N 133. URL: https://www.mdpi.com/2226-4310/7/9/133/htm.
9. Venkatesan A., Lowenthal J., Prem P., Vidaurri M. The impact of satellite constellations on space as an ancestral global commons. *Nature Astronomy*. 2020. Vol. 4. P. 1043–1048. www.nature.com/ natureastronomy.
10. Skibba R. How satellite mega-constellations will change the way we use space. *MIT Technology Review*. 2020. February 26. URL: https://www.technologyreview.com/2020/02/26/905733/satellite-mega-constellations-change-the-way-we-use-space-moon-mars/.
11. Minet M. The Space Legal Issues with Mega-Constellations. *Space Legal Issues*. 2020. November 3. URL: https://www.spacelegalissues.com/mega-constellations-a-gordian-knot/.
12. Siegel E. Astronomy Faces A Mega-Crisis As Satellite Mega-Constellations Loom. 2021. January 19. URL: https://www.forbes.com/sites/startswithabang/2021/01/19/astronomy-faces-a-mega-crisis-as-satellite-mega-constellations-loom/?sh=30597dca300d.
13. Jones A. China is developing plans for a 13,000-satellite mega-constellation. *Space News*. 2021. April 21. URL: https://spacenews.com/china-is-developing-plans-for-a-13000-satellite-communications-megaconstellation/.
14. Reilanda N., Rosengren A.J., Malhotra R., Bombardelli C. Assessing and minimizing collisions in satellite mega-constellations. Published by Elsevier B.V. on behalf of COSPAR. 2021. URL: https://www.sciencedirect.com/science/article/abs/pii/S0273117721000326.
15. Fous J. Mega-constellations and mega-debris. 2016. October 10. URL: https://www.thespacereview.com/ article/3078/1.
16. Space Development Agency Next-Generation Space Architecture. 2019. https://www.airforcemag.com/PDF/DocumentFile/Documents/2019/SDA_Next_Generation_Space_Architecture_RFI%20(1).pdf.
17. Magnuson S. Web Exlusive: Details of the Pentagon's New Space Architecture Revealed. 2019. URL: https://www.nationaldefensemagazine.org/articles/2019/9/19/details-of-the-pentagon-new-space-architecture-revealed.
18. Messier D. Space Development Agency Seeks Next-Gen Architecture in First RFI. 2019. URL: http://www.parabolicarc.com/2019/07/07/space-development-agency-issues-rfi/.
19. Insinna V. Space agency has an ambitious plan to launch 'hundreds' of small satellites. Can it get off the ground? *DefenceNews. Space.* 2019. April 10. URL: 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/.
20. Strout N. Space Development Agency approves design for satellites that can track hypersonic weapons. *C4ISRNET*. 2021. September 20. URL: https://www.c4isrnet.com/battlefield-tech/space/2021/09/20/space-development-agency-approves-design-for-satellites-that-can-track-hypersonic-weapons/.
21. Bondarenko A.T., Mikhalevich S.B., Nikitin A.I., Sapaty P.S. Software of BESM-6 computer for communication with peripheral computers via telephone channels. *Computer Software*. 1970. Vol. 5.
22. Bondarenko A.T., Karpus V.P., Mikhalevich S.B., Nikitin A.I., Sapaty P.S. Information-computing system ABONENT: Tech. Report N B178338. All-Union Scientific and Technical Inform. Centre. Moscow, 1972.
23. Sapaty P.S. A Method of organization of an intercomputer dialogue in the radial computer systems. *The Design of Software and Hardware for Automatic Control Systems, Inst. of Cybernetics Press*. Kiev, 1973.
24. Sapaty P.S. Symbiosis of Real and Simulated Worlds under Spatial Grasp Technology. Springer, 2021. 251 p.
25. Sapaty P.S. Complexity in International Security: A Holistic Spatial Approach. Emerald Publishing, 2019. 160 p**.**
26. Sapaty P.S. Holistic Analysis and Management of Distributed Social Systems. Springer, 2018. 234 p.
27. Sapaty P.S. Managing Distributed Dynamic Systems with Spatial Grasp Technology. Springer, 2017. 284 p.
28. Sapaty P.S. Ruling Distributed Dynamic Worlds. New York: John Wiley & Sons, 2005. 255 p.
29. Sapaty P.S. Mobile Processing in Distributed and Open Environments. New York: John Wiley & Sons, 1999. 410 p.
30. Sapaty P.S. A distributed processing system: European Patent N 0389655, Publ. 10.11.93, European Patent Office. 35 p.
31. Sapaty P.S. Global Network Management under Spatial Grasp Paradigm. *Global Journal of Researches in Engineering: J General Engineering.* 2020. Vol. 20, Issue 5, Version 1.0. P. 58–69. https://engineeringresearch.org/index.php/GJRE/article/view/2082/2013.
32. Sapaty P.S. Managing Multiple Satellite Architectures by Spatial Grasp Technology. *Mathematical machines and systems*. 2021. N 1. P. 3–16. URL: http://www.immsp.kiev.ua/publications/eng/2021_1/.
33. Sapaty P.S. Spatial Grasp as a Model for Space-Based Control and Management Systems. *Mathematical machines and systems*. 2021. N 1. P. 135–138. URL: http://www.immsp.kiev.ua/publications/articles/2021/2021_1/Sapaty_book_1_2021.pdf.
34. Sapaty P.S. Spatial Management of Large Constellations of Small Satellites. *Mathematical machines and systems*. 2021. N 2. P. 3–14. URL: http://www.immsp.kiev.ua/publications/articles/2021/2021_2/02_21_Sapaty.pdf.
35. Sapaty P.S. Global Management of Space Debris Removal Under Spatial Grasp Technology. *Acta Scientific Computer Sciences.* 2021. Vol. 3, Issue 7. URL: https://www.actascientific.com/ASCS/pdf/ASCS-03-0135.pdf.
36. Sapaty P.S. Space Debris Removal under Spatial Grasp Technology. *Network and Communication Technologies*. 2021. N 1, Vol. 6. URL: https://www.ccsenet.org/journal/index.php/nct/article/view/0/45486.
37. Sapaty P.S. Spatial Grasp Model for Management of Dynamic Distributed Systems. *Acta Scientific Computer Sciences*. 2021. Vol. 3, Issue 9. URL: https://www.actascientific.com/_ASCS/pdf/ASCS-03-0170.pdf.
38. Sapaty P.S. Spatial Grasp Model for Dynamic Distributed Systems. *Mathematical machines and systems*. 2021. N 3. P. 3–21. URL: http://www.immsp.kiev.ua/publications/articles/2021/2021_3/03_21_Sapaty.pdf.

Стаття надійшла до редакції 11.10.2021