

Towards Massively Robotized Systems under Spatial Grasp Technology

Peter Simon Sapaty*

Institute of Mathematical Machines and Systems, National Academy of Sciences of Ukraine, Kiev, Ukraine

Abstract

The presentation describes organization of large distributed robotized systems with the help of high level networking ideology and technology allowing for integral, holistic solutions of complex problems in rapidly changing and unpredictable environments. Exemplary scenarios in Spatial Grasp Language for practical tasks oriented on the use of multiple cooperative robotic units operating under unified control provided by the technology are also exhibited and discussed.

Keywords: Self-evolving patterns; Spatial grasp technology; Networked language interpretation; Massively robotized systems; Integral collective solutions; Spatial intelligence; Distributed operational infrastructures

Introduction

Large numbers of robotic facilities have been accumulated worldwide in practically any area of human activity, and their further development is taking place with increased financing and speed in both civil and military areas. But in many cases they still remain as specialized devices rather than intelligent collaborators for humans capable of substituting them if needed, with dissimilar, often proprietary, architectures and specific management and control.

To implant seamlessly advanced robotics into human societies at even greater scale, especially in critical areas, we must develop and use much more general and universal approaches to their tasking and autonomous decision-making, also effective integration with human command and control infrastructures.

The paper describes high level formalism and technology, already prototyped and tested on numerous applications in distributed networked systems, that can express operations and top decisions in both physical and virtual environments regardless of who (humans) or what (robots) should perform them, and in which quantities. This can allow us to make implementation in dynamic and complex environments where manned and/or unmanned resources are determined not a priori but rather at runtime, depending on circumstances.

Spatial Grasp Technology (SGT)

With existing sufficient publications related to SGT including, for example [1-6], we will be concentrating here on its use for solutions related to cooperative work of multiple robotic units, where large unmanned teams can be organized in the form of global-goal-driven holistic and self-organizing structures rather than traditional loose collections often referred to as swarms [7].

Within SGT, a high-level scenario for any task in a distributed world is represented as an active self-evolving pattern rather than traditional program, sequential or parallel, by inheriting holistic and gestalt [8-10] ideas rather than those of communicating agents [11].

This pattern, expressing top semantics and key decisions of the problem to be solved, starting from any world point, spatially propagates, grows, replicates, modifies, covers, interlinks and matches the world. It creates distributed operational infrastructures throughout the space covered, with the final results retained in the environment,

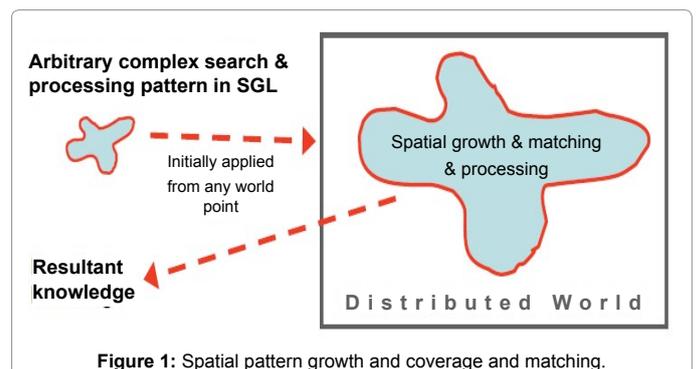
where they were obtained, or returned as a high level knowledge to the scenario starting point (Figure 1).

The technology practically operates as follows. A network of universal control modules U embedded into key system points (humans, robots, sensors, mobile phones, etc.) collectively interprets high-level mission scenarios in Spatial Grasp Language (SGL). Capable of representing any parallel and distributed algorithms, these scenarios can start from any node, covering at runtime the whole system or its parts needed with operations and control (Figure 2).

The spreading scenarios can create knowledge infrastructures arbitrarily distributed between system components (humans, robots, sensors). Navigated by same or other scenarios, these can effectively support distributed databases, command and control (C2), situation awareness, and autonomous decisions. Also simulate any other existing or hypothetical computational and/or control models.

Spatial grasp language

SGL allows us to directly move through, observe, and make any actions and decisions in fully distributed environments (whether physical, virtual, executive or combined). It has universal recursive



*Corresponding author: Peter Simon Sapaty, Institute of Mathematical Machines and Systems, National Academy of Sciences of Ukraine, Kiev, Ukraine, Tel: 380-67-4199223; E-mail: peter.sapaty@gmail.com

Received February 15, 2016; Accepted February 25, 2016; Published February 29, 2016

Citation: Sapaty PS (2016) Towards Massively Robotized Systems under Spatial Grasp Technology. J Comput Sci Syst Biol 9: 069-075. doi:10.4172/jcsb.1000223

Copyright: © 2016 Sapaty PS. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

structure (Figure 3) capable of representing any parallel and distributed algorithms operating over spatially scattered data or other, lower level, distributed systems of arbitrary natures. An SGL scenario develops as parallel transition between sets of progress points, which may be dynamically associated with different physical or virtual locations in distributed worlds.

To conventionalize/simplify SGL programs, traditional to existing programming languages abbreviations of operations also delimiters can sometimes be used too. With the presence of such deviations, the scenario text can be easily adjusted to the SGL standards by a preprocessing converter.

SGL interpreter

The interpreter consists of a number of specialized modules handling and sharing specific data structures. A backbone of the distributed interpreter is its spatial track system providing global awareness and automatic C2 over multiple distributed processes, also creating and managing (including removing when becoming not needed) the distributed information and control resources. The distributed SGL interpreter may have any number of nodes, up to millions or even billions and distributed worldwide. It can operate with dynamic and changeable topology with the number of interacting nodes varying during the scenario evolution, with many scenarios (injected from different nodes) capable of operating at the same time and in the same networked space, sharing common spatial resources.

Elementary Examples in SGL

We are showing here a few elementary SGL scenarios for operations

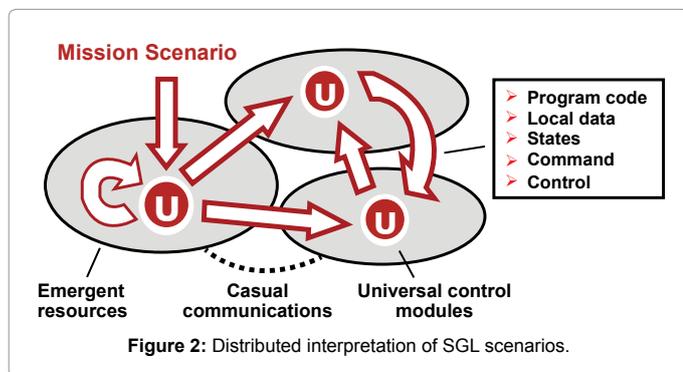


Figure 2: Distributed interpretation of SGL scenarios.

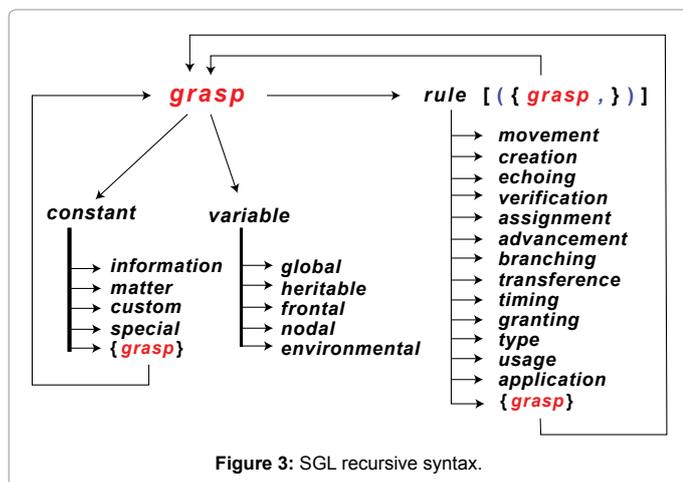


Figure 3: SGL recursive syntax.

in different worlds (physical, virtual, executive and combined, also in a purely computational one), where the same universal syntax and semantics are used. The items in italics are to be substituted by concrete parameters (same can be said for SGL scenarios in the subsequent sections).

(A) Pure computation: Assigning the sum of constants to a variable:

assign (Result, add (21, 33, 99.8)) or

Result = 21 + 33 + 99.8

This scenario can be applied in any world position, with the results retained in the existing variable there (or newly created on the first assignment if initially absent), with the resultant control remaining in the same position.

(B) Physical space: Independent moves in physical space by two coordinates:

move ((x1, y3), (x5, y8)) or

move (x1_y3, x5_y8)

The scenario applied in any world position will cause independent and simultaneous physical movement to the two new positions, with the resultant (split) control retained in the two destinations reached. The rest of the scenario (if any, here absent) would replicate and continue from both positions in parallel.

(C) Virtual space: Create node *Peter* followed by link-node pair stating that he is *father of Alex*:

create (Peter, (+fatherof, Alex)) or

create (Peter, +fatherof_Alex)

The scenario, starting from any world node, will first create isolated node *Peter*, move to it, and then created the named link to node *Alex*, which will be created too. The resultant control will move to node *Alex*, and the rest of the scenario (here absent) will start from this latter virtual node.

(D) Executive space: Ordering *robot* to *extinguish* fire by given coordinates:

hop (robot); extinguish (x_y)

The scenario will first enter the executive node *robot* (we could also give it a particular name), and then order it to activate the embedded procedure *extinguish*, passing the given coordinates to it. After completion of operation, the resultant control will remain in the node *robot*, and the rest of the scenario, if any, would start and continue from this node.

(E) Combined physical-virtual-executive space: Order *John* to find where father of *Alex* is staying, move to this place physically, and alarm anybody there:

hop (John);

move(hop(Alex); follow(-fatherof); WHERE);

alarm(any)

The scenario first enters the executive node *John* and then hops to the existing virtual node *Alex* (say, found in a database or internet), passes the link *fatherof* against its orientation, and enters the virtual node reached (supposed to be *Peter*). The scenario then lifts *Peter's* physical coordinates by using environmental variable *WHERE* (if

Peter's location is registered somewhere, not shown this in the case *c* above). It then returns the obtained coordinates to the starting node *John* who moves by them physically and alarms everybody staying there (say, on a danger of Peter, or in order to find him, which may be detailed in the scenario). The rest of the scenario, if any, will continue from the physical position reached by *John*.

Unified Transition to Unmanned Systems

SGT provides different benefits on the way to obtaining massively robotized up to fully unmanned systems in this century, with only some of them mentioned below.

- It allows us to describe missions in distributed spaces on a variety of levels—from top semantic omitting details of implementations to fully detailed ones, if needed for efficiency. The highly integral, possibly, mixed mission descriptions written in the same universal formalism have no seams between levels, which makes them very compact (up to order of magnitude shorter than in other technologies) and capable of being quickly updated or fully rewritten at runtime.
- The SGL-presented missions are well understandable to human and robotic components, which both operate under the same automatic control resulting from distributed SGL interpretation, where human components of the systems, partially or the whole can be substituted by robotic ones and vice versa, at runtime, without interrupting the mission.
- Concerning massive use of robotic components, SGT with SGL has no limits to the number of their elements or communication structure in between, where the latter may be unsafe and runtime changeable. Fully interpreted self-spreading virus-like scenarios in SGL can keep the whole robotic society as an integral global-goal-driven unit, which due to high internal organization can be externally controlled even by a single remote human operator.
- The 21st century is believed to have a drastically increasing activity in exploration of both Earth and beyond, also growing demands to local and global security, which will require advanced ground, air, and space operations. Many will need to be distributed, cooperative, flexible, self-recovering, global-goal-oriented, automated up to fully automatic, with massive use of unmanned components. SGL can cover the full spectrum of such activities, within the same simple space conquering formalism.
- There may be very critical factors of organizing massively robotic solutions in peculiar spaces, like underwater or in open cosmos, with very low communication bandwidth for the former and great distance for the latter. In both cases SGT can offer the best possible solutions having semantic level, intelligent organization of the whole system resulting in dramatic reduction of communications demands between system components.

Some practical examples of application of collective robotics expressed in SGL are given in the following three sections, whereas multiple robotics applications in SGL can be found in existing publications [12-15].

Coastal Waters Cooperative Patrol

The scenario is as follows. A number of coastal patrol vehicles, which may be surface or underwater, are patrolling simultaneously the coastline, following it and regularly reporting of what they see or discover (sensors dependent) in key points. Let us assume that only two

vehicles are engaged for simplicity (let these be underwater ones named UUV1 and UUV2, as in (Figure 4), which start from the opposite ends, each following the whole route but in different directions.

(a) Coastal map creation: At the beginning we should create the discrete coastal map as a semantic network consisting of coordinates of key points linked with each other by oriented links treating the network as a directed chain. One vehicle should follow this chain along orientation and return against, whereas the second vehicle doing this too, but vice versa, as in Figure 4.

To create this semantic network, the following scenario is sufficient (all links assumed having same name *r*):

```
create (x1_y1; (+r, x2_y2); ...; (+r, x9_y9))
```

This network can be stored in different ways, with all its nodes in a single world point up to each node in a separate point, the latter, say, corresponding to the physical points of Figure 4, assuming they can communicate with each other. The following are two possible examples of their simultaneous operation.

(b) Solution 1: Vehicles moving forward and backward independently by the coastal map, avoiding collisions with each other (the avoidance mechanisms to be present at implementation levels), and report what they see/sense when pass the key points $x_i_y_i$ with a given *vision_depth*:

```
branch(
(hop (x1_y1); R = +r),
(hop (x9_y9); R = -r));
WHERE = CONTENT;
repeat(
repeat (check_report (vision_depth);
WHERE = follow (R));
Invert (R))
```

Each vehicle is synchronously moving in both virtual and physical spaces, with the next key point physical coordinates picked up from the corresponding semantic network nodes. Having reached the end of the virtual chain, they begin moving in the opposite direction through it unless reach the opposite end, and so on.

(c) Solution 2: This solution differs from the previous one in that each vehicle turns back when discovers another vehicle in a given, threshold *distance*, which means that the path ahead has been checked already (or will be) by the "colleague":

```
branch(
(hop (x1_y1); R = +r),
```

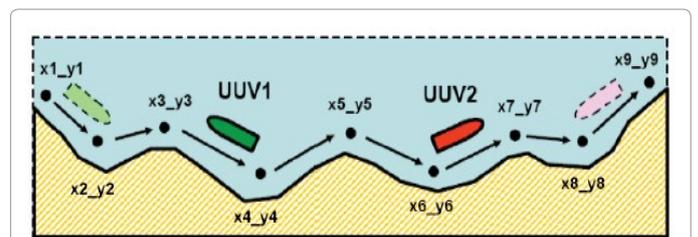


Figure 4: Simultaneous coastal patrol.

```
(hop (x9_y9); R = -r));
WHERE = CONTENT;
repeat (
repeat (check_report (vision_depth);
WHERE = follow (R); none (distance));
invert (R))
```

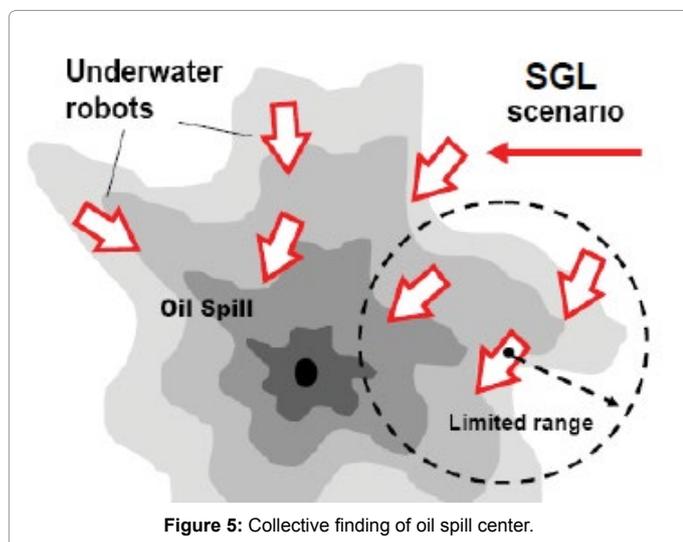
The solutions above can be extended to any number of robotic units operative cooperatively for this and also much more complex scenarios.

Cooperative Finding of Oil Spill Center

This scenario describes the situation where a region of the sea, or *Expected_spill_area* (may be both surface and submersed) is polluted by the evolving oil spill, and the group of communicating surface or underwater (or both) robots, first delivered to the approximate region, tries to detect the spill center cooperatively, reporting the finding, as in Figure 5.

The following SGL scenario starts from any robot, covering eventually the whole group via communications between the units. Each robot tries to move stepwise toward increased spill level in a randomized way (by *move_range*), otherwise returns to the previous position if the move was unsuccessful. Each movement is allowed only if other robots remain in reach (by *communication_range*), to maintain group's connectivity as a system.

```
hop (all);
move (random, Expected_spill_area);
nodal (Level, Direction, Current, Count, Max);
Level = check (spill);
branch (
loop(
Max = maximum (
(hop (communication_range, all); Max),
Max, Level);
```



```
If (Level == Max, increment (Count));
if (Count >= tests_threshold_number,
(output ('Center:', WHERE); abort));
Sleep (Delay),
loop(
Current = WHERE;
or ((WHERE += random (move_range, Direction);
nonempty (hop (communication_range, all));
New = check (spill) > Level;
Level = New; Count = 0;
Direction = angle (Current, WHERE)),
WHERE = Current)))
```

By regular local communications between robots, the currently maximum spill level discovered by the group as a whole is constantly updated and maintained in each robot. And the robot(s) where this maximum level corresponds to the local spill level after a threshold delay (expressed by a number of repeated tests, or *tests_threshold_number*) report the spill's center (which may be submersed too).

Robotic Swarm Operation under SGT

This scenario relates to the use of a large unmanned swarm for highly dynamic distributed operations, where robotic units are supposed to be naval, surface ones, and the operation is a hypothetical swarm attack on an adversary's fleet. There are also real developments in this direction, with recent tests on Virginia's James River (Figure 6) of a swarm of autonomous boats designed to overwhelm enemies [16], where boats operated without direct human control.

SGL can describe and provide goal-driven behaviour of the robotic swarm which can operate autonomously and without any central resources. The initial stage of our scenario is depicted in Figure 7, with the swarm staying outside the expected attack area.

The following SGL solution can start from any robot, covering the whole swarm and causing units to move stepwise to the area by *expected_area_coordinates*, with each step being planned using *next_step_guidelines* and *allowed_distance_between_units*.

```
hop (direct, all);
nodal (
Area = expected_area_coordinates,
Guidelines = next_step_guidelines,
Range1 = allowed_distance_between_units,
```



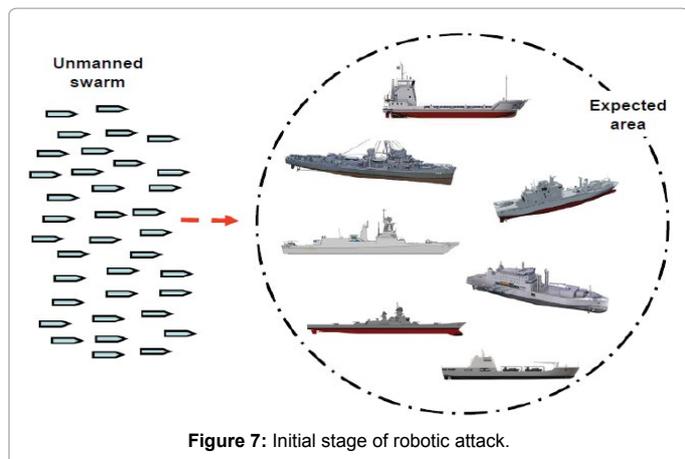


Figure 7: Initial stage of robotic attack.

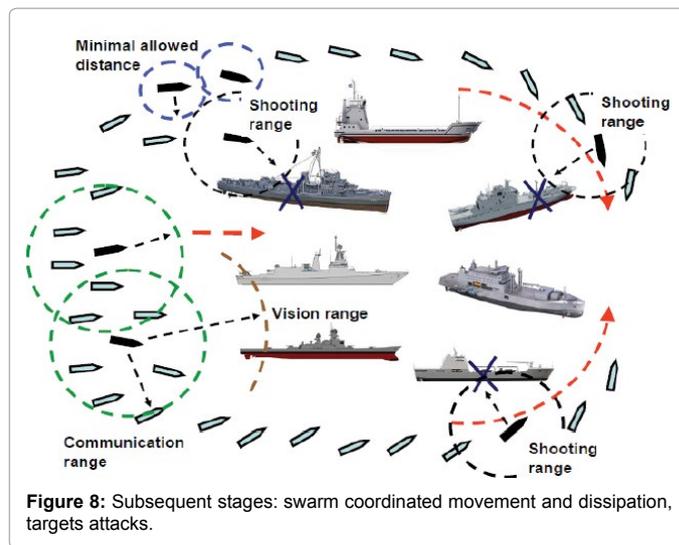


Figure 8: Subsequent stages: swarm coordinated movement and dissipation, targets attacks.

```

Range2 = vision_range,
Range3 = shooting_range,
Range4 = communication_range,
Offer, Seen);
frontal (Transit);
sling(
Offer = next (WHERE, Area, Guidelines);
if (none (center (Offer), Range1),
WHERE += Offer);
branch(
(correct (Area, detect (Range2, all));
Transit = Area; hop (Range4, all);
Update (Area, Transit)),
detect_impact (targets, Range3))

```

Each unit constantly updates the initially given approximate region coordinates by what it currently sees within the *vision_range*, also regularly sharing its own improved area version with the neighboring robotic units staying within *communication_range* (regularly shared from the neighbors too, and thus throughout the entire swarm).

The whole swarm in such a cooperative way is gradually improving global image of the expected area, and in each robot. With the continuously improving area coordinates the next movement steps are planned towards surrounding of the area rather than, say, moving in the direction of its center, in order to prevent possible vessels' escape from the attack.

Independently from this collective move towards the vessels located within the attack area and nearing them (with area coordinates updates resulting, in turn, in the updates of the very moves) another process is trying to detect the adversary's units within the *shooting_range* of the robotic units and attack these targets. A possible development snapshot for this scenario is shown in Figure 8.

Other Applications

The following are only some of the researched, discussed, and reported applications of SGT and SGL summarizing their advantages.

Graph and network theory: Highly parallel and fully distributed solutions of main graph and network problems have been programmed in earlier versions of SGL and demonstrated in distributed computer networks. This confirmed effectiveness of the spatial grasp ideology and paradigm to solve practically any problems linked with complex structures and infrastructures distributed over large spaces and having to operate without any central resources [1,2].

Distributed interactive simulation: The technology had been researched for both live control of large dynamic systems like battlefields and distributed interactive simulation of them (the latter serving as a look-ahead to the former), also any combination thereof, with watershed between the two changing at runtime. The approach was used as part of the international Distributed Interactive Simulation (DIS) project headquartered in Orlando, Florida [17-19].

Solving social problems: SGL appeared to be useful in social systems area, like support of elderly and handicapped people in modern societies, from intelligent homes improving everyday life to tracing, following, and protecting them in complex urban environments, automatically activating existing health support facilities in life threatening situations [12,20,21].

Human terrain: SGT allows this new topic, originally coined in military, to be considered and used in a much broader sense and scale than initially planned, allowing us to solve complex national and international conflicts and problems by intelligent and predominantly peaceful means, while fully obeying the existing ethical standards [22,23].

Sensor networks: Multiple sensors scattered over large territories can behave altogether under SGT as a spatial supercomputer operating under local, unsafe, and restricted communications, also without any central control, but effectively pursuing global goals, where individual sensor units can be both stationary and mobile [24-26].

Intelligence, surveillance and reconnaissance (ISR): SGT can integrate distributed ISR facilities into flexible goal-driven systems operating under unified command and control, which can be automatic. These integrated systems can analyze and properly impact critical infrastructures, both native and adversary's, as well as create new infrastructures for a variety of purposes [27,28].

Air and missile defense: Providing flexible and self-recovering

distributed C2 infrastructures SGT can, for example, effectively use distributed networks of cheap ground or low-altitude sensors to discover, trace, and destroy multiple cruise missiles with complex routes by mobile spatial intelligence, versus existing expensive high-altitude planes, drones, and aerostats [19,29,30].

Crisis management: SGT can support advanced distributed systems for crisis management, where complex relief missions, national and international, can be organized and programmed in SGL quickly, “on the fly”, with the use of any available resources: human, robotic, any other mechanical and electronic ones [31,32].

Information on many other reported applications with the use of SGT or its previous versions can be found elsewhere [33-39].

Conclusions

We have briefly described and discussed the use of the developed high-level networking philosophy, methodology, and technology suitable for the support of unified transition to massively robotized up to fully unmanned systems, with numerous robotic components capable of effectively operating together.

The technology, based on holistic and gestalt ideas rather than traditional communicating agents, can organize any available human and technical resources into integral global-goal-driven teams capable of solving complex problems in rapidly changing environments. This may be particularly important taking into account the rapidly increasing world dynamics in the 21st century with frequent appearance of numerous national and international conflicts and absence of efficient ideologies, mechanisms, and technologies for their suitable solutions.

The technology, already prototyped in different countries and tested on numerous networked applications, can be readily implemented with the use of distributed dynamic networks of communicating interpreters embedded into individual equipment of soldiers, robots, smart sensors, etc. These networks are collectively executing integral mission scenarios represented in Spatial Grasp Language suitable for both human and robotic components, which can be easily swapped with each other at runtime under the unified automatic control, always fulfilling mission objectives.

References

1. Sapaty PS (1999) *Mobile Processing in Distributed and Open Environments*. John Wiley & Sons, New York, USA.
2. Sapaty PS (2005) *Ruling Distributed Dynamic Worlds*. John Wiley & Sons, New York, USA.
3. Sapaty PS (2014) The World as an Integral Distributed Brain under Spatial Grasp Paradigm. In *Intelligent Systems for Science and Information* 542: 65-85.
4. Sapaty PS (2011) Meeting the world challenges with advanced system organizations. In: Juan AC, Joaquim F, Jean-Louis F (eds.) *Informatics in Control Automation and Robotics*. Springer Berlin Heidelberg, Germany.
5. Sapaty PS (2012) Logic flow in active data. In: Delgado-Frias JG, Moore WR (eds.) *VLSI for Artificial Intelligence and Neural Networks*. Springer, New York, USA.
6. Sapaty PS (2008) Distributed technology for global dominance. *Proceedings of SPIE, Defence transformation and net-centric systems*, SPIE Optical Engineering Press, France.
7. <http://www.merriam-webster.com/dictionary/swarm>.
8. Wertheimer M (1924) *Gestalt theory*, Erlangen, Berlin.
9. Sapaty PS (2009) Gestalt-Based Ideology and Technology for Spatial Control of Distributed Dynamic Systems. *International Gestalt Theory Congress*, 16th Scientific Convention of the GTA, University of Osnabrück, Germany.
10. Sapaty PS (2009) Gestalt-based integrity of distributed networked systems. *SPIE Europe Security + Defence*, bcc Berliner Congress Centre, Berlin, Germany.
11. Minsky M (1988) *The Society of Mind*, Simon & Schuster, New York, USA.
12. Sapaty PS (2015) *Military Robotics: Latest Trends and Spatial Grasp Solutions*. *International Journal of Advanced Research in Artificial Intelligence* 4: 9-18.
13. Sapaty PS (2014) Unified Transition to Cooperative Unmanned Systems under Spatial Grasp Paradigm. *International journal Transactions on Networks and Communications* 2: 23-45.
14. Sapaty PS (2010) *High-Level Technology to Manage Distributed Robotized Systems*. *Military Robotics*, Jolly St Ermins, London UK.
15. Sapaty PS (2014) *From Manned to Smart Unmanned Systems: A Unified Transition*. *SMI's Military Robotics*, Holiday Inn Regents Park, London, UK.
16. Hsu J (2014) U.S. Navy Tests Robot Boat Swarm to Overwhelm Enemies. *IEEE Spectrum*, Accessed on: 5 October 2014.
17. Sapaty PS, Corbin MJ, Seidensticker S (1995) *Mobile Intelligence in Distributed Simulations*. 14th Workshop on Standards for the Interoperability of Distributed Simulations, IST UCF, Orlando, FL.
18. Sapaty PS, Corbin MJ, Borst PM (1995) Towards the development of large-scale distributed simulations. 12th Proc. Workshop on Standards for the Interoperability of Distributed Simulations, IST UCF, Orlando, FL.
19. Sapaty PS (2002) Over-Operability in Distributed Simulation and Control. *The MSIAC's M&S Journal Online* 4: 1-8.
20. Sapaty PS, Sugisaka M (2011) *Advanced Networking and Robotics for Societal Engagement and Support of Elders*. 16th International Symposium on Artificial Life and Robotics, Beppu, Oita, Japan
21. Sapaty PS, Sugisaka M (2011) Solving Demographic Problems with Distributed Information Technologies. *Mathematical Machines and Systems Journal* 1: 46-60.
22. Sapaty PS (2014) Distributed Human Terrain Operations for Solving National and International Problems. *International Relations and Diplomacy* 2: 597-622.
23. Sapaty PS (2015) Solving Social Problems by Distributed Human Terrain Operations. *Journal of Mathematical Machines and Systems* 3: 30-43.
24. Sapaty PS (2007) Intelligent management of distributed sensor networks. *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VI*, Orlando, Florida, USA.
25. Sapaty PS, Sugisaka M, Filipe J (2007) Making Sensor Networks Intelligent. 4th International Conference on Informatics in Control, Automation and Robotics, Angers, France.
26. Sapaty PS, Sugisaka M, Delgado-Frias J, Filipe J, Mirenkov N (2008) Intelligent management of distributed dynamic sensor networks. *Artificial Life and Robotics* 12: 81-87.
27. Sapaty PS (2015) Providing Over-operability of Advanced ISR Systems by a High-Level Networking Technology. *SMI's Airborne ISR*, Holiday Inn Kensington Forum, London, United Kingdom.
28. Sapaty PS (2014) Integration of ISR with Advanced Command and Control for Critical Mission Applications. *SMI's ISR conference*, Holiday Inn Regents Park, London, United Kingdom.
29. Sapaty PS (2012) Distributed air & missile defence with spatial grasp technology. *Intelligent Control and Automation* 3: 117-131.
30. Sapaty PS (2015) Distributed Missile Defence with Spatial Grasp Technology. *SMI's Military Space*, Holiday Inn Regents Park, London, United Kingdom.
31. Sapaty PS (2006) Crisis Management with Distributed Processing Technology. *International Transactions on Systems Science and Applications* 1: 81-92.
32. Sapaty PS, Sugisaka M, Finkelstein R, Delgado-Frias J, Mirenkov N (2006) Emergent Societies: An Advanced IT Support of Crisis Relief Missions. 11th International Symposium on Artificial Life and Robotics, Beppu, Japan.
33. Sapaty PS (1990) The Wave Model for advanced knowledge processing. In *CAD Accelerators*, Ambler AP, Agrawal P, Moore WR (Eds.) Elsevier, Netherlands.

34. Sapaty PS. A Distributed Processing System. European Patent No. 0389655, European Patent Office.
35. Sapaty PS (2015) Advanced Naval Operations under Spatial Grasp Technology. International Conference Naval Combat Systems, Park Plaza Victoria, London, United Kingdom.
36. Sapaty PS (2013) Night Vision under Advanced Spatial Intelligence: A key to Battlefield Dominance. SMI's Night Vision Conference, London, United Kingdom.
37. Sapaty PS (2009) Providing Spatial Integrity for Distributed Unmanned Systems. 6th International Conference in Control, Automation and Robotics ICINCO, Milan, Italy.
38. Sapaty PS (2008) Grasping the Whole by Spatial Intelligence: A Higher Level for Distributed Avionics. International conference Military Avionics, Café Royal, London, UK.
39. Sapaty PS (2008) Human-Robotic Teaming: A Compromised Solution. AUVSI's Unmanned Systems North America, San Diego, USA.