UDC 623.764

P.S. SAPATY[*]

ADVANCED AIR AND MISSILE DEFENSE UNDER SPATIAL GRASP TECHNOLOGY

[*]Institute of Mathematical Machines and Systems, National Academy of Sciences of Ukraine, Kyiv, Ukraine

Анотація. *Коротко викладено нову ідеологію та технологію керування для вирішення задач у великих мережевих системах. Запропонований підхід, який базується на активних сценаріях, що здійснюють самонавігацію і самосинхронізацію в розподілених просторах у режимі організованого супервіруса, може встановлювати глобальний контроль над системами довільної природи. Технологія дозволяє ефективно інтегрувати безліч розрізнених та неоднорідних об'єктів, дозволяючи їм працювати разом у цілеспрямованому суперкомп'ютерному режимі. Вона може бути корисною для перспективної протиповітряної та протиракетної оборони різними способами; деякі з них викладені та пояснені у цій статті. Відповідний матеріал прийнятий для презентації на 12-й Міжнародній конференції з Інтегрованої протиповітряної і протиракетної оборони, 27–29 червня в Стокгольмі, Швеція, http://3af-integratedairmissiledefence.com.*

Ключові слова: *протиповітряна та протиракетна оборона, розподілені мережеві системи, глобальний контроль, технологія просторового захоплення, сценарії, що саморозвиваються, цілісні рішення.*

Аннотация. *Кратко изложены новая идеология и технология управления для решения задач в больших сетевых системах. Предложенный подход, основанный на активных сценариях, осуществляющих самонавигацию и самосинхронизацию в распределенных пространствах в режиме организованного супервируса, может устанавливать глобальный контроль над системами любой природы. Технология позволяет эффективно интегрировать множество разрозненных и разнородных объектов, позволяя им работать совместно в целенаправленном суперкомпьютерном режиме. Она может быть полезна для перспективной противовоздушной и противоракетной обороны разными способами; некоторые из них изложены и объяснены в этой статье. Соответствующий материал принят для представления на 12-й Международной конференции по интегрированной противовоздушной и противоракетной обороне, 27–29 июня в Стокгольме, Швеция, http://3af-integratedairmissiledefence.com.*

Ключевые слова: *противовоздушная и противоракетная оборона, распределенные сетевые системы, глобальный контроль, технология пространственного захвата, саморазвивающиеся сценарии, целостные решения.*

Abstract. *A novel control ideology and technology for solving tasks in large distributed networked systems will be briefed. Based on active scenarios self-navigating and self-matching distributed spaces in a highly organized super-virus mode, it can effectively establish global control over large systems of any natures. The technology can use numerous scattered and dissimilar facilities in an integral and holistic way, allowing them to work together in goal-driven supercomputer mode. The approach can be useful for advanced air and missile defense in a variety of ways, some of which described and explained in this paper. The related material has been accepted for presentation at The 12th 3AF Integrated Air and Missile Defence international conference, June 27–29, Stockholm, Sweden, http://3af-integratedairmissiledefence.com.*

Keywords: *air and missile defence, distributed networked systems, global control, spatial grasp technology, self-evolving scenarios, holistic solutions.*

1. Introduction

High-level networking model and technology suitable for effective management of large distributed dynamic air and missile defense systems will be described which can cope with different kinds of unpredictable and asymmetric situations including massive air and missile attacks. Using this Spatial Grasp Technology, SGT, intelligent distributed command and control infrastructures, as demonstrated in detail, can be dynamically set up throughout distributed forces, which can keep global awareness and control, collect and disseminate information on multiple threats and targets, and organize their needed impact.

SGT can effectively withstand cruise missiles using highly organized distributed sensor networks, where individual sensors may be cheap, ground or low-flying, in contrast to the existing expensive high-altitude planes, drones and aerostats, or casual top mountain solutions. Multiple cruise missiles can be grasped by individual mobile intelligence following their physical move electronically via sensor network, not allowing them to escape despite tricky routes, due to holistic sensor network organization behaving as an integral spatial brain covering any area.

For the European type missile defense, it can provide flexible C2 allowing us to grasp multiple incoming missiles in parallel, and lead each missile individually through such stages as their infrared satellite pick up, relaying to sensors and weapons, supporting missile tracking by long-range sensors, and choosing upper or lower-layer available shooters. Due to freely moving intelligent scenarios, not connected in advance to particular physical resources, the whole system can work after indiscriminate failures or damages of any components, with their self-recovery or runtime substitution, always preserving the overall functionality. Another described scenario will be dealing with collective behavior of unmanned vehicles including situation where organized swarm of UAVs operating in SGT is fighting another manned, unmanned or mixed swarm fully autonomously and without external control.

The paper also provides a comparison between the description of distributed operations in traditional battle management languages on atomistic level of communicating military units with its holistic, semantic, equivalent provided by SGT. The latter being much simpler and shorter, also suitable for runtime composition and modification.

2. Spatial Grasp Technology, SGT

2.1. SGT General Issues

Starting from any point of space, SGT [1–3] allows us to create distributed operational infrastructure in a highly dynamic virus-like mode with absolute code mobility in computer networks. These infrastructures, covering any regions needed, can solve complex spatial problems in them without any central resources and in parallel. Such emergent infrastructures can effectively withstand different unpredictable, crisis, and asymmetric situations in distributed systems of both civil and defence orientation. The created infrastructures can self-recover and self-repair after indiscriminate damages while always securing mission objectives. After the task completion, the infrastructures can also self-clean and self-remove if not needed any more. The key element of SGT is its Spatial Grasp Language, SGL, in which all mission scenarios are formulated.

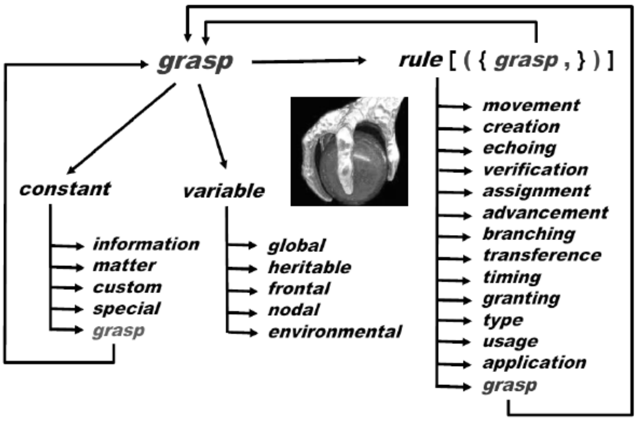

Fig. 1. Spatial Grasp Language

2.2. Spatial Grasp Language and its Distributed Interpretation

Pattern-based SGL can provide highly integral, holistic, gestalt-based solutions directly in physical, virtual, and executive worlds. SGL has universal recursive structure capable of representing any parallel and distributed algorithms in distributed environments (Fig. 1).

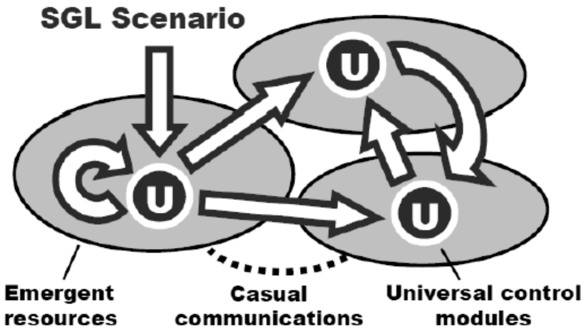

Fig. 2. SGL distributed interpretation

SGL is collectively interpreted by a network of universal control modules U, as SGL interpreters, embedded into key system points (humans, robots, sensors, internet hosts, etc.) with absolute scenario code and data mobility in space (Fig. 2). SGL scenarios can start from any node, covering at runtime the whole system or its parts needed with operations and control.

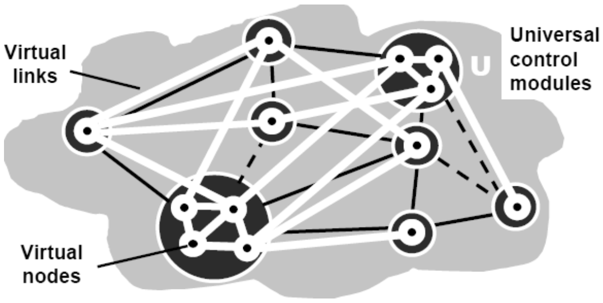

Fig. 3. Creating spatial infrastructures by SGT

2.3. Creating Knowledge Infrastructures

Spreading SGL scenarios can create knowledge infrastructures (Fig. 3) arbitrarily distributed between system components. Navigated by same or other scenarios, these can effectively support distributed databases, C2, situation awareness, and autonomous decisions. Also simulate any other existing or hypothetic computational and/or control models.

2.4. SGL Interpreter

SGL interpreter consists of a number of specialized modules handling & sharing specific data structures (Fig. 4). The whole network of the interpreters can be mobile and open, changing the number of nodes and communication structure between them at runtime.

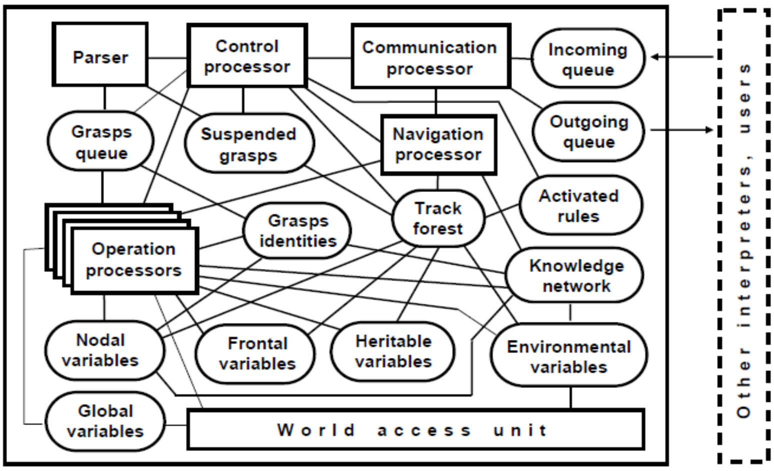

Fig. 4. SGL interpreter organization

A backbone of the distributed interpreter is its spatial track system providing overall integrity, global awareness and automatic C2 over distributed processes. Its main components are shown in Fig. 5.

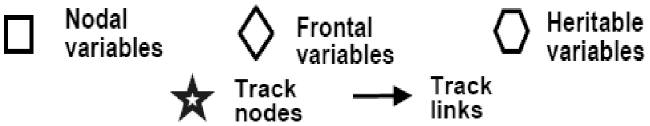

Fig. 5. Main components of the spatial track infrastructure

Figs 6 to 9 show different phases supporting spatial forward and echo processes induced by parallel self-navigating and self-evolving SGL scenarios. Detailed description and behaviour of this dynamic spatial infrastructure organization is explained, for example, in [3].

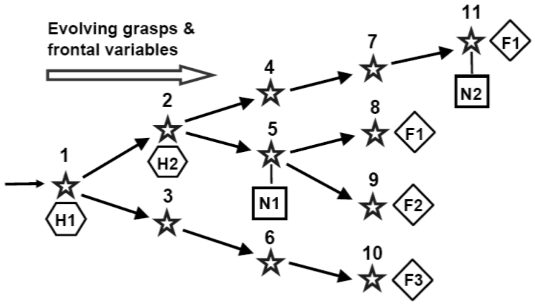

Fig. 6. Forward world grasping

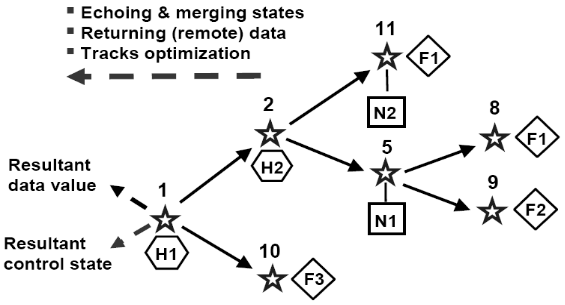

Fig**. 7.** Echoing & tracks optimization

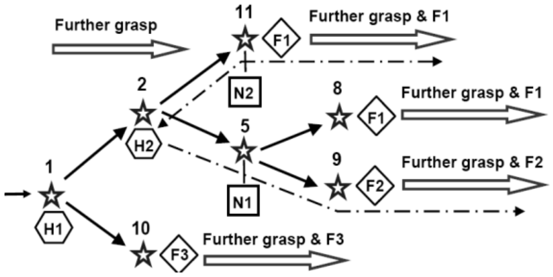

Fig. 8. Further world grasping

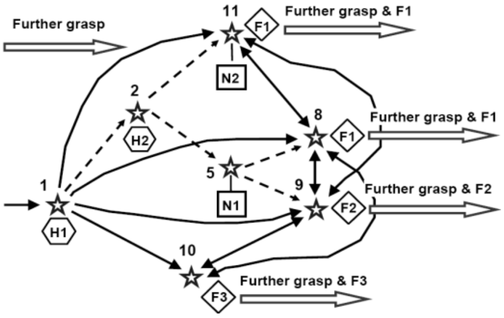

Fig. 9. More advanced track infrastructure

2.5. SGL Interpretation Network as a Universal Spatial Machine

The dynamically networked SGL interpreters extended by and integrated with other facilities and gadgets, like, for example, mobile robots, can form universal spatial machines operating with both information and physical matter (Fig. 10). These networked machines, working without any central resources under intelligent scenarios injected at any time and from any nodes, can perform complex computational, knowledge processing and control operations.

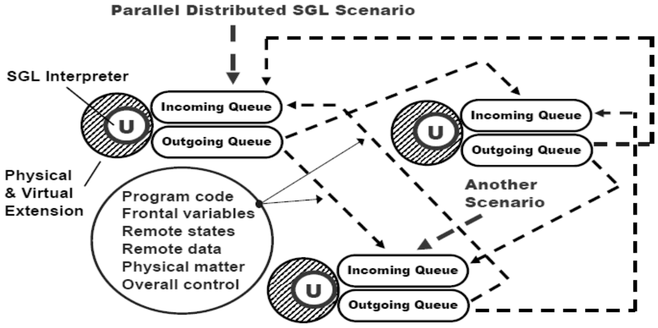

Fig. 10. SGL spatial machine

2.6. Embedding SGL Interpreters in Distributed Systems

By embedding SGL interpreters into robotic vehicles and electronic devices (including those associated with humans like smartphones, laptops or smart watches) we can easily organize any needed collective behavior of them, integrating them into holistic teams (with any similar or dissimilar components) operating under unified and distributed command and control. The collective mission scenario can start from any unit and cover, activate, and control at run time the whole group (as symbolically shown in Figs 11 and 12).

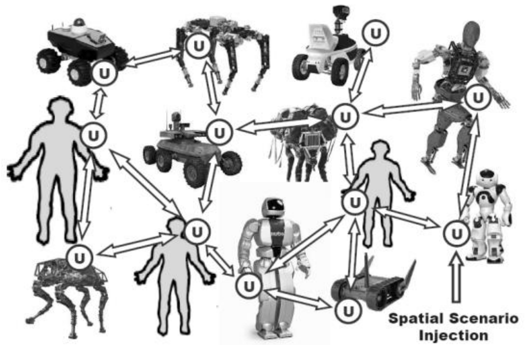

Fig. 11. Heterogeneous human-robotic collectives

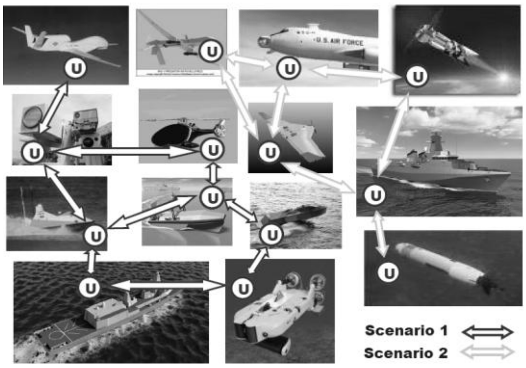

Fig. 12. Heterogeneous manned-unmanned defense solutions

3. Dynamic Creation of Distributed Command and Control Infrastructures

Imagine there is a fleet of sea vessels distributed over some area, and there also exist hostile objects that can be classified as targets to be discovered and eliminated (which may be aerial, surface, or submerged), as shown in Fig. 13.

An example of a hierarchical C2 infrastructure capable of fleet protection and covering all nodes is shown in Fig. 14, starting from a provisional command centre (as unit 1).

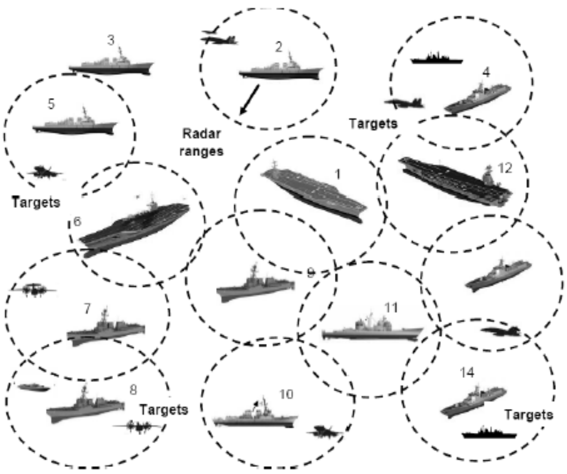

Fig. 13. Distributed fleet and hostile objects

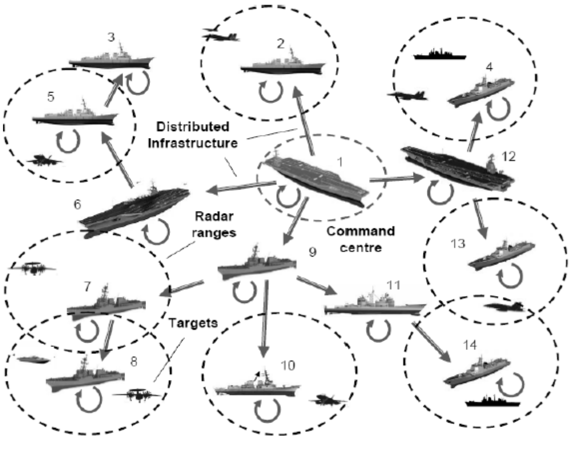

Fig. 14. Hierarchical operational infrastructure

This distributed infrastructure creation and operation scenario in SGL may be as follows, which forms infrastructure links `infra` based on closeness of units to each other, with the given threshold *`allowed_distance`* for units to be considered as C2 neighbours.

```
#1; frontal(Seen, Depth = allowed_distance);
stay(repeat_linkup(+infra,Depth,firstcome));
sling(
 Seen = repeat(free_detect(targets),+infra#);
 repeat(free_select_impact(Seen),+infra#))
```

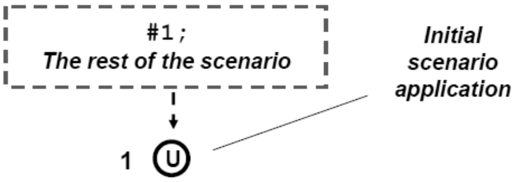

Fig. 15. Initial scenario injection

This self-evolving spatial scenario starts from the component selected as top of the hierarchy, as in Fig. 15.

It creates persistent hierarchical infrastructure covering all nodes in a stepwise top-down breadth-parallel manner by the following scenario fragment, and as in Fig. 16.

```
repeat_linkup(+infra, Depth, firstcome)
```

Triggered by the internal SGL interpretation mechanisms with modifier `firstcome`, the selected nodes can be visited only once by preventing this parallel spatial process from looping, always guaranteeing a tree-shaped resultant infrastructure with top-down oriented links `infra`.

The scenario then uses the created infrastructure in top-down parallel navigation mode to activate all units and detect locally seen targets throughout the whole region, collect the targets and merge them in a parallel bottom-up echo mode via the infrastructure, storing altogether in frontal variable `Seen` at the headquarter node `1`, as in Fig. 17 and the following scenario fragment.

```
Seen = repeat(free_detect(targets), +infra#)
```

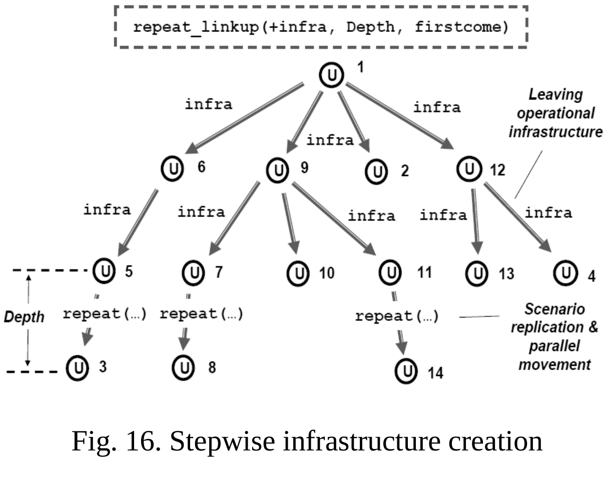
repeat_linkup(+infra, Depth, firstcome)
1
infra
infra
infra
Leaving operational infrastructure
6
9
2
12
infra
infra
infra
infra
infra
5
7
10
11
13
4
Depth
repeat(...)
repeat(...)
repeat(...)
Scenario replication & parallel movement
3
8
14

Fig. 16. Stepwise infrastructure creation

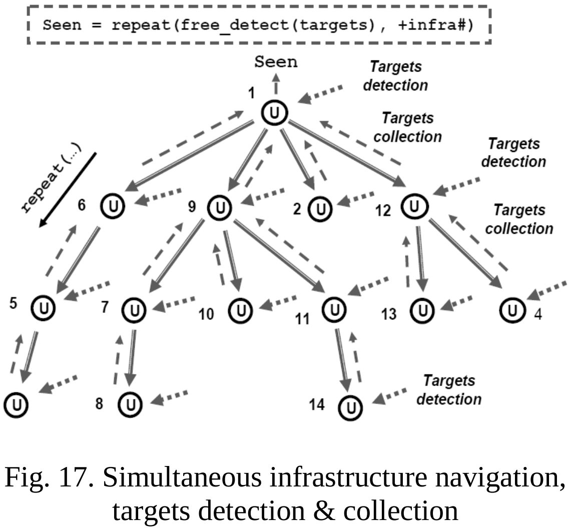

Fig. 17. Simultaneous infrastructure navigation, targets detection & collection

The targets collection by units is organized independently from the continuing global top-down units activation process via the infrastructure, with the use of additional rule `free`, but the discovered targets will be subsequently all merged by the bottom-up collection process.

The collected targets in frontal variable `Seen` at the headquarters node are then replicated and delivered to all units by parallel top-down spatial process using the created infrastructure where units, each having now full information about all targets, individually select the most convenient ones to shoot, as in the Fig. 18 and the scenario fragment that follows.

```
repeat(free_select_impact(Seen), +infra#))
```

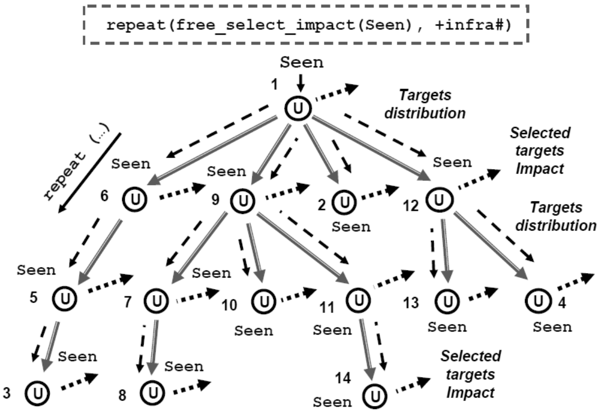

Fig. 18. Simultaneous targets distribution, selection & impact

The top-down targets delivery to all units is organized in parallel with possible targets impact operation by each unit, which is performed independently from the targets distribution process as soon as it receives the targets, which is properly managed with the use of additional rule free too.

The infrastructure-based parallel operations of spatial collection of distributed targets and their subsequent spatial distribution back to individual units with independent impact, shown in Figs. 17 and 18, can be organized together in a repeated mode using the rule sling, as in the full scenario text shown before. More on this organization can be found in [4].

4. Withstanding Cruise Missiles

SGT can effectively organize discovery, tracing, analysing and proper impact of multiple low flying objects like cruise missiles (Fig. 19), with complex and unpredictable routes [5] by cheap distributed sensor networks operating under mobile spatial intelligence provided by the technology.

Fig. 19. Tomahawk as a typical subsonic land-attack cruise missile

Cruise missiles have several advantages over ballistic missiles: they can be updated during flight, often pursuing complex routes to avoid detection. Their low flight altitude makes them very stealthy against air defence radars, and fuel efficient turbofan engines allow cruise missiles to be lighter and cheaper than their ballistic counterparts.

4.1. Existing Solutions

There are few and far from universal solutions for dealing with these types threats. Aerial sensors [6] are the best defence against low-flying cruise missiles, because they offer far better detection and tracking range than ground-based systems. The bad news is that keeping planes in the air all the time is very expensive, and so are the aircraft themselves. Another solution is called Mountain Top [7], where high elevation points on the ground can be used to trace and target low flying missiles, but this casual and totally terrain dependent.

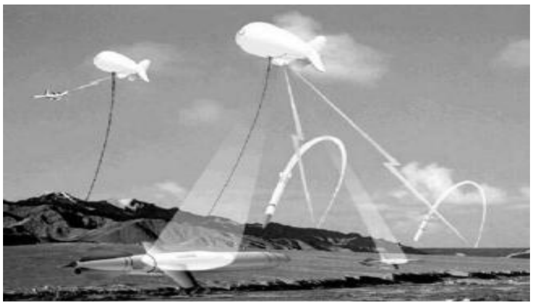

Fig. 20. Existing cruise missile defences: JLENS

The primary challenge becomes the development of a reliable, affordable, long-flying, look-down platform. One that can detect, track and identify incoming missiles, then support over-the-horizon engagements in a timely manner. The Joint Land Attack Cruise Missile Defence Elevated Netted Sensor (JLENS) [8] is an example of such systems, as in Fig. 20 (as far as we know, already terminated).

The unmanned, tethered platforms can complement each other through the operation of both broad-area and precision radar systems, providing an over-the-horizon early warning capability, but such an organisation is extremely expensive and cannot cover the whole theatre, say, as a country.

4.2. Installing SGL Interpreters with Distributed Sensors

Embedding SGL interpreters into networked radar stations can convert the latter into universal distributed self-organized supercomputers capable of solving any problems within the space covered. These may include discovery, tracing, analysing, and destroying multiple aerial objects and low flying cruise missiles. Communicating radars can be effectively integrated with SGL interpreters in large environments of different natures and their combinations, like open land terrain, sea/ocean surface vessels, or urban environment.

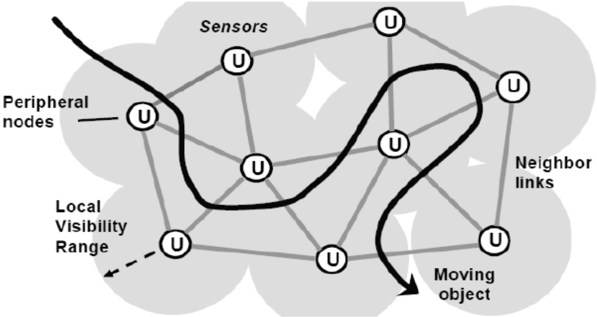

Fig. 21. SGL interpreters installed in communicating sensors with neighborhood links

Individual sensors have limited visibility range, but well organized distributed sensor networks empowered with SGT can provide continuous global vision of complexly moving objects through the space covered (as in Fig. 21), with their detailed study and destruction when required.

4.3. Distributed Missile Tracking Scenario in SGL

The SGL spatial tracking scenario may be as follows. Constantly operating in all region’s peripheral sensors it catches an incoming object it sees and then follows wherever it goes with the help of individual mobile intelligence if the object is not seen from the current point any more (i.e. its visibility becomes lower than some given `Threshold`).

```
frontal(Object, Threshold = min_visibility);
hop(periphery, all);
whirl(
 nonempty(
  Object = search(aerial, Threshold, new));
 release_repeat(
  loop(visibility(Object) >= Threshold);
  max_destination(
      hop(neighbors, all); visibility(Object))))
```

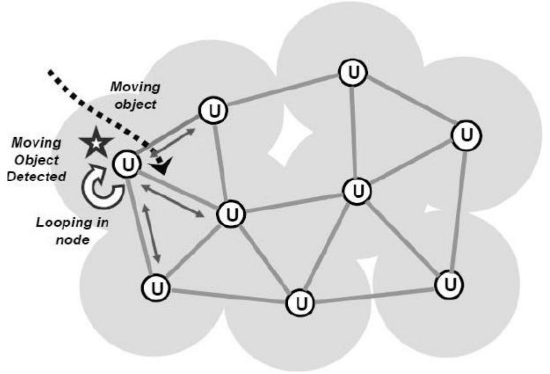

Fig. 22. Catching an alien object by a periphery sensor

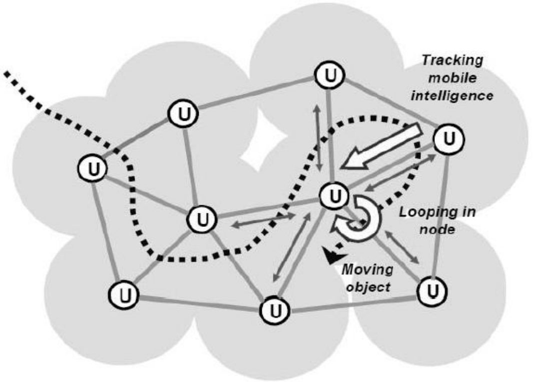

Fig. 23. Mobile intelligence keeping-following the moving object

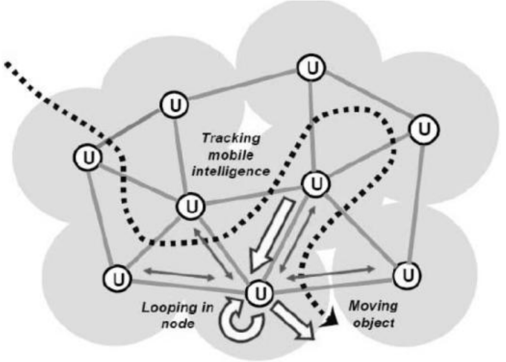

Fig. 24. Mobile object leaving the region if not destroyed

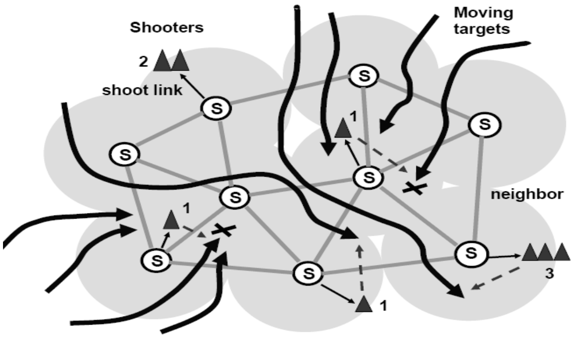

Fig. 25. Simultaneous tracing and destruction of multiple mobile targets in SGT

Some stages of this distributed object tracking dynamics are shown in Figs 22 to 24, where spatial intelligence accompanying the physically moving object via virtual networked space investigates the surrounding region if the object disappears from the current radar station, and then moves to the SGT-empowered neighboring radar where it is seen best.

4.4. Withstanding Multiple Attacks

Multiple, especially mutually coordinated attacks by low flying cruise missiles are considered at present as one of the most dangerous threats. These can be effectively handled (not only traced as in the previous section, but destroyed too) by mobile spatial intelligence of SGT with the use of distributed impact resources, as in the following scenario, also symbolically shown in Fig. 25.

```
nodal(Seen);
frontal(Object, Threshold = min_visibility);
hop(periphery, all);
whirl(
 Object = search(aerial, not_belong(Seen));
 visibility(Object) > Threshold;
 release(
  repeat(
   append(Seen, Object);
   loop(
     visibility(Object) > Threshold;
     if((hop(shoot_link); CONTENT > 0;
         shoot(Object); decrement(CONTENT)),
        (withdraw(Object, Seen); done)));
   withdraw(Object, Seen);
   max_destination(
     hop(neighbor, all); visibility(Object));
     if(visibility(Object) < Threshold),
        (output(Object & ‘ lost’); stop)))))
```

Each peripheral sensor is regularly searching for new targets, and each new target is assigned individual tracking intelligence which will propagate in distributed virtual space following the target’s movement in physical space. If there are available shooters in the vicinity at each stage, a kill vehicle is launched against the target (with their available number is reduced afterwards). If the target is hit, it is removed from the further observation.

The scenario above can be easily extended for the case when different mobile intelligent branches evolving in space can cooperate with each other and with some global optimization processes, also in SGL, say, to optimize the use of limited impact resources scattered throughout the region of control, or to identify and withstand multiple targets as the group ones with collective behavior.

5. Europe-Related Missile Defense Scenario

Let us consider here some scenarios relevant to the already discussed concept of a possible European missile defense system [9–11].

5.1. Missile Defense Main Stages

The missile defense system is supposed to work in the following stages.
Stage 1, shown in Fig. 26, with different steps numbered and having the following meaning.
1: Infrared satellite system picks up heat signatures of hostile missiles launched towards target.
2: Information transmitted to ground stations for processing.
3: Processed information sent to C2 network.
Stage 2, where the C2 network relays information to sensor and weapons systems available in the region, as shown in Fig. 27.

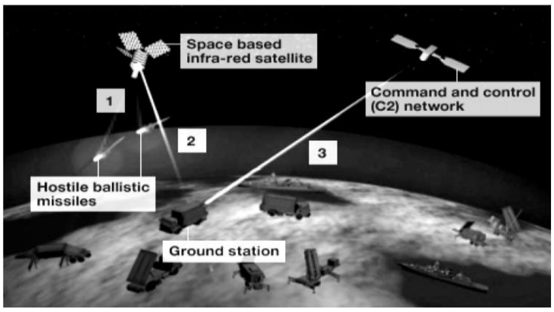
Space based infra-red satellite
1
2
3
Command and control (C2) network
Hostile ballistic missiles
Ground station

Fig. 26. Picking up a hostile missile

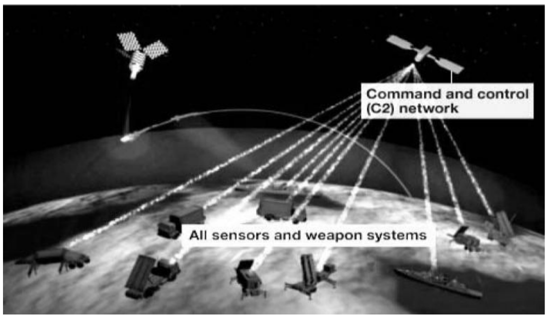
Command and control (C2) network
All sensors and weapon systems

Fig. 27. Relaying the information obtained

Stage 3 develops in the following steps, depicted in Fig. 28.
1: Long-range sensors continue to track the missile to help command system calculate options for destroying them.
2: Information is constantly shared among the sensors and weapons systems.
In Stage 4, the command system has the option of shooting down the hostile missiles while in the upper or lower layers of the atmosphere, using corresponding upper or lower-layer available shooters, as in Fig. 29.

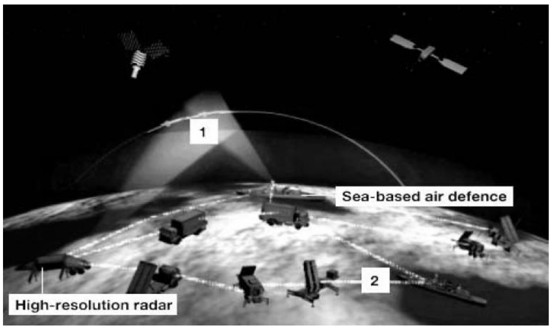

Fig. 28. Missile tracking & information sharing

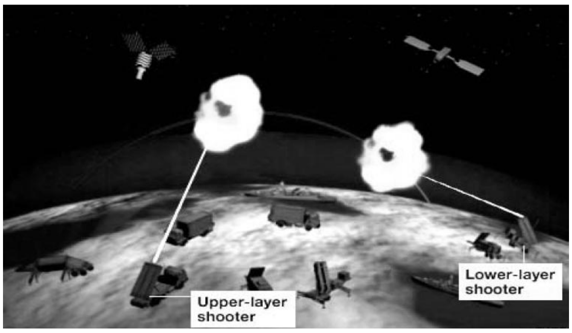

Fig. 29. Options of missile shooting

5.2. Missile Defense Management in SGL

We can symbolically extend the functionality and operations of the missile defense system mentioned above with possible Direct Energy Weapons, DEW (like high power lasers) located in space or on airborne manned or unmanned platforms [12].

Having synchronized DEW with infrared satellite sensors, we can write the whole SGL scenario integrating infrared satellites, DEW facilities, long range sensors, and upper and lower layer shooters into a dynamic flexible, distributed, and self-organized system.

This system would be capable of discovering many hostile objects in parallel, simultaneously and individually tracing them at different stages of their flight, also launching (or relaunching in case of failures) proper impact facilities with verification of their success or failure until the targets are destroyed, as follows.

```
hop(all, infrared_satellite_sensors);
frontal(Target, Threshold = upper_lower);
sling(
 split_discover(missiles,new);Target = VALUE;
 release(
  sling(
   visible(Target); update(Target);
   if((hop(DEW);verify_shoot(Target)),done));
   hop(long_range_sensors);
   sling(
    visible(Target); update(Target);
    if(distsance(Target) > Threshold,
       hop(upper_layer_shooters),
       hop(lower_layer_shooters));
    if(verify_shoot(Target), done));
    output('Alarm! Target: ',Target,' lost');
```

The advantages of this distributed management scenario are that it can be initially applied from any available system component, automatically creating distributed command and control infrastructure particularly oriented on the currently discovered targets and emerging situations. This automatically created distributed system organization can self-recover at runtime after indiscriminate damages to any system components (due to fully interpreted, mobile, virus-like implementation of SGL in distributed networked spaces).

5.3. Broader Integration

In a broader scale, SGT can effectively integrate numerous distributed, worldwide including, missile defence facilities and systems into global-goal-driven complexes operating under unified command and control, which can be fully automatic, especially in highly dynamic and asymmetric situations (Fig. 30).

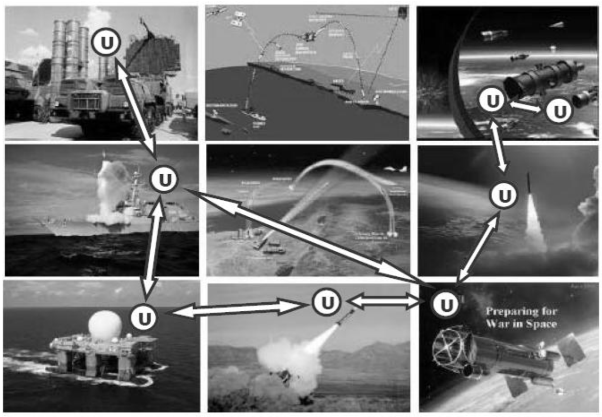

Fig. 30. Integrated global missile defense under SGT

6. Swarm against Swarm Aerial Scenario

We will consider here the case where an unmanned swarm is opposing another, supposedly unwanted, group of aerial vehicles. This, for example, can relate to fighting criminal and spying drones which are currently spreading worldwide [13], and may potentially operate in swarms too.

Main ideas of the following swarm against swarm SGL scenario, with alien drones as `Targets` and friendly drones as `Chasers` are shown in Fig. 31.

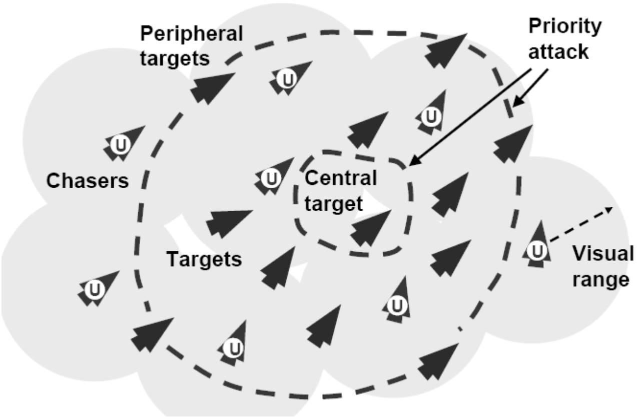

Fig. 31. Fighting group targets with an unmanned swarm

```
frontal(
 Chasers = …, Targets, Next, List, Center);
repeat(
 hop(random, Chasers);
 Targets =
  merge(hop(Chasers); coordinates(visible));
 nonempty(Targets);
 Center = average(Targets);
 List = sortdown(
  split(Targets);
  distance(VALUE, Center) & VALUE);
 List = append(withdraw(List, last), List);
 sling(
  nonempty(List); Next = withdraw(List,1):2;
  min_destination(
   hop(Chasers); STATUS == vacant;
   distance(WHERE, Next));
  STATUS = engaged;
 free(pursue_neutralize(Next);
      STATUS = vacant)))
```

Main steps and details of this scenario:

1. Initial launch of the swarmed chasers (as in Fig. 31, with SGL interpreters U embedded, which can communicate with each other) into the expected operational area.

2. Discovering targets and forming priority list by their physical positions where maximum priority is assigned to topologically central targets as potential control units of the intruders.

3. Other targets are sorted by their distance from the topological center of their group.

4. The most peripheral targets (those in maximum distance from the topological group's center), are assigned higher priority too as potentially having higher chances to escape.

5. Assigning available chasers to targets, classifying them as engaged, chasing and neutralizing targets, and returning into status vacant after performing the task.

6. The vacant chasers are again engaged in the targets selection and impact.

All chaser swarm management has been done exclusively within the swarm itself, without external intervention, which can dramatically simplify outside group tasking and control, and involve any number of unmanned units.

7. High-Level Battle Management in SGL

7.1. Formalization of command and control

Formalization of Command Intent (CI) and Command and Control (C2) in general are among the most urgent and challenging problems on the way to creation of effective multinational forces, integration of simulations with live control, and natural transition to robotized armies. Specialized Battle Management Languages for unambiguous expression of CI and C2 (like BML and its derivatives C-BML, JBML, geoBML, etc.) [14, 15] are not programming languages themselves, needing therefore integration with other linguistic facilities and organizational levels to provide required system parameters.

On the contrary, working directly with both physical and virtual worlds, SGL allows for effective and universal expression of any battlefield scenarios and orders in parallel and fully distributed manner within the same, universal, language syntax and semantics; it also directly supports robotized up to fully robotic systems. And scenarios in SGL are much shorter and simpler than in BML-based languages.

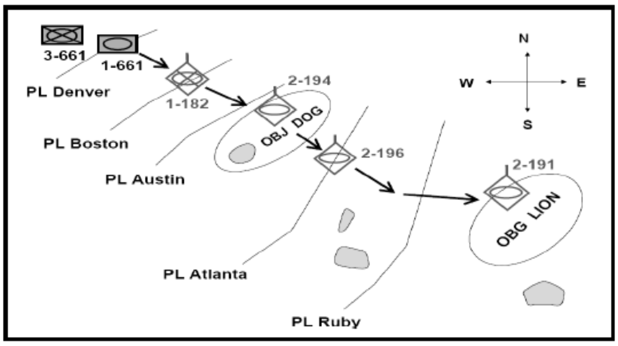

Fig. 32. An exemplary military scenario

7.2. Traditional battle management in BML

Let us consider an example taken from [14] and simplified in Fig. 32. The task is to be performed by two armoured squadrons BN-661 Coy1, and BN-661 Coy3, which are ordered to cooperate in coordination. The operation is divided into four time phases: from TP0 to TP1, from TP1 to TP2, from TP2 to TP3, and from TP3 to TP4, to finally secure objective Lion, and on the way to it, objective Dog. Their coordinated advancement should be achieved by passing Denver, Boston, Austin, Atlanta, and Ruby lines, while fixing and destroying enemy units Red-1-182, Red-2-194, Red-2-196, and Red-2-191.

The tasks from this scenario assigned to Coy1 are written in BML as follows:

```
deploy BN-661 Coy1 at Denver end before TP0
  in-order-to enable label-o11 label-o10;
advance BN-661 Coy1 from Denver to Boston
  start at TP0 in-order-to enable label-o12
  label-o11;
fix BN-661 Coy1 Red-1-182 at Boston end nlt
  TP1 in-order-to enable label-o33 label-o12;
advance BN-661 Coy1 to Austin start at TP1
  in-order-to enable label-o14 label-o13;
fix BN-661 Coy1 Red-2-194 at Dog end nlt TP2
  in-order-to enable label-o35 label-o14;
advance BN-661 Coy1 to Atlanta start at TP2
  in-order-to enable label-o16 label-o15;
fix BN-661 Coy1 Red-2-196 at Atlanta end nlt
  TP3 in-order-to enable label-o37 label-o16;
advance BN-661 Coy1 to Ruby start at TP3
  in-order-to enable label-o18 label-o17;
fix BN-661 Coy1 Red-2-191 at Lion end nlt TP4
  in-order-to enable label-o39 label-o18;
seize BN-661 Coy1 Lion at Lion end nlt TP4
  in-order-to cause label-ci1 label-o19;
```

The tasks assigned to Coy3 in BML are as follows:

```
deploy BN-661 Coy3 at Denver end before TP0
  in-order-to enable label-o32 label-o30;
support BN-661 Coy3 Coy1 at Troy start at TP0
  end at TP4 label-031;
attspt BN-661 Coy3 Red-1-182 from Denver to
  Boston start at TP0 end nlt TP1 in-order-to enable label-o12 label-
o32;
destroy BN-661 Coy3 Red-1-182 at Boston end
  nlt TP1 in-order-to enable label-o13 label-
  o33;
attspt BN-661 Coy3 Red-2-194 from Boston to
  Dog start at TP1 end nlt TP2 in-order-to
  enable label-o14 label-o34;
destroy BN-661 Coy3 Red-2-194 at Dog end nlt
  TP2 in-order-to enable label-o15 label-o35;
attspt BN-661 Coy3 Red-2-196 from Dog to
  Atlanta start at TP2 end nlt TP3 in-order-
  to enable label-o16 label-o36;
destroy BN-661 Coy3 Red-2-196 at Atlanta end
  nlt TP3 in-order-to enable label-o17 label-
  o37;
attspt BN-661 Coy3 Red-2-191 from Atlanta to
  Lion start at TP3 end nlt TP4 in-order-to
  enable label-o18 label-o38;
destroy BN-661 Coy3 Red-2-191 at Lion end nlt
  TP3 in-order-to enable label-o19 label-o39;
```

7.3. Same Management Scenario in SGL

This scenario can be presented in SGL as follows.

```
FIXER:BN_661_Coy1;
SUPPORTER_DESTROYER:BN_661_Coy3;
deploy(Denver, T:TP0);
advance_destroy(
 (PL:Boston, TARGET:Red_1_182, T:TP1),
 (PL:Austin,OBJ:DOG,TARGET:Red_2_194, T:TP2),
 (PL:Atlanta, TARGET:Red_2_196, T:TP3),
 (PL:Ruby,OBJ:LION,TARGET:Red_2_191, T:TP4));
seize(LION, T:TP4)
```

This semantic level description is much clearer, and more compact than if written in BML on the level of interacting individual units. This simplicity may allow us redefine the whole scenario or its parts at runtime, on the fly, when the goals and environment change rapidly, also naturally engage robotic units instead of manned components. We may further represent this battlefield scenario at other levels, for example, moving upwards with its generalization, as follows:

• Not mentioning own forces, which may become clear at runtime only:

```
deploy(Denver, T:TP0);
advance_destroy(
 (PL:Boston, TARGET:Red_1_182, T:TP1),
 (PL:Austin,OBJ:DOG,TARGET:Red_2_194, T:TP2),
 (PL:Atlanta, TARGET:Red_2_196, T:TP3),
 (PL:Ruby,OBJ:LION,TARGET:Red_2_191, T:TP4));
seize(LION, T:TP4)
```

• Further up, not mentioning adversary’s forces, which may not be known in advance but should be destroyed if discovered, to move ahead:

```
deploy(Denver, T:TP0);
advance(
 (PL:Boston, T:TP1),
 (PL:Austin, OBJ:DOG, T:TP2),
 (PL:Atlanta, T:TP3),
 (PL:Ruby, OBJ:LION, T:TP4));
seize(LION, T:TP4)
```

• Further up, setting main stages only, with starting and final time only known:

```
deploy(Denver, T:TP0);
advance(PL:(Boston, Austin, Atlanta, Ruby));
seize(LION, T:TP4)
```

• And final goal only (or just Command Intent, CI):

```
seize(LION, T:TP4)
```

Having the same formal language for all system levels and their any mixtures provides us with high flexibility for organization of advanced missions, especially with limited or undefined in advance resources and unknown environments, also possibility of potentially unlimited engagement of robotic components under unified command and control.

8. Conclusions

The approach presented can organize multiple distributed air and missile defence facilities into integral global goal-driven systems capable to withstand numerous unpredictable and dangerous situations in our rapidly changing world. The resultant technology had a number of trial implementations in different countries with its latest version being patented again. It can be implemented on an agreement on any platform needed and within limited period of time. More on SGT and its numerous applications can be found in the existing publications [1–4, 11, 12, 16–21].

REFERENCES

1. Sapaty P.S. Mobile Processing in Distributed and Open Environments / Sapaty P.S. – New York: John Wiley & Sons, 1999. – 436 p.

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

3. Sapaty P. Managing Distributed Dynamic Systems with Spatial Grasp Technology / Sapaty P. – Springer, 2017. – 284 p.

4. Sapaty P.S. Advanced Maritime Operations Under Network-Centric Organizations [Електронний ресурс] / P.S. Sapaty // Proc. of the International Conference Warship 2016: Advanced Technologies In Naval Design, Construction & Operation. – Guildhall, Bath, UK, 2016. – 15-16 June. – 12 p. – Режим доступу: https://www.rina.org.uk/res/WAR16%20Brocure%20WEB.pdf.

5. Xian Y. Cruise Missile Route Planning Based on Quantum Immune Clone Algorithm [Електронний ресурс] / Y. Xian // J. of Information & Computational Science. – 2012. – Vol. 9, N 8. – P. 2097 – 2105. – Режим доступу: www.joics.com.

6. Osborn K. F-35 Intercepts Cruise Missile To Defend Ship During Important Test [Електронний ресурс] / K. Osborn // The BUZZ. – 2016. – Sept. 14. – 2 p. – Режим доступу: http://nationalinterest.org/blog/the-buzz/f-35-intercepts-cruise-missile-defend-ship-during-important-17707.

7. Zinger W.H. Mountain Top: Beyond-the-Horizon Cruise Missile Defense [Електронний ресурс] / W.H. Zinger, J.A. Krill. – Режим доступу: http://www.jhuapl.edu/techdigest/TD/td1804/zinger.pdf.

8. LENS: Co-ordinating Cruise Missile Defense [Електронний ресурс] / And More. – Режим доступу: http://www.defenseindustrydaily.com/jlens-coordinating-cruise-missile-defense-and-more-02921/.

9. Ballistic missile defence, NATO [Електронний ресурс]. – 2016. – Режим доступу: http://www.nato.int/cps/en/natolive/topics_49635.htm.

10. NATO missile defence system [Електронний ресурс]. – Wikipedia. – Режим доступу: https://en.wikipedia.org/wiki/NATO_missile_defence_system.

11. Sapaty P.S. Distributed Air & Missile Defense with Spatial Grasp Technology / P.S. Sapaty // International Journal Intelligent Control and Automation (ICA), Scientific Research. – 2012. – Vol. 3, N 2. – P. 117 – 131.

12. Sapaty P. High-level Organisation and Management of Directed Energy Systems / P. Sapaty // Proc. Directed Energy Weapons 2010, March 25-26, CCT, Canary Wharf. – London UK. – 55 p.

13. Sanchez A. Worst Case Scenario: The Criminal Use of Drones [Електронний ресурс] / A. Sanchez, C. McKibben // Council on Hemispheric Affairs. – October 28 (Friday), 2016. – Режим доступу: http://www.coha.org/worst-case-scenario-the-criminal-use-of-drones/.

14. Command and Control Lexical Grammar (C2LG) Specification [Електронний ресурс] / U. Schade, M.R. Hieb, M. Frey [et al.] // FKIE Technical Report ITF/2010/02. – July 2010. – Режим доступу: http://c4i.gmu.edu/eventsInfo/conferences/2011/BMLsymposium2011/papers/BML-Symposium-Schade.pdf.

15. Coalition Battle Management Language (C-BML), RTO Technical Report TR-MSG-048 [Електронний ресурс]. – February 2012. – Режим доступу: http://www.dtic.mil/dtic/tr/fulltext/u2/a559456.pd.

16. Sapaty P. Air & Missile Defense with Spatial Grasp Technology / P. Sapaty // Unmanned Systems Asia, Workshop B. – Singapore, 2011. – 65 p.

17. Sapaty P. High-Level Management of Distributed Air and Missile Defence Systems / P. Sapaty // Air and Missile Defence. – India, New Delhi, 2012. – 67 p.
18. Sapaty P. Providing Over-operability of Advanced ISR Systems by a High-Level Networking Technology / P. Sapaty // Airborne ISR. – London, 2015. – 58 p.
19. Sapaty P. Military Robotics: Latest Trends and Spatial Grasp Solutions / P. Sapaty // IJARAI. – 2015. – Vol. 4, N 4. – P. 9 – 18.
20. Sapaty P. Distributed Missile Defence with Spatial Grasp Technology / P. Sapaty // Military Space. – London, 2015. – 69 p.

Стаття надійшла до редакції 19.06.2017