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

UDC 623.764

P.S. SAPATY*

HOLISTIC SPATIAL ANALYSIS OF DISTRIBUTED WORLDS

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

Анотація. *У статті досліджуються різноманітні способи застосування розробленої моделі просторового захоплення, технології аналізу великих розподілених систем і середовищ, а також ряд прикладів вирішення типових пов'язаних із ними задач за допомогою Мови просторового захоплення (МПЗ). Парадигма просторового захоплення (ПЗ) дозволяє вирішувати складні проблеми цілісним і повністю розподіленим способом. Дана парадигма розвивається у розподілених середовищах як активні нескінченні хвилі або навіть віруси і охоплює рішення просторових проблем у режимі паралельного узгодження патернів, принципово відрізняючись від традиційних уявлень про системи та їх рішень як частин, що обмінюються повідомленнями. У роботі наводяться результуючі дані про Технологію просторового захоплення, де її інтерпретатори МПЗ можуть об'єднуватися в мережу як потужні просторові комп'ютери, що охоплюють будь-які наземні і небесні середовища, а також вирішувати проблеми без будь-яких централізованих ресурсів. До нових сфер більш широкого застосування ПЗ належать основні проблеми навколишнього середовища, глобальні системи, відкриття нових світів, наука про Землю і діяльність NASA, спрямована на дослідження планет. Стаття містить описи рішень практичних проблем за допомогою МПЗ, пов'язаних із різними світами, включаючи групову поведінку морських тварин, особливості географічних рельєфів, управління транспортними мережами і дослідження інформаційних мереж. Розроблена парадигма дає можливість безпосереднього вираження вищої семантики та цілісних методів вирішення складних проблем. Вона динамічно створює необхідні середовища для її реалізації, забезпечуючи таким чином чіткий шлях від визначення проблеми до її практичного вирішення. Формульні високорівневі рішення у МПЗ є надзвичайно компактними, адже досить часто вони є в сто разів коротшими, ніж в інших мовах, а їх реалізацію можна здійснювати на будь-яких існуючих платформах, як і для попередніх версій мови у різних країнах.*

Ключові слова: *глобальні системи, відкриття нового світу, НАСА ЛРД, Технологія просторового захоплення, Мова просторового захоплення, розподілена інтерпретація, розподілена мережа, узгодження просторового патерну.*

Abstract. *The paper investigates extended applications of the developed spatial grasp model and technology for analyzing large distributed systems and environments, as well as some examples of solving typical problems in them in the Spatial Grasp Language (SGL). The Spatial Grasp (SG) paradigm allows solving complex problems in a holistic and fully distributed way. It develops in distributed spaces as active ubiquitous waves or even viruses and grasps solutions to spatial problems in parallel pattern-matching mode, fundamentally differing from traditional representations of systems and their solutions as parts that exchange messages. The resultant Spatial Grasp Technology details are briefed where its SGL interpreters can be networked as powerful spatial computers covering any terrestrial and celestial environments and solving problems without any centralized resources. The extended areas for new SG applications include basic environmental issues, global systems, discovering new worlds, Earth science, and planetary exploration activities at NASA. The paper contains descriptions of solutions in SGL to practical problems related to different worlds, including group behavior of marine animals, details of geographical terrain, management of transport networks, and investigation of information networks. The developed paradigm allows direct expression of top semantics and holistic methods for solving complex problems and dynamically composes the needed implementation environments, thus providing the strictest way from problem*

definition to a practical solution. The formula-like high-level solutions in SGL are extremely compact, often a hundred times shorter than in other languages, and its implementation can be accomplished on any existing platforms, as for the previous language versions in different countries.
Keywords: *global systems, new world discovery, NASA JPL, Spatial Grasp Technology, Spatial Grasp Language, distributed interpretation, distributed networking, spatial pattern matching.*

DOI: 10.34121/1028-9763-2022-2-3-18

1. Introduction

Having been ideologically born more than half a century ago and called WAVE in its childhood, it represents a fundamentally different approach to solving problems in large distributed systems and networks, where instead of placement of communicating activities inside them ahead of problem-solving it processes and controls them by dynamically covering and matching their physical or virtual bodies with active recursive holistic code which spreads like waves and viruses. SG can also explicitly implement, simulate or mimic any other existing models and approaches, including the creation of distributed systems with any structures and topologies which appear just automatically on SG implementation levels. SG matured over decades on numerous applications including graph and network theory, computer networking, collective robotics, simulation of battlefields, psychology, sociology, security, disaster management, and the latest as orbital systems. The results and experience obtained, with numerous international publications, seven books included, encouraged the author to deepen and extend the approach for solving much wider classes of problems.

The rest of the paper is organized as follows. Section 2 names extended research areas for the developed technology applications, including understanding the world around us, main environmental problems, global systems, discovering new worlds, and NASA JPL strategic technologies focused on distributed systems, communication, navigation, and robotics. Section 3 describes the basics of the Spatial Grasp Model and Technology including the types of worlds it operates with, Spatial Grasp Language (SGL) with its universal constructs called rules, different types of variables, elementary programming examples, and the language spatial interpretation issues. Section 4 gives examples of how to address the worlds explored in physical and virtual spaces. Section 5 provides examples in SGL of investigation of group behavior of marine animals like finding minimal and maximal distances between sharks, continuous observation of their swarms, distances of movements and extension of the occupied territory, and also speeds of individual sharks. Section 6 gives examples in SGL investigating unknown terrain features like finding heights and coordinates of its hill/mountain summits. Section 7 shows how to solve problems in SGL in large transport networks like finding the quickest way from Start to End points, taking into account road congestions and planned vehicle speed, with subsequent physical movement using the way found. Section 8 investigates large information networks in SGL finding different paths in them, discovering the weakest network components like articulation points, and strongest parts like cliques. Section 9 concludes the paper, and the subsequent References section provides detailed references for all discussed and investigated topics.

The aim of this paper is to investigate extended, up to global, areas for potential application of the developed Spatial Grasp (SG) paradigm and based on it Spatial Grasp Technology (SGT).

2. Research areas of distributed worlds and systems

The following extended areas of advanced distributed technology applications are covered in the research:

The world around us

When we consider the answers to such questions as: What helps us know the world around us? Why studying the world around us is so important? How do we understand the world around us? These and many other questions are addressed in [1].

Major environmental problems

These include pollution, soil degradation, global warming, overpopulation, depletion of natural resources, generating unsustainable waste, waste disposal, deforestation, polar ice caps, loss of biodiversity, climate change, ocean acidification, the nitrogen cycle, ozone layer depletion, acid rain, water pollution, overfishing, urban sprawl, public health issues, genetic engineering, etc. [2].

Global systems

Global system is an economic and political construct in which capital, management, employment, knowledge, natural resources and organizations are fully internationalized. The main determinants of the global system are nation states, multinational corporations, international institutions, and non-governmental organizations. The science of epidemics is one of the successes of Global Systems Science (GSS) – an interdisciplinary approach to modeling the complex, multi-faceted and intertwined problems of the modern world. Another example is the use of network science in financial regulation dealing with many interconnected financial factors. As its name suggests, most of those problems have a global context, but GSS addresses policy issues at all levels – from the individual to local communities to nations to regions. Understanding the main elements of Global Systems Science is one of the goals of the planned research. The details and more data about global systems can be found in [3].

Discovering new worlds

We want to find new planets around other stars which are commonly referred to as exoplanets. When you have a large number of planets, you can start looking for patterns, trends, and hints about the planet-formation process. The primary goal of the search for planets is to understand planet formation and therefore to understand the origins of the solar system. We're trying to learn about their physical characteristics, such as their radii, masses, average densities, and atmospheric properties. For systems of planets, we're interested in how planets interact gravitationally with one another. The exact nature of those gravitational interactions gives us hints about how planetary orbits evolve after they form. And that probably has a lot to do with how architectures of planetary systems eventually come to be. The mentioned above and more in this area can be found in [4, 5].

Strategic technologies of NASA JPL

JPL is a world leader in planetary exploration, Earth science, and space-based astronomy leverages investments in innovative technology development that support the next generation of NASA missions, solving technical and scientific problems of national and international significance [6]. Strategic JPL technologies will let:

– understand how Earth works as a system and how it is changing;
– help pave the way for human exploration of space;
– understand how our Solar System formed and how it is evolving;
– understand how life emerged on Earth and possibly elsewhere in our Solar System;
– understand the diversity of planetary systems in our galaxy;
– understand how the Universe began and how it is evolving;
– use our unique expertise to benefit the nation and planet Earth.

Of particular interest to our investigation are the following areas of JPL activity: distributed systems; communication and navigation; robotics and mobility systems. More on NASA JPL activities can be found in [6].

3. Holistic Spatial Grasp Model and Technology

The main ideas of the patented and extensively published distributed control technology are briefed here, with many other details which can be easily found in [7–15].

3.1. Spatial Grasp Technology basics

Within Spatial Grasp Technology (SGT), a high-level scenario for any task to be performed in a distributed world is represented as an active self-evolving pattern rather than a traditional program, sequential or parallel one. This pattern, written in a high-level Spatial Grasp Language (SGL) and expressing top semantics of the problem to be solved, can start from any point of the world. Then it spatially propagates, replicates, modifies, covers and matches the distributed world in a parallel wave-like mode, while echoing the reached control states and data found or obtained for making decisions at higher levels and further space navigation, as symbolically shown in Fig. 1, see also [7–15].

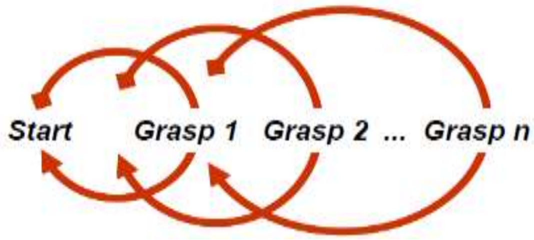


Figure 1 – The main idea of Spatial Grasp Model

3.2. The worlds SGT operates with

SGT allows direct operation with different world representations: Physical World (PW), considered as continuous and infinite where each point can be identified and accessed by physical coordinates; Virtual World (VW) which is discrete and consists of nodes and semantic links between them; and Executive world (EW) consisting of active “doers” with communication possibilities between them. Different kinds of combinations of these worlds can also be possible within the same formalism, like Virtual-Physical World (VPW), Virtual-Execution World (VEW), Execution-Physical World (EPW), and Virtual-Execution-Physical World (VEPW) combining all features of the previous cases. More details on the worlds SGT operates with can be found in [10–15].

3.3. Spatial Grasp Language syntax

The SGL top-level syntax is shown in Fig. 2.

```
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
```

Figure 2 – Basic recursive structure of Spatial Grasp Language

An SGL scenario, called grasp, applied in some point of the distributed space, can just be a constant directly providing the result to be associated with this point. It can be a variable whose content, assigned to it previously when staying in this or (remotely) in another space point (as variables may have non-local meaning and coverage), provides the result in the application point as well. It can also be a rule (expressing certain action, control, description, or context) optionally accompanied with operands separated by comma (if multiple) and embraced in parentheses. These operands can be of any nature and complexity (including arbitrary scenarios themselves) and defined recursively as grasp, i.e. can be constants, variables, or any rules with operands (i.e., as grasps again), and so on, with more details in [14–15]. The full description of the latest SGL versions can be found in [11–15].

3.4. SGL rules

Rules, starting in some world points, can organize navigation of the world sequentially, in parallel, or any combinations thereof. They can result in staying in the same application point or can cause movement to other world points with the obtained results to be left there, as in the final points of the rule. Such results can also be collected, processed, and returned to the starting point of the rule, the latter serving as the final one on this rule. The final world points reached after the rule invocation can themselves become starting ones for other rules. The rules, due to recursive language organization, can form arbitrary operational and control infrastructures expressing any sequential, parallel, hierarchical, centralized, localized, mixed and up to fully decentralized and distributed algorithms. More on the nature and the application of SGL rules can be found in [10–15].

3.5. SGL variables

SGL Variables include Global variables (the most expensive and rarely used ones) which can serve any SGL scenarios and be shared by them, also by their different branches; Heritable variables appearing within a scenario step and serving all subsequent, descendent steps; Frontal variables serving and accompanying the scenario evolution, being transferred between subsequent steps; Environmental variables allowing us to access, analyze, and possibly change different features of physical, virtual and executive worlds during their navigation; and finally, Nodal variables as a property of the world positions reached by scenarios and shared with other scenarios in the same positions. More on SGL variables and examples of their usage are in [10–15].

3.6. Elementary SGL programming examples

- `add(15,7)`

Finds the sum of two values when staying in some world point and leaves the result there. Traditional style can be used as well: `15+7`.

- `assign(Result, add(15,7))`

The found sum of two values is assigned to a variable `Result` which will be staying in the same world point. Using traditional style is allowed as well: `Result = 15+7`.

- `move(`*`x_y`*`)`

From the current world point there is made a physical move to another physical point with the given coordinates.

- `advance(move(`*`x1_y1`*`), move(`*`x2_y2`*`))`

From the current world point there is made a physical move by the coordinates *`x1_y1`*, and then from the reached destination – another move by the coordinates *`x2_y2`*. Using a traditional semicolon as a separator of the operations following one another instead of the rule `advance` can be allowed as well, like: `move(`*`x1_y1`*`); move(`*`x2_y2`*`)`. The resultant control will be associated with the node reached by the second move.

- `create(John)`

There is created an isolated virtual node `John`.

- `hop(John); create(link(+father), node(Peter))`

After hopping into the virtual node `John`, and if it exists, the network is extended with information that `John` is the `father` of `Peter`, by adding this link-node pair to the network. After completion of this two-step scenario, its resultant control will be associated with node `Peter`. The second operation can be written shorter if positions of its operands clearly identify their meanings: `create(+father, Peter)`.

- `move(x_y); repeat(shift(dx_dy); TEMPERATURE > 0)`

Starting from the world point with proper coordinates *x_y*, it organizes a repetitive movement in physical space by the same coordinate shifts *dx_dy* until the `TEMPERATURE` in the reached locations remains above zero. The resultant control will be associated with the latest world point reached which satisfied the temperature condition.

- `output(move(x_y); repeat(shift(dx_dy); TEMPERATURE > 0); WHERE)`

Extends the previous example with printing the coordinates (using environmental variable `WHERE` for lifting them) of the latest positively reached node is physical space, whatever remote it might be, by issuing this result in the position from which this scenario was launched, with final control remaining in it too.

3.7. SGL spatial interpretation

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-scenarios without any central resources or control, as symbolically depicted in Fig. 3. SGL interpreters are named U as universal computational and management nodes which may be stationary or requested and runtime located in proper space points on the demand of SGL scenarios. Details of SGL networked interpretation can be found in [10–15].

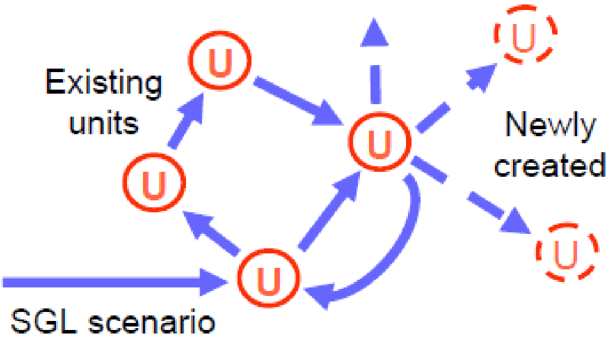


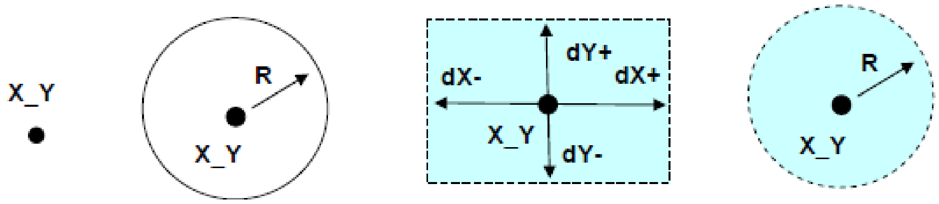
Figure 3 – Parallel evolution of spatial grasp scenarios

4. Addressing spatial areas of investigation

Addressing areas of interest may be in physical and virtual spaces, as well as in their combinations. In physical spaces, addressing and movement may be organized in different dimensions [16–18]. Only a few examples are shown below.

Expressing locations in two-dimensional worlds

There can be the following variants: just a single point with X_Y coordinates as in Fig. 4 *a*; all points are distanced on value R from an X_Y point as in Fig. 4 *b*; all points which may be located in the continuous space starting from X_Y coordinate and limited by dX and dY deviations from as in Fig. 4 *c*; all locations of the region started in X_Y point and restricted by distance R from it as in Fig. 4 *d*.

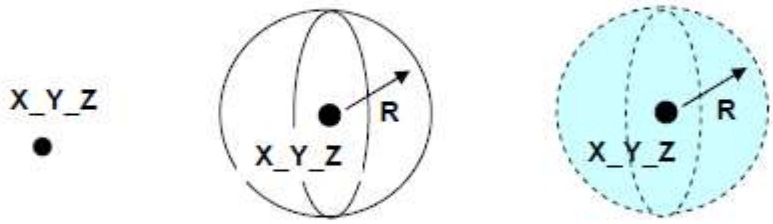


a *b* *c* *d*

Figure 4 – Examples of addressing locations in two-dimensional worlds

Addressing positions in three-dimensional spaces

There can be the following variants: just a single point with X_Y_Z coordinates as in Fig. 5 *a*; all points on the sphere distanced on value R from an X_Y_Z point as in Fig. 5 *b*; all points inside the sphere of radius R with the centre in X_Y_Z point as in Fig. 5 *c*.



a *b* *c*

Figure 5 – Examples of addressing positions in three-dimensional worlds

Addressing in virtual spaces

Different addressing methods can be used as well, including their rich repertoire available on the Internet. Here are some examples:
https://www.google.com/;
https://www.bbc.co.uk/sounds/play/live:bbc_world_service;
https://www.amazon.com/s?k=sapaty+peter&i=stripbooks-intl-ship&ref=nb_sb_noss.

5. Investigation of group behavior of marine animals

This section relates to the investigation of collective behavior and swarming of marine animals [19–25], with some known facts as follows. Basking sharks sometimes congregate in groups of up to 1,400, and swarms of up to over a thousand basking sharks have been spotted along the north east US. Aerial surveys on endangered North Atlantic whales have revealed massive groups of the world's second-largest fish. As big as basking sharks are 32 feet long and can be outsized only by the whale shark. Experts have several theories about why basking sharks congregate. Other shark species are known to gather for feeding, mating, and protection from predators. A record-breaking sighting of about 1,400 sharks in November 2013 included several young sharks, which indicates that the group was likely feeding on zooplankton instead of mating. The study also suggests that the sharks could be gathering to reduce the drag caused by their open mouths during feeding, allowing them to draft off each other to conserve energy.

In the paper, there is being investigated the potential use of SGT for features which include the number of sharks in a swarm, average distance between sharks, maximal territory occupied by the swarm, voice communications frequency, average depth, maximal depth, average speed of the movement, and others. Fig. 6 demonstrates how SGL interpreter modules can be associated with the body of sharks.

The following examples in SGL can provide different research results.

Finding minimal and maximal distance from any shark unit to all other units

```
nodal(Area = ..., Coordinates, Dmin, Dmax);
Coordinates = search(Area);
move_associate(Coordinates); frontal(W) = WHERE;
Dmin = min(hop(all_others); distance(WHERE, W));
Dmax = max(hop(all_others); distance(WHERE, W))
```

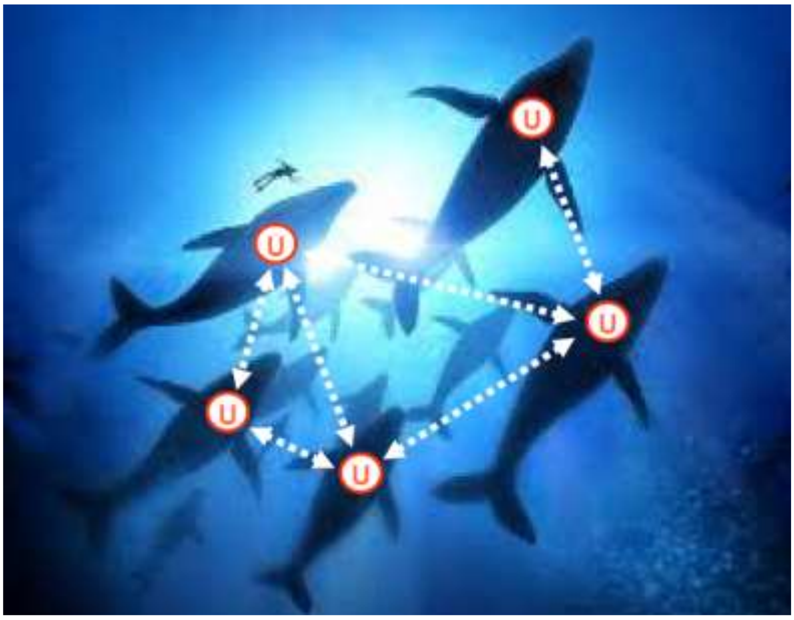


Figure 6 – Investigation of populations in the sea environments

The search of the area (addressed in Section 4) by advanced sensors provides a set of `Coordinates` of the objects of interest. By these coordinates, exploratory units (U) with SGL interpreters are delivered and associated with the objects. The association may be by physical attachment to the objects or just staying close to them and following their movement. Each U tries to contact all other units by sending its coordinates (picked by the environmental variable `WHERE` and delivered in the frontal variable `W`) and finding distances to them (the latter having coordinates in its own `WHERE`). The obtained distances are returned to the starting unit with their minimum assigned to the nodal variable `Dmin`. A similar operation provides maximum distance to other units in `Dmax`.

Continuous finding of the minimal and maximal distances between all shark units

```
nodal(Area = ..., Coordinates, Dmin, Dmax,
      DMIN = infinite; DMAX = 0);
Coordinates = search(Area);
move_associate(Coordinates);
frontal(W = WHERE, DN, DX);
repeat(
  Dmin = min(hop(others); distance(WHERE, W));
  Dmax = max(hop(others); distance(WHERE, W));
  if(Dmin < DMIN, DMIN = Dmin);
  if(Dmax > DMAX, DMAX = Dmax);
  stay(DN = DMIN; DX = DMAX; hop(others);
       if(DN < DMIN, DMIN = DN);
```

```
      if(DX > DMAX, DMAX = DX));
  sleep(delay))
```

This extends the previous scenario to operate in cycling with certain delays between repetitions where each time the obtained `Dmin` and `Dmax` values in units are compared with `DMIN` and `DMAX` accumulating absolute minimal and maximal distance values whenever obtained in these nodes, updating them correspondingly. This means that with time, `DMIN` can only diminish and `DMAX` only increase. But this goes even further where for each contact of other nodes these `DMIN` and `DMAX` values are compared with similar ones at the destinations (using frontal variables `DN` and `DM` for transferring their values), and updating them accordingly. This means that in time, each unit will have the same minimal and maximal values in `DMIN` and `DMAX` in all communications between units, thus providing the global values on the whole swarm.

External request

```
hop(any); output(DMIN, DMAX)
```

By making an external request to any unit of the group after a certain time, we will be able to get these updated globalized `DMIN` and `DMAX` values on the whole group. The external request can also be made cyclic by contacting the same or different units and getting the globalized results gradually improving with each repetition and time.

```
repeat(
  hop(any); output(DMIN, DMAX);
  sleep(delay))
```

The global `DMAX` distance can also be used as an approximate indicator of how large territory was occupied by the whole swarm.

Finding distance covered by a shark unit over time given

```
hop(any);
nodal(W) = WHERE; sleep(time);
output(distance(W, WHERE))
```

This is done by using the coordinates of previous and new locations and calculating the distance between them.

Determining an average speed of a unit over time

```
hop(any);
nodal(W) = WHERE; sleep(time);
output(distance(W, WHERE)/time)
```

This is done by using the coordinates of previous and new locations and the time between the two measurements.

6. Investigation of an unknown terrain

Investigation of qualities of different terrain may include the following methods and operations (with many others found in [26–32]): classifying terrain types, finding the highest points like mountains and hills, tracing valleys and canyons, outlining plain regions and dunes, tracing rivers and their branches, and use of mountain resources. Some related operations are depicted in SGL below, as well as its communicating interpreters navigating the distributed terrain in parallel as in Fig. 7.

Finding heights and coordinates of all found hill/mountain summits

```
nodal(Start_coord) = ...;
frontal(Max_dist = ..., Max_numb = ...);
output(
 move_all(Start_coord);
 repeat(
   Next_coord = find_all(Max_dist, Max_numb, height > WHERE[Z]);
   nonempty(Next_coord);
   move_all(Next_coord); if(occupied, quit));
 append(WHERE[Z], WHERE))
```

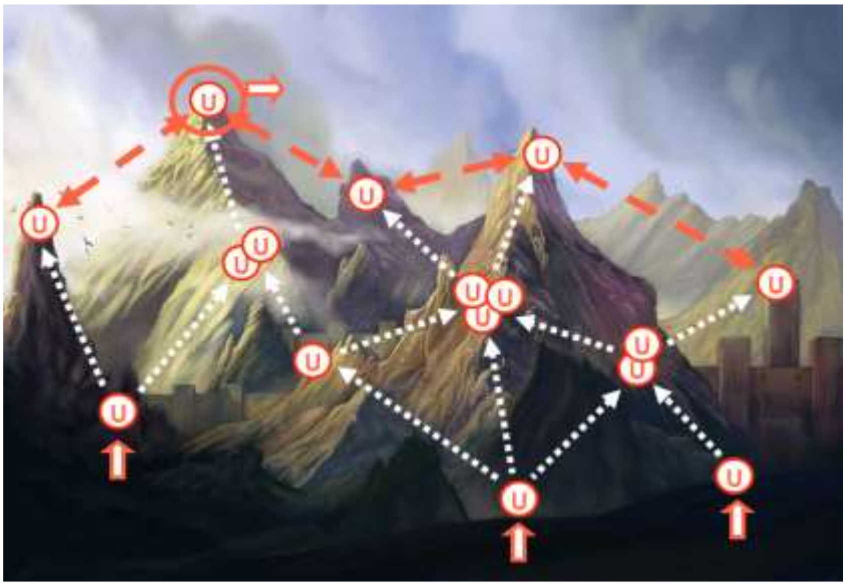


Figure 7 – Investigation of a mountainous terrain

The terrain investigation may be complicated by restricted visibility like in bad weather or on other planets with unknown conditions. The scenario for the chosen set of starting coordinates `Start_coord` launches into all of them exploratory units U which repeatedly try to move to new higher points by finding proper next step locations with the distance limited by the capability of their sensors (using the max-distance threshold `Max_dist`). If there are discovered more than one of such new points, the units can be replicated (if it is technically achievable with a threshold number given) with quitting of activity if the new locations are already `occupied` by other units. This repetitive parallel ascending process terminates if no higher locations are found by different branches, replying to the external system that launched this scenario of heights `WHERE[Z]` and coordinates `WHERE` of all summits found while publishing the results obtained by the rule `output`.

Finding the height and coordinates of the highest hill/mountain

```
Start_coord = ...;
frontal(Max_dist = ..., Max_numb = ...);
output(
 max(
  move_all(Start_coord);
  repeat(
    Next_coord = find_all(Max_dist, Max_numb, hight > WHERE[Z]);
    nonempty(Next_coord);
```

```
    move_all(Next_coord); if(occupied, quit));
  append(WHERE[Z], WHERE))))
```

With slight changes, finding height and coordinates only of the highest hill/mountain can be done by the additional embracing of the whole scenario before the publication of its results with the feedback-based rule `max`.

7. Investigation of large transport networks

We are also investigating the organization of large transport networks [33–37] and the possibility of solving different problems in them under SGT. These may include traffic congestion and parking difficulties, longer commuting, accidents and safety, fuel efficiency, pollution and noise, and safety. A related example in SGL may be as follows, with communicating SGL interpreters preliminary installed in different road junctions or capable of moving on the roads themselves, either physically or virtually, as symbolically shown in Fig. 8. The given below scenario of finding the quickest way from Start to End points in the transport network takes into account the current road loads and congestions, the expected speed of a vehicle for which the solution is to be found, and the restricted physical area for finding the solution.

Figure 8 – Investigation of city transport systems

Finding the quickest way from Start to End locations

```
frontal(Reached, Route, Area = ..., Speed = ...);
nodal(Start = ..., End = ..., Time = infinite, Path);
sequence(
  (hop(Start; Route = NAME;
   repeat(hop_noback(all_roads); belong(WHERE, Area);
          increment(Reached, duration(LINK, load, Speed));
          Reached < Time; Time = Reached;
          Route = append(Route, NAME); Path = Route)),
  output(hop(End); Path, Time))
```

Originating in the `Start` node, the scenario spreads through all possible links-roads within the declared `Area` while bringing into each junction node the `Reached` time (increased at each link by taking into account the `LINK` details, its current *load*, and expected vehicle `Speed`). When `Reached` appears to be lower than the previously recorded `Time` from the `Start`, it substitutes the `Time` value in nodes. While driving on the road, the scenario also accumulates a possible `Route` consisting of names of passed junctions. It is fixed in nodes as

Path if the current arrival to them has a lesser value in the frontal variable Reached than the previously recorded in Time. This global process originated in Start and spreading through links-road with the correcting solutions to nodes each time they are reentered with better parameters will terminate where each node will register the best Time of reaching it and also the best Path to it from Start. After this, by directly contacting the End node (as well as any other node), it is possible to directly get the best values of Path and Time in it from the Start. More details and other variants of finding shortest paths in distributed networks can be found elsewhere, including [8–13].

Driving a car using the way found

```
move(Start); repeat(move(withdraw(Path)))
```

Using the Path found from Start to End (the latter can be any network node) and picked up at the End node, now it is possible to organize stepwise driving a car, departing from Start and each time withdrawing from Path the next junction name/address to be reached, unless Path becomes empty.

8. Investigation of information networks

Investigation of different kinds of information networks [38–42] is made to have the possibility of solving problems in them in SGL. These may include high bandwidth usage, high CPU usage, DNS issues, physical connectivity issues, malfunctioning devices or equipment, interference in the wireless network, and many others. Some examples of solutions in SGL and their different variants and detailed explanations can also be found in [8–15]. Into account there should be taken that communicating SGL interpreters can be preliminarily installed in network nodes, they are also capable of moving themselves freely through the network links and dynamically associating with different nodes (Fig. 9).

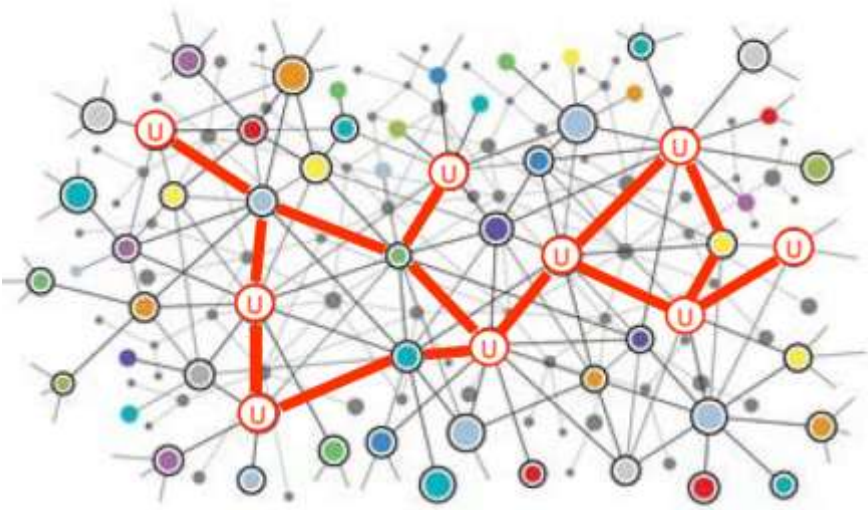


Figure 9 – Solving problems in large data networks

Finding all simple paths from Start to End nodes under certain conditions

```
nodal(Start) = ...;
frontal(End = ..., Path, Length, Forbidden = (r1, r3),
        Threshold = ...);
hop(Start);
output_repeat(
```

```
append(NAME, Path);
increment(Length); Length <= Threshold;
if(NAME == End, blind_unit(Path));
hop_first(links_notbelong(Forbidden)))
```

The scenario additionally provides conditions that these paths should not exceed their `Length` by the given `Threshold` and they should not go through the `Forbidden` types of links. The growth of the paths, which are gradually accumulated in mobile `Path` variables spreading together with parallel network navigation, terminates upon reaching the `End` node. After that all found paths are returned to the `Start` node and printed there as independent units.

Finding articulation points

```
hop(all_nodes); COLOR = NAME;
and((hop_link(random); repeat(hop_first(all_links)),
     hop_first(all_links),
     output(NAME))
```

These are the weakest network components which, when deleted, split the network into disconnected parts. The scenario, starting in all network nodes, first goes by any single link to a neighboring node and marks the whole part of the network starting from it with a special `COLOR`. Then it tries to move through all other links from the same starting node and see if at least one node reached is not marked with the same color. That will indicate that this starting node separates the whole network into disconnected parts, and its name will be printed. The same procedure is organized for all the nodes, each performing it with its own color which is used here as this node's `NAME`.

Finding maximal cliques

```
hop(all_nodes);
COLOR = NAME; frontal(Clique = NAME, Threshold = 3);
repeat(hop_first(all_links);
       if(and_parallel(hop(any_link, all_nodes(Clique))),
          if(BACK > NAME, append(Clique, NAME), done),
          fail));
count(Clique) >= Threshold; output(Clique)
```

These are the strongest network parts represented as full graphs with any node connected to any other one. The scenario finding all cliques in the network having the number of nodes equal or higher than the given `Threshold` is operating as follows. Starting from all the nodes with a unique `COLOR` (as the node's `NAME`), it spreads through the links to other nodes and adds each new node to the frontal variable `Clique` if it has a connection with all previous clique nodes until this becomes possible, then outputs the obtained clique in `Clique`. To avoid duplicates, which can appear as any clique can be independently formed from all its nodes, the clique formation can only be continued in a unique order by comparing the value of a new node in `NAME` with the value of the previous node in environmental variable `BACK` (thus using for comparison just node names), immediately terminating otherwise.

9. Conclusions

There have been outlined the possible extended applications of Spatial Grasp Technology, briefed its latest version with the basic Spatial Grasp Language and its networked implementation. The communicating language interpreters of the software or hardware nature can be dynamically installed in thousands, millions, and even billions of copies worldwide forming altogether powerful and self-organized spatial machines capable of working without any central resources. Exemplary solutions to practical problems in different worlds have been shown in SGL which can work in both terrestrial and celestial environments like, for example, the investigation of an unknown terrain on other planets or the collective behavior of previously unknown creatures there. Regardless of the nature of the worlds considered and tasks to be solved in them, there is used the same universal approach of invading, covering, and matching these worlds, and, if needed, massively and cooperatively staying and moving in them in a highly coordinated parallel and distributed mode while investigating their problems and offering clear solutions. Such solutions may also be considered at much higher levels than usual like, for example, the ones based on and providing a sort of global awareness and even consciousness. These were already analyzed and discussed in our previous paper, as in [15], for which the current one continues and extends research areas for the paradigm and technology developed. A new book on this spatial model is being prepared as well as a sequel to the previous seven books, as well as a new patent on the latest technology version with so far unpublished details of its efficient software or hardware implementation.

REFERENCES

1. What help us to know the world around us. *FAQ-ANS*. 2021. August 13. URL: https://faq-ans.com/en/Q%26A/page=ef9b919ba475fd24194141634399321e.

2. 20 Major Current Environmental Problems. URL: https://www.conserve-energy-future.com/ 15-current-environmental-problems.php.

3. Dum R., Johnson J. Global Systems Science and Policy. Research Gate. 2016. URL: https://www.researchgate.net/publication/302896479_Global_Systems_Science_and_Policy.

4. Woo M. Discovering New Worlds. Caltech Media Relations. 2010. October 20. URL: https://www.caltech.edu/about/news/discovering-new-worlds-1811.

5. New Worlds, New Horizons in Astronomy and Astrophysics, Panel reports. The National Academies Press. Washington, DC, 2010. URL: https://www.nap.edu/catalog/12982/panel-reports-new-worlds-new-horizons-in-astronomy-and-astrophysics.

6. Strategic Technologies, Jet Propulsion Laboratory. 2019. URL: https://scienceandtechnology.jpl.nasa.gov/sites/default/files/documents/JPL_Strategic_Technologies_2019.pdf/.

7. Sapaty P.S. A distributed processing system: European Patent N 0389655, Publ. 10.11.93. 35 p.

8. Sapaty P.S. Mobile Processing in Distributed and Open Environments. New York: John Wiley & Sons, 1999. 410 p.

9. Sapaty P.S. Ruling Distributed Dynamic Worlds. New York: John Wiley & Sons, 2005. 255 p.

10. Sapaty P.S. Managing Distributed Dynamic Systems with Spatial Grasp Technology, Springer, 2017. 284 p.

11. Sapaty P.S. Holistic Analysis and Management of Distributed Social Systems, Springer, 2018. 234 p.

12. Sapaty P.S. Complexity in International Security: A Holistic Spatial Approach. Emerald Publishing, 2019. 160 p.

13. Sapaty P.S. Symbiosis of Real and Simulated Worlds under Spatial Grasp Technology. Springer, 2021. 251 p.

14. Sapaty P.S. Spatial Grasp as a Model for Space-based Control and Management Systems. CRC Press, 2022. 280 p.

15. Sapaty P.S. Comprehending Distributed Worlds with the Spatial Grasp Paradigm. *Mathematical machines and systems*. 2022. N 1. URL: http://www.immsp.kiev.ua/publications/articles/2022/2022_1/01_22_Sapaty.pdf.

16. Freeman W.H. Motion in Two and Three Dimensions. 2014. URL: https://bcs.whfreeman.com/WebPub/Physics/TiplerPhysics6e/reprint-PDFs/Tipler_Physics6e _Chapter_3.pdf.

17. Motion in Two or Three Dimensions, Pearson Education Inc., publishing as Pearson Addison-Wesley. 2008. URL: https://people.tamu.edu/~mahapatra/teaching/ch3.pdf.

18. Six-dimensional space. URL: https://en.wikipedia.org/wiki/Six-dimensional_space.

19. Canizo J.A., Carrillo J.A., Rosado J. Collective Behavior of Animals: Swarming and Complex Patterns. ReserachGate. 2010. December. URL: https://www.researchgate.net/publication/49615753_Collective_Behavior_of_Animals_Swarming_and_Complex_Patterns.
20. Swarms of Huge Sharks Discovered, Baffling Experts. *WEIRD & WILD*. 2018. April 12. URL: https://www.nationalgeographic.com/animals/article/animals-sharks-swarms-oceans.
21. Kuhn A., Erni D., Denke M. Empowering Collections with Swarm Behavior. University of Bern, 2010. July 1. URL: https://arxiv.org/pdf/1007.0159.pdf.
22. Ioannou A.C. Swarm intelligence in fish? The difficulty in demonstrating distributed and self-organized collective intelligence in (some) animal groups. *Behavioral Processes*. 2017. Vol. 141, Part 2. August. P. 141–151. URL: https://www.sciencedirect.com/ science/article/ pii/ S0376635716302881.
23. Acevedo-Gutiérrez A. Group Behavior, Encyclopedia of Marine Mammals. 2009. URL: https://www.wwu.edu/faculty/aceveda/PDFs/lab%20papers/Acevedo%2018%20group%20behavior.pdf.
24. Rasmussen A. Do Whales live in Groups? How do Groups of Whales Behave? *AZ Animals*. 2021. November 23. URL: https://a-z-animals.com/blog/do-whales-live-in-groups-how-do-groups-of-whales-behave/.
25. Killer Whale Social Structure. URL: https://www.killer-whale.org/killer-whale-social-structure/.
26. Unit B1m – Problems in the Mountain Environment, International Environment Forum A Bahá'í inspired organization for environment and sustainability. URL: https://iefworld.org/remb1m.htm.
27. Peniak M., Marocco D., Cangelosi A. Autonomous Robot Exploration of Unknown Terrain: A Preliminary Model of Mars Rover Robot. 2008. URL: http://robotics.estec.esa.int/ASTRA/Astra2008/S10/10_05_Peniak.pdf.
28. Koenig S., Likhachev M. Fast Replanning for Navigation in Unknown Terrain. *IEEE Transaction on Robotics*. 2005. Vol. 21, N 3. URL: https://www.cs.cmu.edu/~maxim/files/ dlite_tro05.pdf.
29. Flight in Mountainous Terrain. URL: https://skybrary.aero/articles/flight-mountainous-terrain.
30. Vexinath A., Dzierzak L. 15 Different Types of Terrain You Should Know About: Complete Guide. 2022. February 19. URL: https://outforia.com/types-of-terrain/.
31. Bhardwaj A. Investigating the Terrain Complexity from ATL06 ICESat-2 Data for Terrain Elevation and Its Use for Assessment of Openly Accessible InSAR Based DEMs in Parts of Himalaya's. Eng. Proc. 2021. November 1. URL: https://www.mdpi.com/2673-4591/10/1/65.
32. Wieder W.R., Knowles J.F., Blanken P.D., Swenson S.C., Suding K.N. Ecosystem function in complex mountain terrain: Combining models and long-term observations to advance process-based understanding. *Geophysical Research: Biogeosciences*. 2017. Vol. 122 (4). P. 825–845. URL: https://www.researchgate.net/publication/315636071_Ecosystem_function_in_complex_mountain_terrain_Combining_models_and_long-term_observations_to_advance_process based _understanding.
33. Rodrigue J.-P. Urban Transport Challenges, The Geography of Transport Systems. URL: https://transportgeography.org/contents/chapter8/urban-transport-challenges/.
34. Zavitsas K., Kaparias I., Bell M.G.H. Transport problems in cities. CONDUITS Consortium. Imperial College London, 2010. URL: https://trimis.ec.europa.eu/sites/default/files/project/documents/20120402_173932_45110_D%201.1%20-%20Transport%20problems%20in%20cities%20-%20v3.pdf.
35. Urban transportation systems of 25 global cities. McKinsey&Company. 2021. July. URL: https://www.mckinsey.com/~/media/mckinsey/business%20functions/operations/our%20insights/bulding%20a%20transport%20system%20that%20works%20new%20charts%20five%20insights%20from%20our%2025%20city%20report%20new/elements-of-success-urban-transportation-systems-of-25-global-cities-july-2021.pdf.
36. Chenel T., Moynihan Q. 15 cities that are home to some of the world's best transport solutions. I*NSIDER*. 2021. August 20. URL: https://www.businessinsider.com/15-cities-worlds-best-transport-solutions-2019-12.
37. 5 Transportation Issues And Challenges Facing US Cities And Their Solutions. REMIX PRIMER SERIES. 2021. March 23. URL: https://www.remix.com/blog/5-transportation-challenges-facing-us-cities.
38. Lamberti A. 6 Most Common Network Problems. Obkio. 2021. August 2. URL: https://obkio.com/blog/common-network-problems/.
39. Fang Y., Wang K., Lin X., Zhang W. Cohesive Subgraph Search over Big Heterogeneous Information Networks: Applications, Challenges and Solutions. *SIGMOD/PODS '21*: proc. of the 2021 International Conference on Management of Data. 2021. June. URL: https://dl.acm.org/doi/10.1145/3448016.3457538.

40. Kleinberg J. Algorithms for Information Networks. Carnegie Mellon. Spring 2005. URL: http://www.cs.cornell.edu/home/kleinber/sp05course.html.
41. Ahern K.R. Information networks: Evidence from illegal insider trading tips. *Journal of Financial Economics*. 2017. Vol. 125, Issue 1. P. 26–47. URL: https://www.sciencedirect.com/science/article/abs/pii/S0304405X17300570.
42. 10 Network Performance Monitoring (NPM) Tools You Shouldn't Miss. *TEK-TOOLS*. 2020. June 3. URL: https://www.tek-tools.com/network/free-network-monitoring-tools.

Стаття надійшла до редакції 04.04.2022