# Title: **Future C2 Architecture for Distributed Execution: A Case Study of Intelligent Particles**

## Topic 9: **C2 Architectures and Technologies**

Authors:

**Georgiy M. Levchuk**
Aptima Inc.,
12 Gill Street, Suite 1400
Woburn, MA 01801
Phone: 781-935-3966x267
Fax: 781-935-4385
e-mail: georgiy@aptima.com

**Nathan Schurr**
Aptima Inc.,
12 Gill Street, Suite 1400
Woburn, MA 01801
Phone: 781-935-3966x253
Fax: 781-935-4385
e-mail: nschurr@aptima.com

**Darby Grande**
Aptima Inc.,
12 Gill Street, Suite 1400
Woburn, MA 01801
Phone: 781-935-3966x225
Fax: 781-935-4385
e-mail: dgrande@aptima.com

**Webb Stacy**
Aptima Inc.,
12 Gill Street, Suite 1400
Woburn, MA 01801
Phone: 781-935-3966x237
Fax: 781-935-4385
e-mail: wstacy@aptima.com

Correspondence:

**Georgiy M. Levchuk**
Aptima Inc.,
12 Gill Street, Suite 1400
Woburn, MA 01801
Phone: 781-935-3966x267
Fax: 781-935-4385
e-mail: georgiy@aptima.com

# Future C2 Architecture for Distributed Execution: A Case Study of Intelligent Particles

Georgiy Levchuk[1a], Nathan Schurr[a], Darby Grande[a], and Webb Stacy[a]
[a]Aptima Inc.

## Abstract

As the scale of automated and unmanned forces increases, a crucial challenge of controlling, coordinating, and synchronizing the operations of individual force components must be solved. In asymmetric future environments, including urban security operations or disaster relief missions, the traditional hierarchical command and control (C2) organizational structures would not provide enough flexibility needed to adapt the tactical operations. To enable agile functioning of the force, instead of such centrally-controlled operations, researchers recently explored the fully distributed control strategies. However, fully distributed operations are only possible for simple team missions, and are not effective in executing the plans that have complex resource, temporal, and relational dependencies. Our previous research focused on designing hybrid organizational command, control, communication, and information structures, which combine benefits of hierarchical and heterarchical organizational principles to enable agile operations.

In this paper, we describe the application of hybrid C2 design to the domain of automated shape assembly, which we performed in the project called Cognitive Particles sponsored by DARPA, Defense Sciences Office. The objective of this research was to develop the processes and control architecture for the automated particle-based shape formation for a large variety of shape structures and under significant uncertainties in the construction process. The roles of commanders are fulfilled by “active particles”, and the roles of force units are fulfilled by “reactive particles”. The particles coordinate to execute the overall plan based on specified shape objectives; these processes are similar to command, control, and coordination requirements of automated and semi-automated forces. Current particle formation frameworks are based on the rigid skeleton assemblies, which can only handle a fixed set of predefined shapes, and cannot adapt the construction to the action uncertainty or changes in the environment. As such, the particle-based shape assembly provided a perfect case study to test the concepts of the agile organizational structures and processes.

## Motivation

### *Planning and Controlling Semi-Automated Forces*

Traditional planning, organizational design, and tactical execution processes of U.S. military depended on having relatively complete knowledge of the threat (e.g., composition of enemy forces, doctrine, likely operational and tactical situations and geographic conditions). However, both conventional adversaries and asymmetric threats confronted in the Current Operational Environment (COE) can no longer be fully engaged using conventional approaches and organizations. They require more facile, dynamic organizational structures that enable agile and precise operational and tactical actions. As a response to volatile environments, organizations struggle to balance stability against flexibility, specialization against generalization, and centralization against decentralization (Alstyne, 1997).

A traditional command and control (C2) *hierarchy* has a topology that largely restricts interactions among members of the organization to direct superior/subordinate interactions and whose number of levels is determined by the limits of span of command (Alberts and Hayes, 2003). Its approach to command and control is characterized by centralized planning, decomposition of tasks, and control processes that largely rely on deconfliction. A *heterarchy* is an emergent, self-organizing form that resembles a network or a

[1] georgiy@aptima.com; phone 781-935-3966x267; fax 781-935-4385; www.aptima.com

fishnet (Alberts and Hayes, 2003; Levchuk et al., 2003, 2004; Yu, Tu, and Pattipati, 2008). It has lateral or distributed authority, has no fixed superior and has bi-directional relationships among team members.

An organization, which utilizes the beneficial characteristics of both hierarchy and heterarchy and can evolve over time, is termed a *hybrid* organization (Levchuk et al., 2003, 2004). Hybrid networked organizations uncouple command from control: command is involved in setting the initial conditions and providing the overall intent, while control is not a function of command, but an emergent property that is a function of the initial conditions, the environment, and the adversaries. Such organizations are designed to be agile while at the same time able to execute complex plans that require joint synchronized operations. Since agility requires not only the ability to respond to the changing environment, but also maintain a high level of performance in succeeding objectives (Alberts and Hayes, 2003), hybrid organizational structures bring both the ability to adapt the organization and stay organized and coordinated over the time of the mission.

U.S. Military is undertaking a gradual redesign of its organizational units to provide modular forces focused on joint and expeditionary capabilities. For example, the current U.S. Army’s Brigade Combat Team (BCT), the outcome of the Unit of Action (UA) concept, typifies this change (FMI 3-90.6). These new tactical formations are characterized by modularity of force composition (including joint, allied or coalition units), which allows resources to be rapidly configured or ‘packaged’ for specific mission requirements. Modular forces enhance the ability to quickly respond to wide range of contingencies with proper force composition (neither too large nor too small).

The constraints faced by human C2 organizational redesign seem less obvious in the control of semi-automated forces, where the single or multiple human controllers operate the autonomous units. These units may include unmanned aerial vehicles (UAVs), underwater vehicles (UVs), or other types of air and ground robotic forces and platforms. However, many similarities with human organizations recently started to emerge as the more unit autonomy and intelligence is introduced into the robotic systems, ranging from path planning, locomotion, target identification, and action selection. As shown in Figure 1, as the unit intelligence and autonomy is increasing from mesoparticles to humans, many organizational similarities remain and can be exploited in specifying plans, designing command and control architectures, and developing tactical execution strategies.

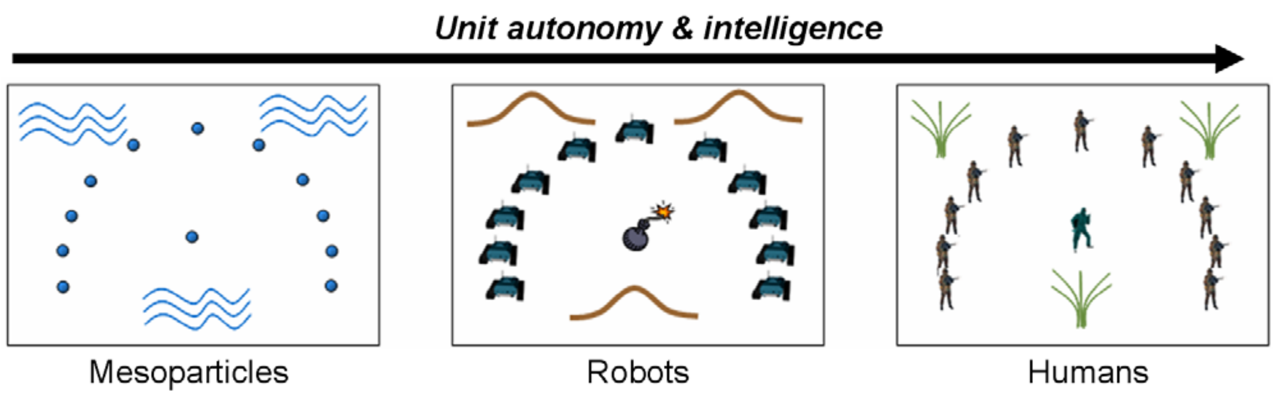



**Figure 1: The spectrum of controllable teams: from mesoparticles to humans**

In our previous research, we validated in virtual empirical studies that alternative C2 organizational designs can significantly outperform traditional structures (Kleinman et al., 2003; Entin, 1999; Entin et al., 2003; Levchuk et al., 2003, 2006). Still, experimenting in real-world settings with C2 designs for human organizations presents many doctrinal challenges. We instead focus our current research on enhancing this work in semi-automated and fully-automated command and control domains. One such domain is the automated assembly of the shapes with specified properties given a collection of active and reactive particles. This paper describes the distributed command and control architecture and concomitant testbed we have developed to conduct automated shape assembly under various uncertainties in the control effects and the observations.

## *Statement of the Problem Analyzed in our Case Study*

In this paper, we describe the application of our hybrid C2 design framework to the domain of automated shape assembly. This work was performed under the project called Cognitive Particles sponsored by DARPA, Defense Sciences Office. The objective of this project was to develop intelligent autonomous components (particles) that are capable of coordinating to move and form an arbitrary object in the uncertain environment. This research was part of a larger emerging field often referred to as Programmable Matter because of its goal of creating a particle substance that can be programmed to change its physical properties (e.g., shape, density, color, etc.). Most current research in Programmable Matter focuses on the lower level issues of how to enable its component particles to successfully complete actions such as move, communicate, and join with adjacent particles. Instead, the Cognitive Particles project focused on the control and coordination challenges that arise once these lower-level actions are made possible.

In our research, we focused on three key questions:

- *What are the coordination challenges of automated shape assembly?*
- *What is necessary for the automated object assembly planning and execution?*
- *What are important metrics of object formation, and how does the object plan and execution affect them?*

Cognitive Particles has begun to answer each of these questions, which has led to accomplishments in the areas of (1) Design and development of a Cognitive Particle Testbed, (2) Cognitive Particles theory development, and (3) Cognitive Particles metrics and results. These are all readily extensible to the Programmable Matter field and to the command and control of semi-automated forces. Our accomplishments, lessons learned, and future directions are detailed below.

## *Cognitive Particles and Agile Command and Control*

Current shape assembly frameworks are based on the rigid skeleton-based formation, which while allowing the fully distributed shape forming can only handle a fixed set of predefined shapes, and cannot adapt the construction to the assembly success uncertainty or changes in the environment (Jacobs et al., 2002; Jacobs and Wei, 2006; Zheng and Jacobs, 2006). These traditional assemblies resemble industrial age command and control in both structures of the organization (the skeleton becomes a rigid hierarchy) for shape formation and the processes employed to conduct the assembly (the local assembly units follow the limited set of rules to satisfy the given objectives and have little or no initiative). As such, the particle-based shape assembly provided a perfect case study to test the concepts of the agile organizational structures and processes. We needed to satisfy the objectives of assembling a diverse set of objects based on only general properties and succeeding in assembling the shapes under various adverse conditions including gravitational and friction forces, failures of primitive locomotion and connection actions, failures of the individual active particles to communicate commands, uncertainties in the observations of the current state of the (sub)shapes, etc. As the result, we needed to develop agile command and control structures and processes for the particle assembly domain. Our framework for controlling shape assembly captures the following six attributes of agile C2 systems (Alberts and Hayes, 2003):

- **Robustness:** Our framework allows designing, planning, and assembling any shape that can be defined by its physical, structural, and visual properties. This is in contrast with traditional assembly techniques that rely on the skeleton or scaffolding design and thus are only appropriate for a predefined fixed set of shapes;
- **Resilience:** The cognitive particles framework allows to recover the shape formation from the failures of the construction process (incorrect physical orientation and substructures preventing the further assembly can be adjusted by local disassembly and corrective force applications) and failures of the controlling elements (failing active particles can be replaced by other active particles and roles can be reallocated to continue assembly);

- **Responsiveness:** The active particles are self-synchronizing their individual operations and dynamically adjusting the next commands and communications based on the observations of the forming connections and 3D positioning of the building blocks of the shape;
- **Flexibility:** The shape planning process allows to develop diverse contingencies to execute the same shape assembly in different steps, as well as form different shape configurations to fulfill the same set of higher-level shape objectives;
- **Innovation:** The changes in assembly instructions can be performed in situations when multiple (potentially competing) assemblies occur, when little knowledge exists about the state of the shape, and when little is known about the particle locomotion and connection capabilities.
- **Adaptation:** Our framework allows the organization of active particles to adapt the organizational networks, the structure of the shape, the temporal plan of the structure assembly, the roles of active particles, and reactive particles selected to fulfill assembly blocks.

In the next sections, we provide details of our shape assembly framework, and discuss its benefits to the effectiveness of the assembly process and more generally to the design of agile command and control organizations.

## Definition of Concepts for Automated Shape Assembly

In this section, we define the main concepts of objects, shapes, their content, the physical environment, planning, command, and control processes to assemble the shapes.

### *Defining Objective of the Assembly*

The objective is to construct a certain **physical object of interest** from the set of elementary components. The *object description*, which is developed by the user, specifies the properties of the object in the form of its shape and kinetics. The *shape definition* might be of descriptive nature, but must be translated into a topological specification of how elementary components may form this shape. Such specification, described in detail in the next section, can be either defined by the user, or can be automatically derived using *quantization* of the shape form, given that the assembly system can match the description of the form with known shape. For example, the user may desire to build a “ball”; the system must understand that the ball is of spherical form with a surface equally distanced from the center of the shape. Starting the shape assembly at the object description and shape definition, our framework allows constructing a wide range of objects using alternative plans and execution arrangements, making it *robust* to the requirements of the user.

The *kinetics definition* for the object includes the desired properties that the user may wish to obtain. For example, the user may specify a desire for the ball to stay intact under a heavy pressure and to have a high bounce capability. Such definitions can be translated into topological properties of the component elements of the shape and their connections. An example might be a “baseball” that has multiple layers with different properties (cork, rubber, and mixture of the two, with liner components) designed to achieve desired resistance and weight properties.

### *Elementary Components of the Assembly*

The elementary components that can be used to build a shape must be defined. As a construction company may use bricks or wood components to build a house, or a military brigade-size force may use platoons as the non-decomposable units to plan and execute its missions, the shape assembly controller may specify the type of elementary components available to conduct the assembly. In Cognitive Particles project, we have used the *particles* with shape of a small 3-dimentional cube as the elementary components (Figure 2a). The size of the particles was fixed to be the same, while the color could be varied for the visualization purposes and to design the shapes with specific color features. The particles were assumed to be able to connect to each other along their faces maintaining alignment along the corresponding faces (Figure 2b).

Without loss of generality, we considered the elementary particles with *homogeneous physical properties* – that is, we assumed that all elementary particles are made of (or require) the same material to be manufactured, and thus are equivalent in the physical world. We made a similar assumption about the connections among particles, assuming that a single type of connection exists. This can be expended to particles with heterogeneous properties, as is needed for construction of a baseball as described above, as well as heterogeneous links, for example rigid joints, springy links, or rag-doll-like connections.

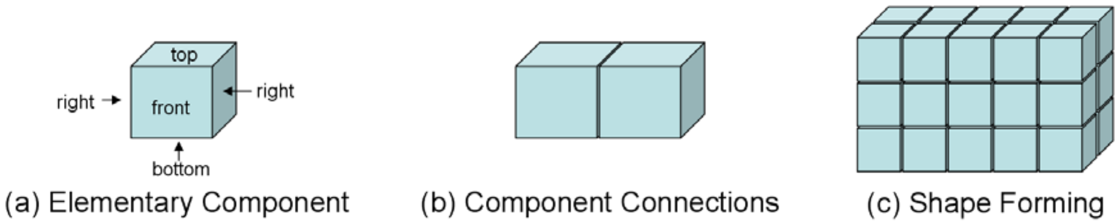
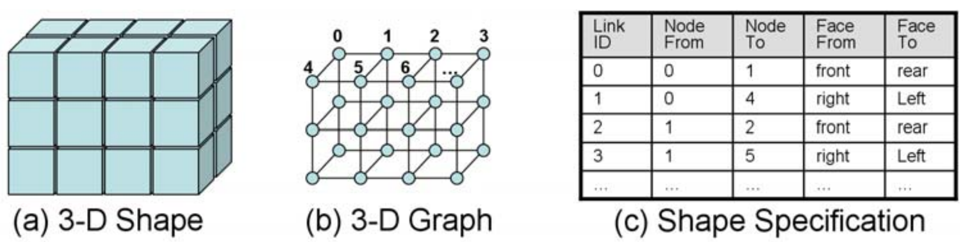



**Figure 2: Elementary components: particles and their connections**

### *Shape Structure Specification*

In the domain of multi-unit force control, the main objective is specified as the final state of the environment in which the forces operate. Similarly, in the shape assembly domain, the objective is specified as the state of the particle matter. This state is a structure of the desired shape (Figure 3a), and can be topologically defined as a 3-D graph. In this graph, nodes are particles that must constitute the shape, and links are joints between these particles. The nodes in this graph can be indexed with integer values (Figure 3b); then, links carry information about the connecting node indexes and faces of the corresponding cube particles. Figure 3 shows an example of the shape representation. We call this topological construct a *node-link specification* of the desired shape. As we are dealing with well-aligned cube particles and homogeneous links, the geo-spatial information (e.g., location and orientation) of the particle in the shape is not needed. Moreover, this information is redundant, because we assume the user does not care what orientation the shape will be manufactured at.




| Link ID | Node From | Node To | Face From | Face To |
|---|---|---|---|---|
| 0 | 0 | 1 | front | rear |
| 1 | 0 | 4 | right | Left |
| 2 | 1 | 2 | front | rear |
| 3 | 1 | 5 | right | Left |
| ... | ... | ... | ... | ... |

**Figure 3: Node-link specification of desired shape**

Quantitatively, the shape topology can be specified as an *attributed network*, where the nodes represent particles, links represent the joints among them, and the attributes on nodes and links describe the profiles (or properties) of the particles and their connections. In our work, we only assumed that the connections have attribute information – defined using faces of the particles the connection is supposed to join. Such network can be defined using a triplet $G=(V,E,A)$, where $V=\{1,...,|V|\}$ are nodes corresponding to components of the shape, $E$ is a set of links between them, and $A$ is a set of attributes on links and nodes determining the properties of the particles and their connections ($A=\|a_{ij}\|$, where $a_{ii}$ is attributes vector for node $i$ and $a_{ij}$ is attribute vector for link between nodes $i$ and $j$). This description can define both directed and undirected shape specifications (in case of undirected specification the attributed matrix $A$ is symmetric).

We define the physical (current) particle network using the variables $G_C=(V_C,E_C,A_C)$ and a desired shape using $G_D=(V_D,E_D,A_D)$. Current physical structure changes over time, as the shape is being built or disassembled. The desired shape structure remains constant according to the node-link specification of the shape topology. Note that the user might specify *multiple objects* as the objectives for the manufacturing, where only a single object must be manufactured. Such specification may be needed when the user can be satisfied with obtaining any of the several objects with various degrees, and the manufacturing process has cost-benefit tradeoff. For example, the user might desire to build pliers or scissors, and while pliers would match the most to the needs, the construction of the scissors may be simpler and this object would satisfy the requirements to a certain degree. The system then must intelligently weigh in different values to come up with the specific node-link spec to be executed during assembly.

# Organizational Structure, Execution Planning, and Execution Processes in Automated Shape Assembly

## *The Organizational Structure*

As the commanders of military organizations control their subordinate forces, plan operations, and coordinate among each other, these intelligent functions must be present in efficient shape assemblies. In our work, we rely on the assembly being conducted with the help of particles that have embedded intelligence. Availability of such particles is limited, but they bring the value of distributed shape assembly that is not available in a centralized construction processes. As the result, we distinguish two types of particles:

- **Reactive particles:** these are standard particles (sometimes referred to as *resources*) with limited memory and no intelligence, and can be “told” to create connections with other particles.
- **Active particles:** these are particles that have memory and intelligence. Sometimes referred to as *commanders*, active particles can decide about the instructions that must be executed to construct a shape, communicate information, request information, send commands to other active particles, and send instructions to reactive particles. Active particles are allowed significant freedom in selecting their roles, choosing the reactive particles for assembly, and developing and coordinating execution of subplans. Thus, active particles are the source of *innovation* and *responsiveness* needed to support the agility of the shape assembly.

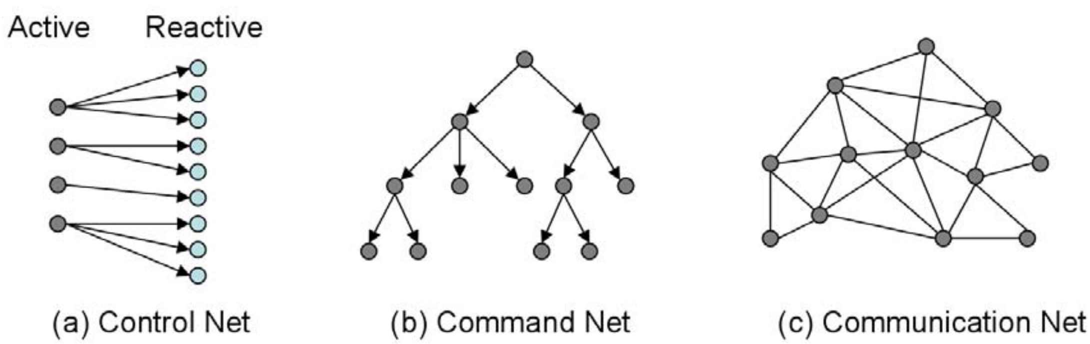



**Figure 4: Networks in the assembly organization**

Both active and reactive particles possess ability to move in the environment and create joints (connections) with other particles. Active and reactive particles form the organization, referred to as *command and control* assembly organization (Figure 4). It has the following three networks, following similar constructs for the military organizational designs (Levchuk et al., 2002, 2005, 2006):

- **Control network:** we define an assignment of reactive particles to active particles. An active particle can send instructions only to those reactive particles that it is assigned in a control network. A control network can be defined using a variable $c_{ij} = 1$ if the active particle $i$ is assigned reactive particle $j$ (otherwise $c_{ij}=0$). Only one active particle can *control* the reactive particle – that is, $\sum_i c_{ij}=1$. Essentially, the control network is a *bipartite graph*.

- **Command network:** we define a command hierarchy using the variables $h_{ij}=1$ if active particle $i$ is a commander of active particle $j$ (otherwise $h_{ij}=0$). In the hierarchical command, active particle can have only a single commander, - that is, $\sum_i h_{ij}\le 1$, - with a single “top commander” of the command network (for this node we will have $\sum_i h_{ij}=0$).
- **Communication and information flow network:** we define the ability of active particles to exchange information with other active particles using variables $n_{ij}=1$ if active particle $i$ can send information to active particle $j$ (otherwise $n_{ij}=0$).

The organizational networks of the particles may be changed over time. This will enable the structural and process adaptation – one of the key attributes of the agile organizations. A communication network may depend on the geo-spatial distribution of the particles – that is, on the ability of the particles to transmit the information (e.g., using wireless peer-to-peer communication the other particles may pose obstacles and the distance may change the ability to communicate information). On the other hand, command and control networks are defined more as “roles” – that is, the command and control relationships should not be changing significantly over time unless the organization is adapting to the environment – see discussion in the “Future Directions” section.

Different organizations (variables $<C,H,N>=<\|c_{ij}\|,\|h_{ij}\|,\|n_{ij}\|>$) would allow different control processes, trading off more distributed execution with higher levels of control over this process. This will result in different assembly execution times and even accuracy for some of the shapes. As the result, to achieve higher degree of shape assembly correctness and decrease the assembly time, the particle organization must be matched to (tailored to, congruent with) the shape node-link specification and corresponding shape temporal plan (defined in next subsection).

### *The Execution Planning*

The military organizations need to develop the plans to conduct their missions in the most efficient manner. Similarly, the shape assembly needs to be planned to allow for efficient distribution of responsibilities to the active particles. The desired shape structure $G_D=(V_D,E_D,A_D)$ will be built by the C2 assembly organization. To utilize the ability of the active particles to generate instructions and supervise the shape execution process in parallel, we can create a *shape assembly plan* that has two major components (Figure 5):

- **Shape decomposition** defined as multiple subsets $V_D^s$ of node set $V_D$ ($\bigcup_s V_D^s=V_D; V_D^s\cap V_D^r=\varnothing$).Note that accordingly, we can define a subshape $s$ as $G_D^s=(V_D^s,E_D^s,A_D^s)$, where $E_D^s$ are links among nodes in $V_D^s$ and $A_D^s$ are corresponding attributes of nodes and links. We can define the subshape $s$ using variables $u_{si}$, where $u_{si}=1$ if the node $i\in V_D$ is in the subshape $s$, that is $i\in V_D^s$, and $u_{si}=0$ otherwise. Then, $V_D^s=\{i\in V_D:u_{si}=1\}, E_D^s=\{(i,j)\in E_D:u_{si}\cdot u_{sj}=1\}$.
- **Shape temporal plan** defined as a precedence graph of subshapes $G_D^s$. This can be done using variables $p_{sr}=1$ if $G_D^s$ must be constructed before starting $G_D^r$ (and $p_{sr}=0$ if no such restrictions exist).

In Figure 5, a shape is decomposed into four subshapes. Figure 5c shows an example of the shape temporal plan, where precedence constraints define the temporal ordering between building the subshapes. In this example, the shape assembly will start with subshape $G_D^1$, then subshapes $G_D^2$ and $G_D^3$ could be assembled in parallel, and then a subshape $G_D^4$ will complete the shape construction. Such temporal constraints must be tracked over time, with subshapes assembly activated only when all its predecessor subshapes in the shape temporal plan have been constructed successfully. This monitoring is done by the active particles in the assembly C2 organization. We can assign the responsibilities of the subshape activation to the active particles that are supervisors of the active particles building the constituent subshapes (a subshape and all its predecessors). When active particle completes its subshape, it reports this status to the supervising active particle, which then determines if the next subshapes could be activated. For the example in Figure 4, Figure 6 shows the hierarchy of active particles and the activation responsibilities.

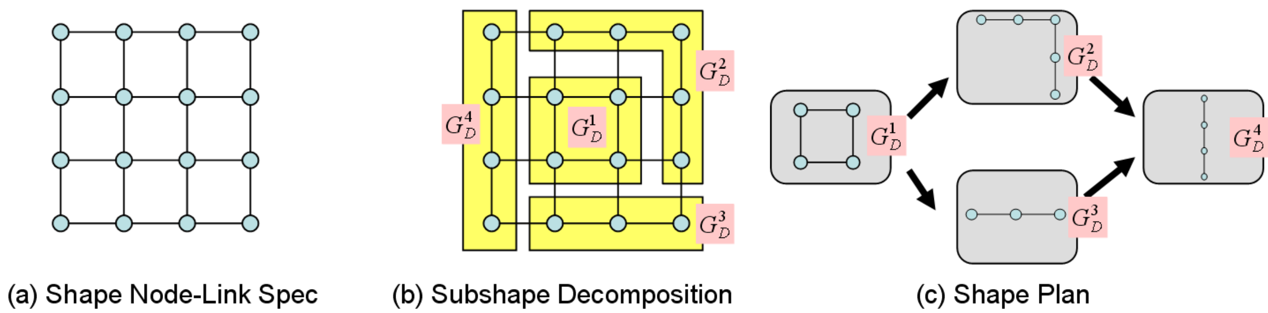



**Figure 5: Example of shape assembly planning**

A shape decomposition is used to assign the subshapes to active particles. We can define this assignment using the variables $x_{si} = 1$ if the subshape $G_D^s$ is assigned to active particle $i$ and $x_{si} = 0$ otherwise. In our work, we assigned only a single subshape to active particle, so that $\sum_i x_{si} = 1, \sum_s x_{si} \le 1$.

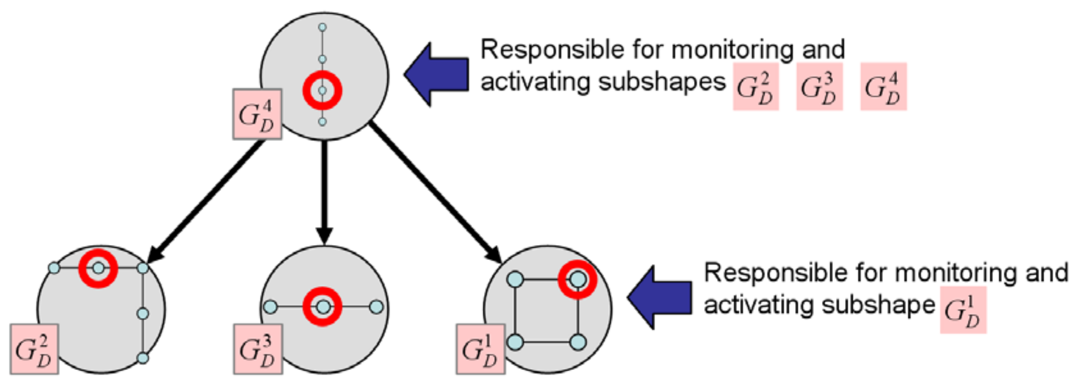



**Figure 6: Example of subshape-to-active particle allocation and activation responsibility assignment (the roles of active particles in the subshape are selected by active particles; in this figure, these roles are marked with red circles)**

Different shape decompositions can be developed, and each decomposition can be associated with multiple diverse temporal assembly plans. This feature of our framework is an example of the *flexibility* needed by the agile C2 systems, because it enables the C2 organizations to employ alternative execution strategies tailored to the resources and the situation while still achieving the original objectives.

### *The Execution Processes*

When the mission plan and the structure of the military organization are specified, tactical processes and rules need to be defined to execute this plan. Similarly in the shape assembly domain, we need to specify the rules for allocating the command and control roles of assembly to the particles and generating execution instructions during the assembly process. The nodes $V_D$ in a shape node-link specification $G_D = (V_D, E_D, A_D)$ are essentially a set of roles that must be filled by the physical particles. Any particle – active or reactive – may fill the role of the desired shape. As the result, we need to find a *role mapping matrix* $S = \|s_{kj}\|_{k \in V_C, j \in V_D}$, where variables $s_{ij}$ define the particle-to-role assignment. That is, $s_{ij} = 1$ if particle $i$ is mapped to (is assigned a role of) the particle $j$ in the desired shape network. When the roles are selected, the connections (joints) must be built. That is, for the two particles $k, m \in V_C$, if they are assigned the roles $i, j \in V_D$ (that is, $s_{ki} = s_{mj} = 1$), then there must be a joint between $k, m$ if $e_{ij}^D = 1$ with attributes $a_{ij}^D$ and no joint if $e_{ij}^D = 0$. When the joints are constructed successfully, we have: $e_{km}^C = e_{ij}^D, a_{km}^C = a_{ij}^D$. For a completely successfully built shape, we can write:

$$e_{km}^C = \sum_{ij} s_{ki} s_{mj} e_{ij}^D, a_{km}^C = \sum_{ij} s_{ki} s_{mj} a_{ij}^D .$$

As the result, two instructions must be generated: *role assignment* and *joint construction*. The first instruction helps the active particle maintain knowledge about the roles of its own and its controlled reactive particles. The second instruction is needed to execute connections by reactive particles.

## Automated Shape Assembly Process: Summary of Steps

In this section, we summarize the steps used for the shape assembly, starting with the shape definition and ending with the physical component assembly.

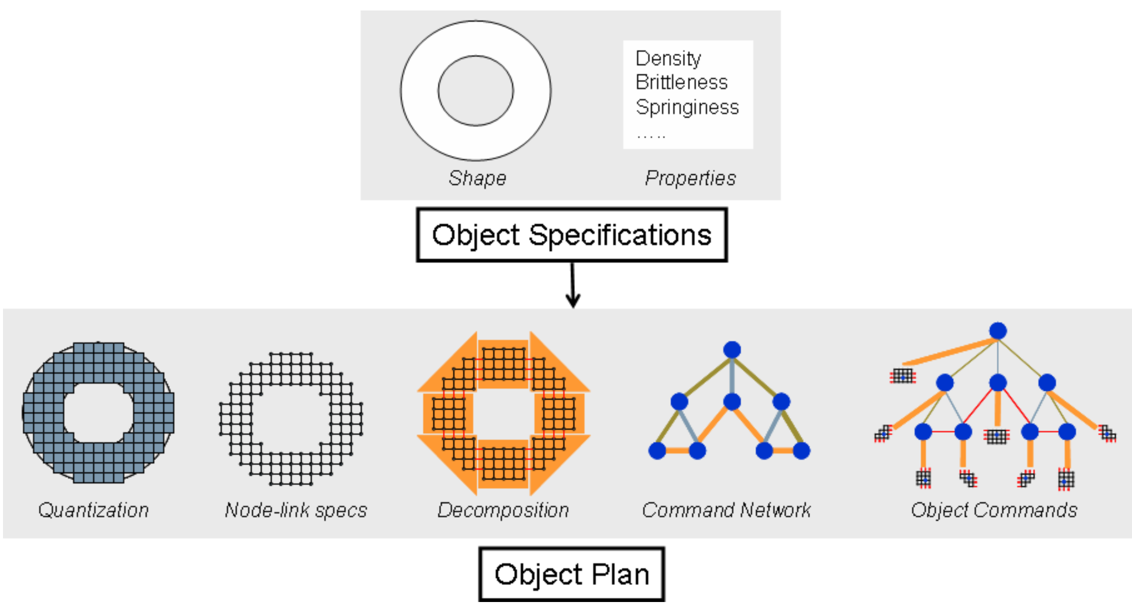



**Figure 7: Example of Shape Assembly Plan (off-line process)**

### *Assembly Planning*

According to the above, the following *assembly planning* is performed off-line to create the *object plan*. This process is defined in the following steps (Figure 7):

**Step 1:** Perform quantization of the shape to develop a 3-D component model

**Step 2:** Extract 3-D Graph from the component 3-D model and node-link specification

**Step 3:** Conduct shape decomposition and develop shape temporal plan

**Step 4:** Design the C2 organization to support the shape assembly, including command, control, and communication networks

**Step 5:** Assign the shape assembly plan elements (the subshapes) and the subshape activation responsibilities to the active particles in C2 organization for the assembly execution

After assembly planning, the C2 organization is ready to start execute the shape assembly. Note that some of the steps above can be performed jointly to improve the efficiency of the product solutions (e.g., steps 4 and 5).

### Assembly Execution

The following *assembly execution* is performed on-line to create the *shape*. This process is performed by active particles and defined in the following steps (Figure 8):

**Step 1:** Assign unfilled roles in the subplan to the active particles

**Step 2:** Determine the remaining set of unfilled roles

**Step 3:** Allocate the remaining unfilled roles to a subset of available and unused controlled reactive particles.

**Step 4:** Generate connection (joint) instructions based on mismatch between the current state of the shape and desired shape, and send these instructions to reactive particles

**Step 5:** Update the subshape unfilled roles

**Step 6:** When the subshape is finished, report to the active particle monitoring its success; if an active particle receives a report of subshape completion, update the successor subshapes in the shape temporal plan and activate the subshapes if possible.

**Step 7:** When the subshape is finished, connect/fuse this subshape with existing (already constructed) subshapes

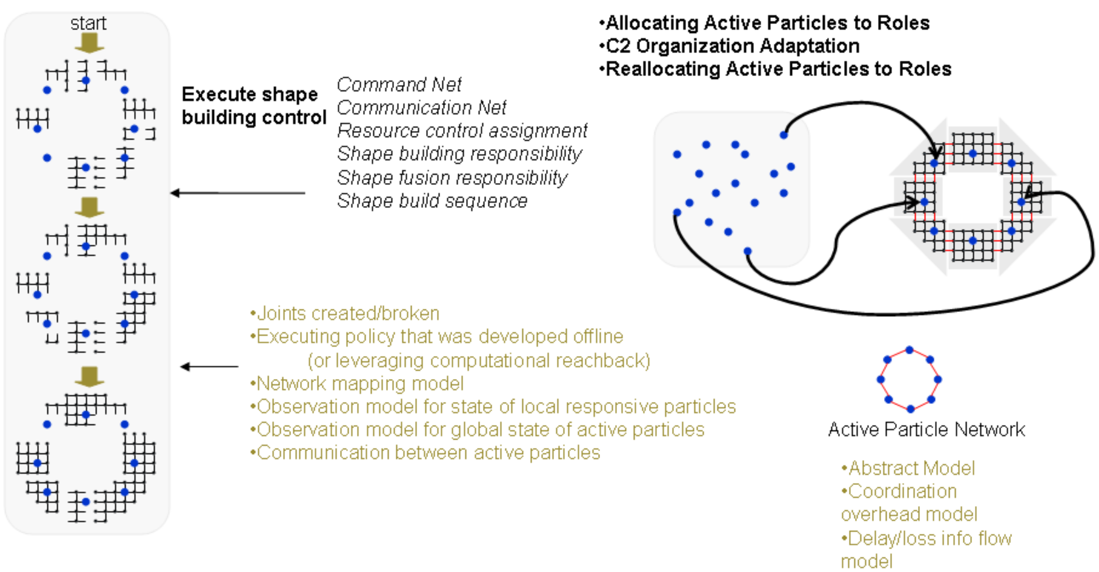



**Figure 8: Example of Shape Assembly Execution (on-line process)**

The process described in previous steps and illustrated in Figure 8 enables the agility of the shape assembly execution through the following features of the system. First, it is robust to the particle specification and only requires the ability to quantify the assembly plan and its contingencies. It is resilient to the changes in the environment, as it allows substituting the active and reactive particles that fail to execute their instructions. Since the execution is carried out by active particles and the process gives significant autonomy to them, our framework enables high responsiveness during the assembly execution while at the same time supporting building very complex shapes. In addition, the assembly commands can be created with various degrees of randomness in how and what reactive particles are selected for the assembly; this will create the innovation that is often required to avoid unforeseen circumstances, especially in situations of parallel and competitive assemblies. Since the plans and subplans can be changed over time without the time-consuming reorganization procedures, this supports yet another attribute of the agile C2 organizations – the flexibility of the shape assembly. Finally, we allow the particle organization to be changed over time; supporting the change of organizational structure, as well as the execution workflow as described above, will enable the full adaptation functionality that is needed by agile C2 systems.

# Modeling Algorithms in Cognitive Particles

In this section, we describe specific models we used for planning and execution of the shape assembly. We draw on these algorithms from the military organizational structure and execution policy design work that dealt with various aspects of performance and processes in distributed execution settings.

## *Shape Decomposition and Control Network Design*

*Due to the computational complexity, the shape decomposition and control network design is performed offline.* The decomposition of the object node-link specification $G_D = (V_D, E_D, A_D)$ into a set of subshapes $G_D^s = (V_D^s, E_D^s, A_D^s)$ can be obtained manually, but the complexity of decisions about such decomposition for large-component object prevents the user from constructing the composition in real time. Instead, we investigated the automated decomposition approaches that trade-off three main variables:

- **Internal workload of active particles:** as the subshape assembly must be executed by the active particles, the workload of the subshape assembly (e.g., the number of nodes in the subshape and the joints/links that must be constructed) becomes important. This is due to the limit on the memory and computational power that active particles may possess. More formally, the internal workload for the subshape $G_D^s = (V_D^s, E_D^s, A_D^s)$ is defined as the weighted sum of the nodes and links in the shape: $w^I(s) = w_{link} \sum_{i,j} u_{si} u_{sj} e_{ij}^D + w_{node} \sum_i u_{si}$ , where $w_{link}, w_{node}$ are the loads of building single link and monitoring single node respectively.
- **External workload of active particles:** the subshapes must be “fused” together to form the shape. Such fusion must be conducted by coordinating between the active particles that control reactive particles that must have joints constructed between them. More formally, the external workload for the subshape $G_D^s = (V_D^s, E_D^s, A_D^s)$ is defined as the weighted sum of the links with other subshapes: $w^E(s) = w_{link} \sum_{r \neq s} \sum_{i,j} u_{si} u_{rj} e_{ij}^D$
- **Complexity of subshape sequencing:** the subshapes must be sequenced to enable the fusion to occur. Some decompositions result in efficient parallelization of the subshape construction process, while other decompositions may result in the sequential shape building and as the result higher cost and delays of the construction process. The subshape sequencing is addressed in the next subsection describing shape temporal plan design.

In the above, the notion of “workload” is introduced to model the coordination among active particles:

- *Internal coordination* – to control the assembly of subshape managed by an active particle; and
- *External coordination* – to fuse its subshape with subshapes constructed by other active particles.

The shape decomposition results in the total workload of the subshape assemblies equal to $w(s) = w^E(s) + w^I(s) = w_{link} \sum_r \sum_{i,j} u_{si} u_{rj} e_{ij}^D + w_{node} \sum_i u_{si}$ . We can then define the objectives or constraints for shape decomposition based on *balancing* or *constraining* these workloads. Such balancing or constraining is required due to limited memory and computational power at the active particles.

As the result, the shape decomposition can be posed as an optimization problem: we need to find a clustering of the nodes of the shape that achieves some optimization of the inter- and intra-cluster properties. One example of such problem is to minimize the squared sum of subshape workloads $\min_{u_{si}} \sum_s w^2(s)$, while another example is to maximize the entropy $\max_{u_{si}} \sum_s \frac{w(s)}{\sum_r w(r)} \log \frac{w(s)}{\sum_r w(r)}$. Both problem formulations would result in balancing the workloads of subshapes assembly. This optimization can be carried out using non-linear optimization techniques, with barrier functions and Lagrangian relaxation providing the most efficient solutions. In our work, we used *Tabu search algorithm* that iteratively found a sub-optimal subshapes using the “manipulations” of the solution to move to another solution. Many manipulations are possible; instead, we focus only on a limited manipulation set that allows simple update and objective/constraints recomputation:

1. Change of assignment of node to a different subshape
2. Swap assignments of 2 nodes between their subshapes
3. Crossover in assignment vector of two subshapes, which results in swapping of the assignments of multiple nodes

The Tabu algorithm maintains a list of assignments that must not be changed for some period of time. It is also allowing (with a small probability) the manipulations of the solution to occur that result in degradation in the value of objective function, which allows the search to avoid local optimums.

## *Shape Temporal Plan Design*

In this section, we describe how the shape temporal plan and subshape fusion temporal constraints can be generated. *Due to the computational complexity, the shape temporal planning design is performed offline.* First, we note that the requirement for sequencing the shape construction comes from the situations in which one subshape is “inside” another subshape. Three examples of this situation, with four subshapes color-coded, are shown in Figure 9. In both examples, subshape A is inside subshape B. In Figures 9a and 9b, the assembly of subshape A does not have to precede assembly of subshape B, because both subshapes can be constructed in parallel and then subshape A can “slide into” subshape B (see Figure 10). This is not the case with example of Figure 9c, in which if the subshapes A and B are constructed in parallel, A cannot be fit into B and thus B would have to have a part of it disassembled. We thus require, to avoid unnecessary disassemblies, to construct the subshape A first, and then continue constructing the “surface” of subshape B by first creating the joints of particles of A and B. That is, the “fusion” of A and B must start before the construction of subshape B. One of the ways to do this is to construct a single “external” link (joint) from a node in A and node in B, and then proceed iteratively (the iterative construction approach is described in more details in “Iterative Role Selection” section).

Also note that there are no requirements to sequence the subshape fusion in Figure 9a, because all fused shapes can similarly “slide into” the other subshapes. This is not the case for decomposition examples of Figures 9b and 9c: if B is fused with D, or A is fused with D, the other subshapes cannot be moved to fit the structure.

Only the example in Figure 9c requires the sequencing of the shape construction. Whenever this happens, we require the fusion joints to be constructed first. While in other circumstances some sequencing of subshape fusion is not necessary, we simplify the planning process by introducing “by design” temporal constraints. In our model, if otherwise not specified, the fusion will occur after the individual subshapes have been assembled. That is, we only constrain the necessary temporal fusion sequences, and allow other fusion to occur opportunistically.

Our framework can incorporate the construction and planning of a diverse set of models, which makes it robust to the requirements of the assembly. Our process can produce multiple shape decompositions and diverse set of assembly plans, thus improving the agility of the shape assembly in terms of flexibility of the planning and execution processes.

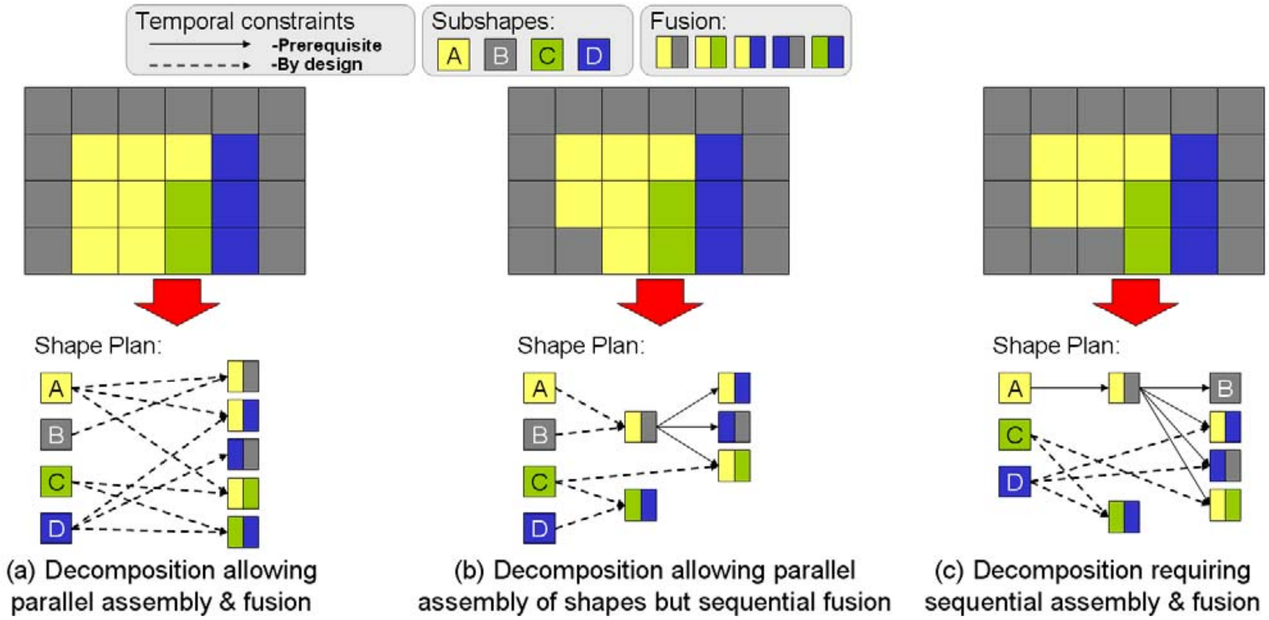



**Figure 9: Two Examples of Alternative Shape decompositions requiring Different Shape Plan Temporal Constraints**

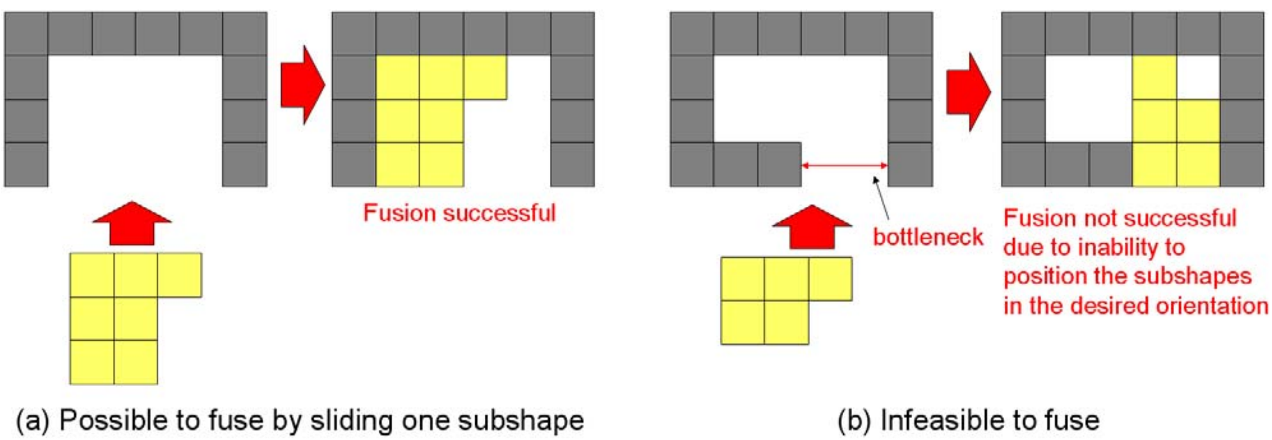



**Figure 10: Two example of the fusion of two subshapes**

### *Subshape Assembly Execution: Iterative Role Selection*

The subshapes are assembled on-line: active particles conduct necessary computations, updates, generate decisions, and communicate them to each other and to reactive particles. Both types of particles then execute the instructions by performing *move* and *connect* actions.

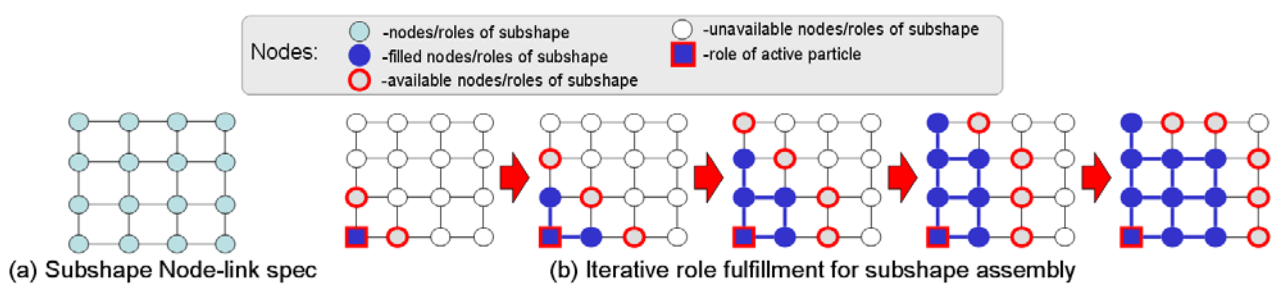



**Figure 11: Example of Iterative Subshape Assembly**

When the assembly of subshape allocated to active particle is activated, its construction proceeds iteratively as follows. Initially, an active particle selects a ***role*** (a node in the subshape specification) for itself. Then, the algorithm iteratively determines the next available roles to fill, and selects reactive particles for these roles. The role is said to be *available* if it must be connected to a particle that is *finished*, - that is, there exists a joint in the node-link specification of the subshape between this role node and the role node of the finished particle. A particle is said to be finished if is has a role selected and all the joints have been formed. The status of roles is updated, and this process is repeated for the next set of "available" roles in the shape plan. An example of this process in 2-dimensional space is shown in Figure 11. While all available particles are considered at every iteration, not all of them are fulfilled and assembled.

When a set of available roles is selected, the algorithm selects the particles to fill these roles based on a probabilistic assignment algorithm. This algorithm considers all available reactive particles (reactive only – since the active particles have already been selected in the first iteration) that are controlled by the active particle. First, we calculate a set of values $d_{kj}$ for each pair of available role in the shape temporal plan $j \in V_D$ and a reactive particle $k \in V_C$. We do this based on the current $x_k, y_k, z_k$ position of the active particles that already have assigned roles (nodes of the subplan) that have links with node $j \in V_D$ in the subshape $G_D^r$. We calculate the average distance of the active particle $k \in V_C$ to all active particles that already are assigned the roles from the subshape $G_D^r$ (determined by role mapping matrix $s_{kj}$):

$d_{kj}(r) = \frac{1}{\sum_{i \in V_D} e_{ij}^D} \sum_{m \in V_C} \sum_{i \in V_D} e_{ij}^D s_{mi} u_{ri} \left( (x_k - x_m)^2 + (y_k - y_m)^2 + (z_k - z_m)^2 \right)$. Alternatively, we could have calculated the exact position of where the new role should be located based on already filled roles (reactive particles) and the lengths and orientations of the connections.

We then minimize the objective equal to the summation of distances $\sum_{k \in} d_{kj}(r) s_{kj}$, which is equivalent to the *assignment problem formulation*. However, the assignment solution, while optimal at the time of the distances calculation, will quickly lose optimality since the particles move almost constantly. In addition, the assignment algorithm is of polynomial complexity, and we were looking for a linear complexity real-time solution. As the result, we decided to avoid using the assignment algorithms (such as *auction algorithm*) and use instead the randomized assignment, which is selecting a 0-1 matrix using the distances as probabilistic weights. Such an approach can be viewed as *multi-dimensional soft-max*. The resulting assignment was of linear complexity and provided solutions that were robust to particle movement.

Since the subplan execution is carried over independently by active particles, and since only after the subplans are constructed the subshape fusion will occur, these features improve the agility of the shape assembly by making the shape constriction highly responsive to the local changes in the environment, such as failures of reactive particles to move and connect. Various local rules and randomization procedures can be used to improve the effects of control in uncertain situations, which resemble innovation mechanisms in agile C2 systems.

## *Active Particle Command Network Design*

To avoid decision-making confusion associated with the distribution of control, military organizations impose a *command structure* (i.e., *superior-subordinate* or *supported-supporting relations*) on their team members. One of the goals in creating a specific command structure is to match the induced superior-subordinate relationships among commanders with the coordination required to complete the mission. Different definitions of this matching lead to different formulations of the organizational command structure design problem.

For shape assembly controlled by the organization of active particles, we employ the same formalisms used for military command and control. The simplest command structure is a hierarchy with a single commander "root" active particle and all other active particles being subordinates to it. The problem with this setup is the overload of the monitoring and conflict resolution that will be imposed on the root active particle. Instead, we want to design the command structure among active particles to match the assembly coordination required among them. That is, the command structure must match the shape decomposition and plan designs.

We consider the situation when the coordination between any two active particles needed during shape construction requires the participation (e.g., monitoring, status update, approval, information passing, etc.) of all active particles involved in the corresponding command (superior-subordinate) path spanning two coordinating particles. That is, this accounts for passing command-related information only via command structure network links, such that each active particle can communicate command-related messages only with its immediate superior/subordinate particles. The associated *coordination overhead* adds the extra load to each active particle involved in the decision cycle.

In our research, we limited the command networks topologies to a tree structure – enforcing a natural hierarchical relationships for commanders (each commander has at most a single superior, and only one "root" commander does not have a superior). Such a command structure sometimes is referred to as a tree. In this case, if $R_{s,r}$ defines the coordination requirements among active particles (e.g., this can be defined based on the subshape fusion required to be coordinated by the active particles, in which case $R_{s,r} = w_{link} \sum_{i,j} u_{si} u_{rj} e_{ij}^D$ ), then $\boldsymbol{O}(m) = \sum_s \sum_{r>s} R_{s,r} \cdot \mathbf{1}(m \in \text{path in T from } s \text{ to } r)$ defines the coordination overhead for active particle $m$ ("path" is found in the command hierarchy tree $T$, where a single path exists between any two nodes). The coordination overhead load is redundant and could potentially be avoided using a different command structure configuration.

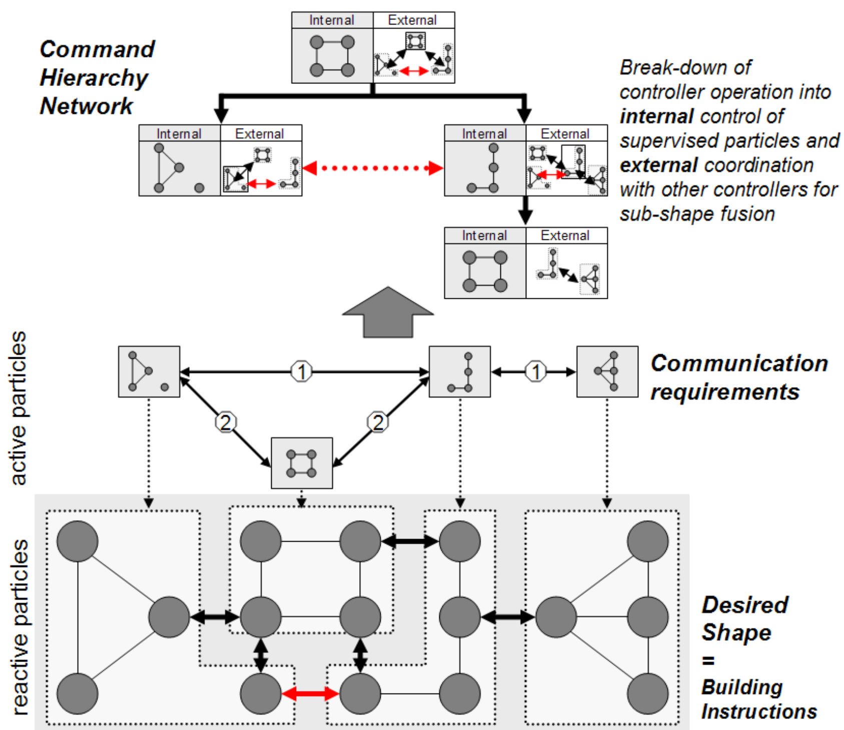



**Figure 12: Example of Processes in Command Structure Induced by Shape Decomposition (Red arrows indicate the overhead coordination)**

As the result, we are interested in designing the command hierarchies that minimize the total overhead coordination in the C2 organization computed as $\sum_m \boldsymbol{O}(m)$. Such command structure can be found suing the *minimum coordination cost tree* design algorithm which uses the max-flow (min-cut) approach to cluster the active particles and generate the tree structure. This algorithm has been used successfully for military C2 structure design in several previous projects for Navy and Army alternative organizational design analysis. We refer the readers to the following papers for the detailed algorithm description (Levchuk et al., 2002; 2006). In the simulations we describe in later sections, we have experimented with several command structures. An example of how the shape decomposition interacted with command structure is shown in Figure 12.

Our framework allows dynamically matching and adapting the structures of the particle organization to the evolving situation and assembly requirements. This process resembles the agile military command and control that can introduce the C2 and plan changes in the face of changing situation on the ground. It is important, however, to maintain the correspondence between the C2 structures and the plan decomposition because it will improve the ability of agile organizations to perform complex tasks and missions. The process described in Figure 12 enables us to maintain the correspondence between the formal organizational networks and the assembly state, and thus monitor the deviation of the assembly process and environment and make quick and efficient adjustments to the construction process as needed to improve the performance of complex shape assembly.

### *Active Particle Communication Network Design*

The communication network must also be designed to match the coordination among particles and corresponding communication of command-based and other types of information. This can be achieved using the optimal network design concepts to minimize the delays during communication (Levchuk et al., 2002; 2003; 3004; 2006). The reason for “designing” the communication structure is to avoid the overload that may occur when all one-to-one communication channels are open and to limit the memory required to store communicated information.

In our research, we also looked at the dependencies of the communication structure on the spatial positions of the active particles. The communication bandwidth could be a function of the distance between particles and the objects between them that become obstacles to wireless transmissions. We have also looked at the wired communication network design. For the latter, the constraints on the number of communication links and their bandwidth are even more essential than for the wireless communication networks.

### *Shape Plan Execution*

Using the shape temporal plan variables $p_{sr} \in \{0,1\}$, we start defining the *in-degree* of the subshape $G_D^r$ as $n(r) = \sum_s p_{sr}$. Initially, this variable is equal to the number of the subshapes that must be constructed immediately before $G_D^r$ can be started. As subshape $G_D^s$ is finished, we update the in-degree parameters $n(r) = n(r) - p_{sr}$. If $n(r) = 0$, the subshape $G_D^r$ is activated for construction. The activation and parameter updates happen at active particles who are assigned the responsibility for subshape $G_D^r$ activation.

## Results

In this section, we describe the prototype of shape assembly we have designed in Cognitive Particles project, and present the results from computational experiments. We present an example of the analysis we have conducted assessing the agility of our assembly framework.

### *Simulation Testbed*

To create an environment for testing, validating and comparing the shape assembly theories and algorithms, we developed the simulation testbed. This testbed allowed us to develop principles relevant for guided, reversible shape assembly formation.  As we began design of the testbed, it became clear that it would be helpful to be working in 3D with realistic physics.  We therefore integrated Java-based AnyLogic prototype modeling environment from XJ technologies with two packages: (i) a package called Irrlicht 3D (for 3D modeling and visualization), and (ii) a package called Newton Dynamics (for physics simulation). These packages added 3-dimensional representation, collision detection and physics-related behaviors.

The high-level architecture design for the testbed is shown in Figure 13. The testbed consisted of four main components:

- The **Shape Object Specification** component was implemented as a database holding the node-link specifications of desired shapes and the shape assembly plans (shape decomposition and temporal plans);
- The **Particle Simulation** component, based on the Newton Dynamics physics engine, provided a realistic constructive simulation of the particle environment with physics-based effects (collisions, gravity, friction, joints, etc.) and tactical instruction execution (force-based particle shake, movements of the particles to create joints, collision analysis, callback for execution events, etc.).
- The **Control Simulation** component implemented the C2 design and distributed C2 assembly construction functionality. C2 structure design algorithms can be used for designing command, communication, and control networks of active particles. The user can also specify their own C2 structures. The actions for active particles to construct their sub-shapes and coordinate subshape fusion were defined using iterative role selection algorithm. The instructions were communicated and coordinated using the active particle C2 structures (command, communication, and control networks).
- The **User Interaction** component had 2-D viewer & controller (2-D layout, simulations and manual interaction controls, and measures), and 3-D viewer (3-D rendering of assembly based on Irrlicht engine).

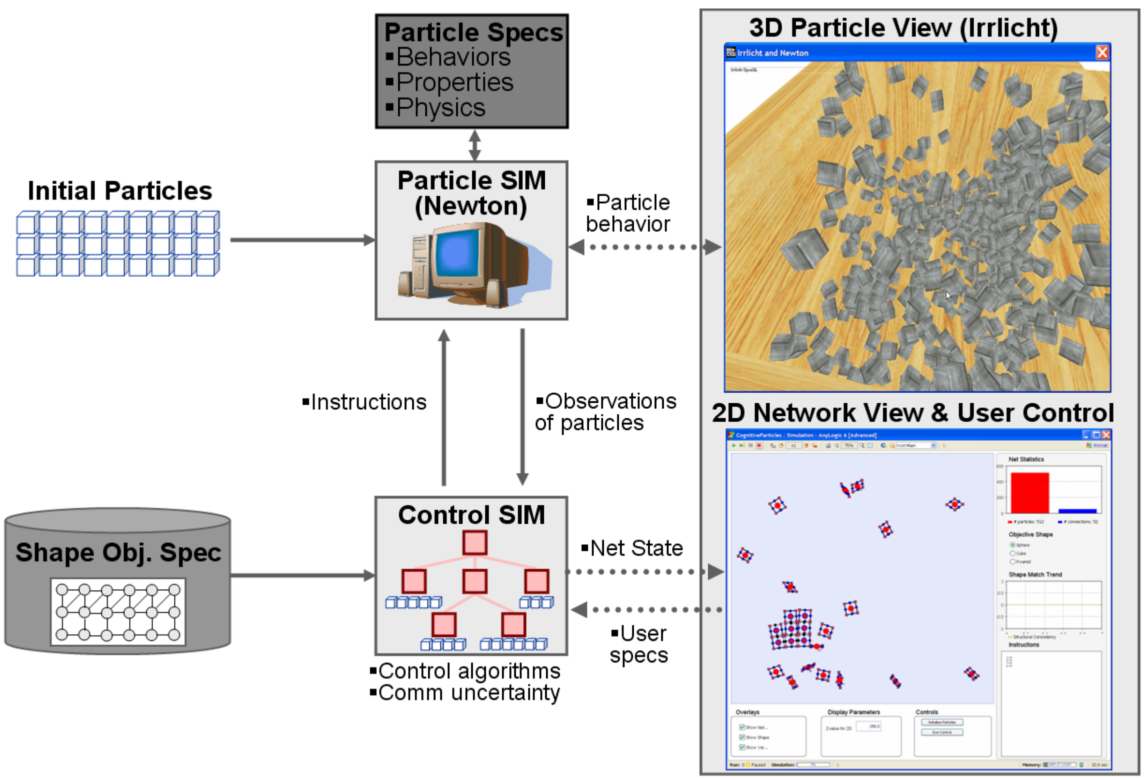



**Figure 13: Cognitive Particles testbed components**

We implemented the elementary particles as 3-D cube shapes with additional state representations. We have modified the particle class to store the information about its physics, visualization representations, organizational responsibilities, and connection properties. Command, communication and control networks were specified using directed graph data structures. These C2 networks were instantiated and updated over time using publish-subscribe mechanism; these networks enabled information routing and message queuing. The instruction messages to create/delete particles, change their states (e.g., color), and create/delete joints among particles were passed from Control SIM to Particle SIM and queued for execution. The Particle SIM implemented several methods, including generation of particles set, particle shake, gravity control, coloring, and joint formation. As the result, the simulation provided a platform with on-line dynamic shape assembly visualization (Figure 14).

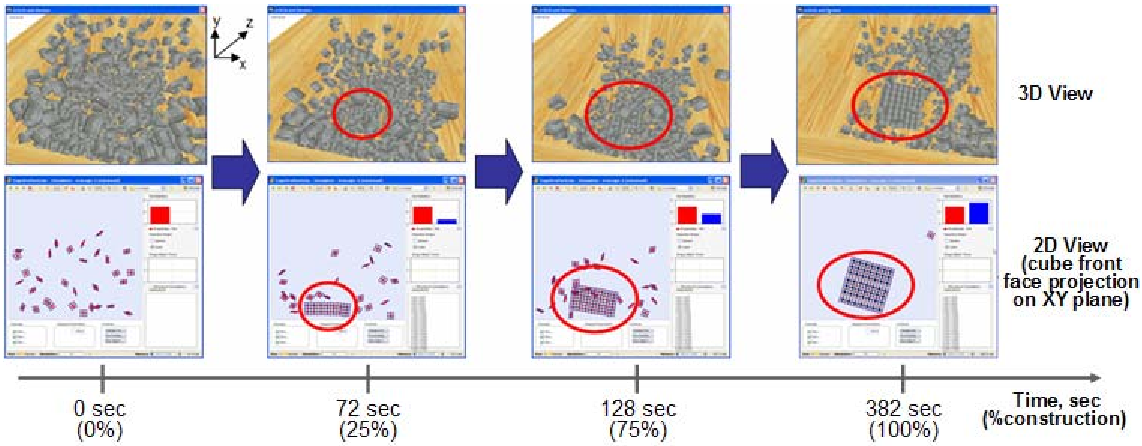



**Figure 14: Example of cube shape assembly in Cognitive Particles testbed**

### *Experimental Metrics*

To truly assess the agility of the particle assembly framework, we need to understand the impact that various challenges occurring during the shape formation might have on the success of the assembly. As such, our main metrics included measures of performance of the assembly in terms of time and match (accuracy) of the assembled shape and the objective shape, and the measures of the process in terms of the expended resources and energy. Thus, to evaluate effectiveness of different assembly models and processes, – including varying designs of C2 networks, plans, and execution rules, – we conducted experiments using the following set of metrics:

- **Metric 1 --- Timeliness:** Time to complete execution of shape plan
- **Metric 2 --- Accuracy:** Differences between currently assembled object and desired shape plan. In our experiments, we measured the *percentage completion* instead of accuracy, because we have the exact match of the particle ID's to the shape specs when the shape assembly occurs.
- **Metric 3 --- Resources:** Amount of assembly resources which represent the cost of control in terms of manufacturing the required components. We computed this metric as the number of parallel channels of execution, i.e. number of active particles performing commanders' roles in C2 particle organization)
- **Metric 4 --- Energy:** Energy expended by particles to execute the assembly, which represents the cost of control to maintain the execution process.

While robustness and resilience features of the agile process can be formally measured by assessing the sensitivity of the assembly to the various external conditions (e.g., failures of the particles, physical forces, communication uncertainty, varying of the assembly requirements, etc.), the other features – i.e. responsiveness, flexibility, innovation, and adaptation, – are only enabling factors to achieve better performance scores and cannot be objectively measured. This is due to the fact that to achieve agility, a key trade-off between the time of the decision and its quality must be satisfied. The final objective is not just to change, or to produce the best possible decision, but to make the right changes at the right time. As the result, we decided to proceed with assessing the performance differences between various assembly configurations. In our current research, we are developing the procedures to identify required agility features for various characteristics of the resources, environment and assembly objectives.

### *Experimental Hypotheses and Run Setup*

The objective of the assembly research we have conducted in this project was to develop an automated intelligent assembly control framework, including a model and a testbed that would achieve a feasible and efficient construction of shapes of interest. We started our analysis with simple shapes – including cube, sphere, pyramid – and moved to more complex shapes (e.g., wrench) that could have functional components.

The main hypothesis of our research was that for efficient automated shape assembly, there needs to be a *match between shape decomposition, temporal plan, the C2 particle organization, and assembly metrics* (Figure 15). The notion of the “match” between these components is know as the *congruence concept* in military and socio-technical command and control organizational analysis (Levchuk et al., 2002; 2003; Kleinman et al., 2003). While possessing the same “intelligence” (algorithms for developing assembly instructions and communicating them to the physical world), the mismatch between these components might result in waste of resources or energy, and delays or failures of shape assembly.

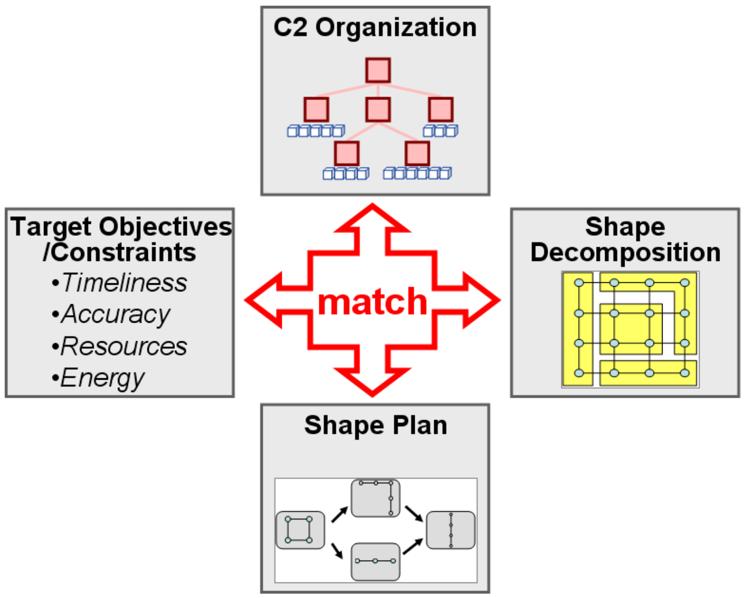



**Figure 15: Conceptual Representation of the Correspondence among Assembly Elements**

The match between shape specs, plan, organization and metrics must be dynamic because (i) the shape construction changes over time, and (ii) the environment impacts what shapes can be constructed and in which order. In the example shown below, we were able to develop the assembly designs that required no changes in the particle networks over time and only minor changes to the shape assembly plans, while the “tactical” execution of the assembly plan was the largest source of the assembly adaptation.

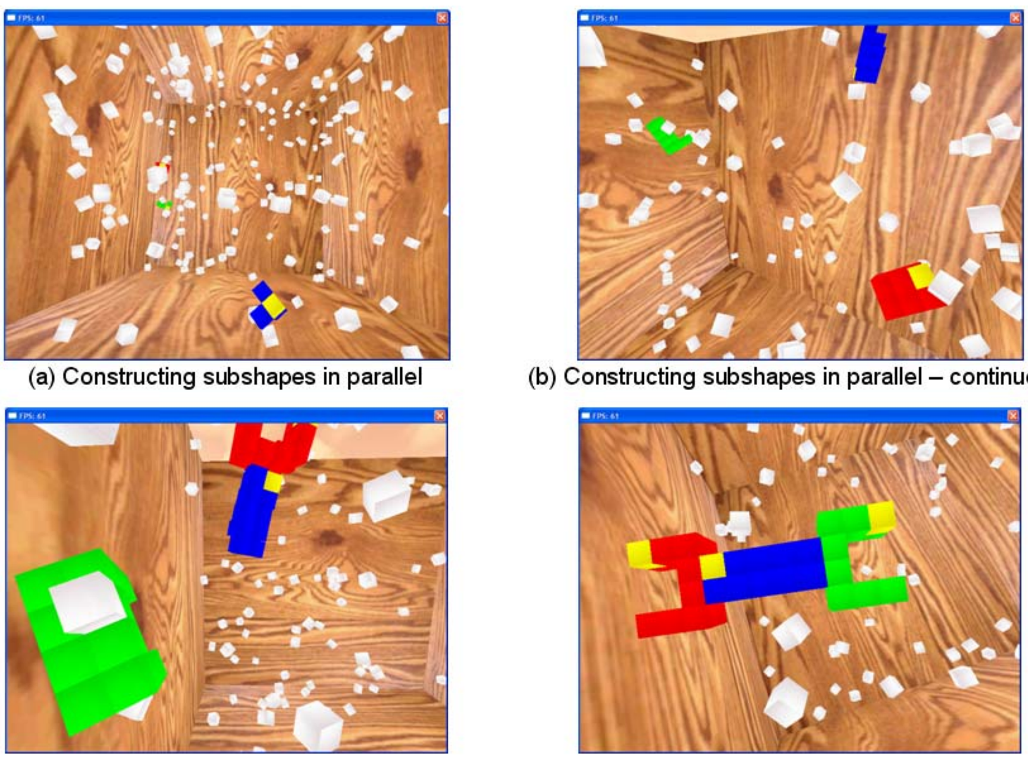
We have conducted several simulation runs, results for which are described below. For each run, the uncertainty model (probability of success of an instruction communicated to a subordinate) was fixed at 80%. For every run, there were fixed inputs for the shape decomposition and shape temporal plan. The number of active particles (commanders in C2 network) was fixed to equal the number of subshapes in the decomposition. We fixed the organizational command structure intentionally to be “flat” (i.e., a single superior commander active particle and all other active particles being subordinate to it). The communication structure was complete (all-to-all) with only geo-spatial restrictions (distances, obstacles). We thus compared the success of shape assembly under different shape decomposition, shape planning, and resources (number of active particles) constraints. In the following, we describe the example results of the computational assembly experiments using the measure of the percentage of shape assembly completion.

### *Example of Shape Assembly*

In Figure 16, we show an example of the shape assembly for a complex wrench object. The example illustrates how the shape assembly is happening over time by following the shape temporal plan and conducting distributed parallel subshape assemblies and subshape fusion activities by active particles. In this example, there are three active particles indicated with yellow color cubes in Figure 16. Each active particle is controlling 100 reactive particles. The active particles are constructing three subshapes indicated with distinct colors: red, blue, and green. White particles indicate reactive particles available as resources for the shape assembly if needed, but which are not currently part of the shape (that is, they have not been selected to fulfill shape node roles and not connected to the shape structure).

### *Comparing Assembly Strategies*

For the example sensitivity analysis of different shape assembly strategies, we developed three decompositions of a wrench (Figure 17). The first is a wrench controlled by one active particle, in which the construction is sequential (Figure 17a). The second is a wrench decomposed into two equal halves, each controlled by one of the two active particles and then fused together (Figure 17b); in this decomposition, both parallel and sequential shape assembly are possible. In the third decomposition (Figure 17c), the three active particles each take one part of the wrench to construct in parallel and then fuse the subshapes together; in this decomposition, we only analyzed a parallel shape assembly by each active particle and consequent shape fusion.



(a) Constructing subshapes in parallel

(b) Constructing subshapes in parallel – continued…

(c) Subshapes completed

(b) Subshapes fused – shape assembly completed

**Figure 16: Assembling a shape of wrench (Figures *a* and *b* show each of the active particles constructing their own subshapes; Figure *c* shows that each active particle finished its subshape and is attempting to fuse its subshape with others’; Figure *d* shows the completed wrench)**

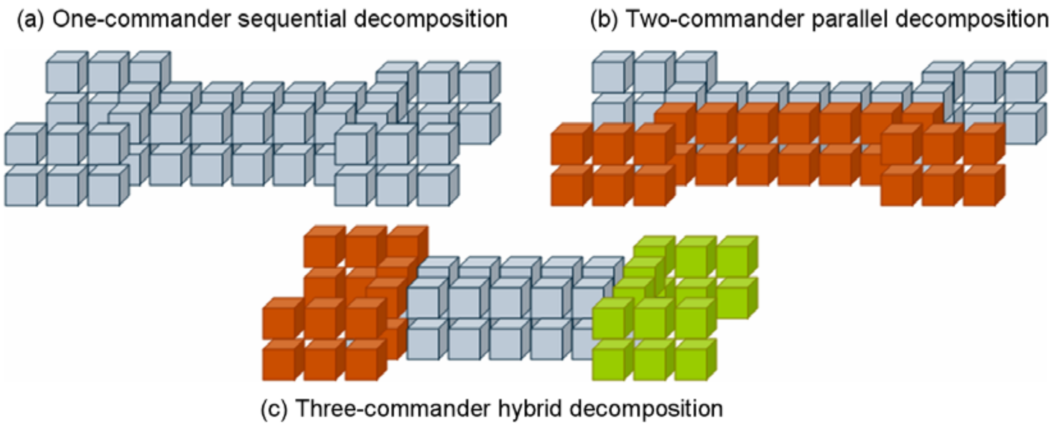



**Figure 17: Wrench shape decompositions**

For these experiments, we initialize the bucket of particles with 300 particles. In the following, we show an example of the comparison of the following 5 assembly strategies:

- *Strategy 1:* Single active particle (sequential shape construction)
- *Strategy 2:* Two active particles with sequential shape construction
- *Strategy 3:* Two active particles with parallel shape construction
- *Strategy 4:* Three active particles with parallel shape construction
- *Strategy 5:* Three active particles with parallel shape construction incorporating hake” operation

Figure 18 shows example of the results of the average assembly completion over time. The percentage completion is based on the ratio of executed shape joints to the objective number of joints. The completion of above 70% indicated that most of the shape has been successfully constructed while some of the connections (joints among particles) might have not been completed. The lower metric indicates either a delay in the assembly process, in which case the assembly can be completed given more time, or a possibility that the shape assembly is broken, which may happen when the particles are stuck in the physical world and unable to execute the connection commands. The latter happened when the pieces of the assembly were stuck against the walls of the bucket, other subshapes, or other particles, and the commands for the movements translated into the forces have been incorrectly calculated.

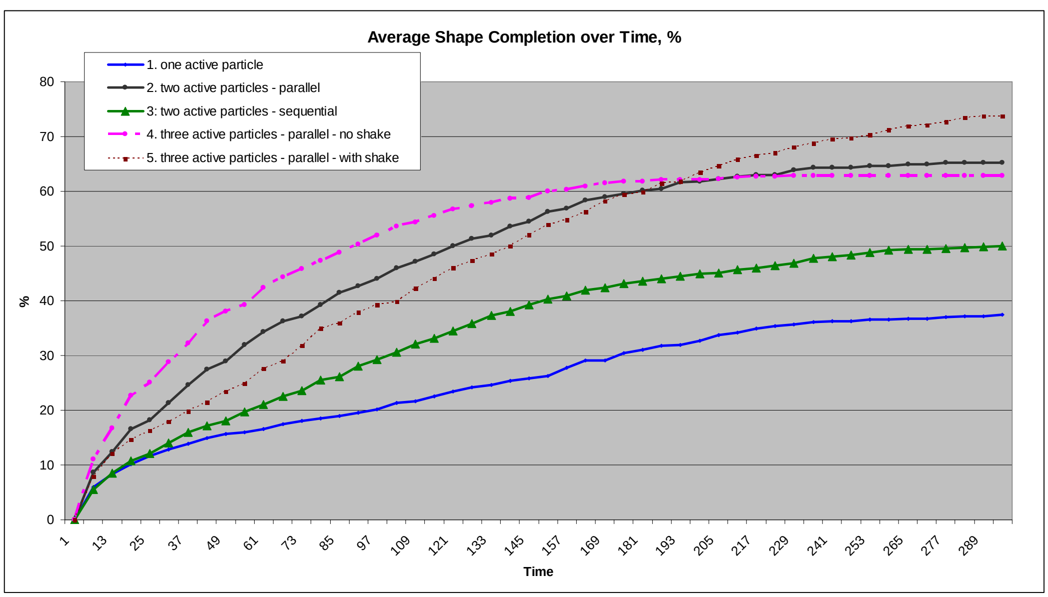



**Figure 18: Example of the comparison of assembly strategies using the metric of assembly completion**

From these results, we have the following observations. First, a single active particle had trouble constructing the 52-particle shape plan, and in most of the runs, the construction process was stalled. When the second active particle was added, the shape construction improved, with significant increase of the average shape completion rates when the two active particles have conducted their assemblies in parallel. However, the addition of the third active particle had little or no impact. The assembly without shake function was better in the initial stages of assembly than the 2-active parallel assembly, because of the parallelism of the assembly process allowed for fasted formation of the constituent subshapes. However, at the end the three subshapes often got stuck and could not be fused together, resulting in many stalled simulation runs and consequent lower average shape completion rate. Finally, the shake operation that we uniformly applied in the final assembly strategy reduced the initial rate of the completion by redistributing the particles further from the intended assembly points and thus slowing the joint formations, but on the other hand the shake has improved the final completion rates by removing the stalling situations during the fusion of assembled subshapes. To achieve efficient construction it is necessary to balance interdependent forces of random shake, gravity, and magnetic propulsion.

## Relevance of Results to C2 of Semi-Automated Forces

These results have the following consequences for the command and control of semi-automated forces. Clearly the ability to conduct the tactical operations in parallel may improve the completion time of the mission. Parallel execution is also needed in self-synchronized operations. However, it introduces the following challenges. First, the self-synchronized parallel mission execution requires additional command elements. The availability of commanders, similar to the availability (and concomitant cost) of the intelligent active particles, might be a constraint for the specific force composition. Second, the decomposition of the mission into tasks that can be executed by commanders completely independently is often not possible; instead, the feasible local parallelization of individual operations of command cells introduces the requirements for complex pre- or post-coordination among corresponding commanders – including the temporal and synchronization dependencies. For example, the fusion of the subshapes required after each subshape is assembled in parallel by a responsible commander can become a very complex operation and not feasible to execute using standard self-synchronization mechanisms – see Figures 9-10 for the example of the assembly complexity and potential infeasibility of the assembly sequencing operation. Oftentimes, self-synchronization is challenging because in complex situations it is hard to have the command elements possess the same situation understanding and perceptions. The challenge of unity of command in such situations has not been resolved, because the unity of effort has been hard to implement when the overall mission is not decomposable into simpler operations. However, the parallelism of task execution and the complexity of the resulting mission-task plan can be balanced optimally for each specific situation, and this balance must be incorporated in the agile C2 principles

Third, similar to the introduction of the shake operation in the particle assembly, new communication technologies and tactical coordination procedures might have different impacts of the success of the mission execution, and must be used targeting a specific need during the mission execution. Finally, adaptive mission execution is needed often in the situations when the original plan can no longer be successfully followed. A similar planning-execution framework we have described in this paper can be enhanced with a full close-loop solution, updating the plans, organization, roles, and tactical coordination rules over time as the mission progresses.

## Lessons Learned

A number of different lessons and have been learned as a results of the Cognitive Particles project:

- *Trade-off between online and offline planning:* Performing offline planning is favorable due to limited computation power. We can design the shape decomposition, the shape temporal plan, and the organizational structures that could potentially result in optimized assembly. Online processing is favorable due to dynamic, unpredictable sequence of events. When the shape construction process starts to fail (which can happen due to failures in communication or instruction execution, particle mobility constraints, subshapes being stuck and not able to fuse, etc.), the old plan can no longer produce successful results. Consequently, the plan needs to be changed – and the control process (shape assembly execution) must be adjusted online. However, completely distributed plan redesign with limited computational power may be infeasible.

- *Benefits of designing efficient original plan in improving agility:* Objective shape elements (or mission tasks in the military domain) can be divided into separate components that can serve as an agreed method of distributing work for parallel execution. However, such decomposition must match the available resources (active particles, commanders) and their functionality (computational properties, skills), as well as the current environment situation. Most promising approach is to design the plan with contingencies, which may detail the alternative steps through which the shape can be constructed.
- *Benefits of distributing command and control in improving agility:* Distributing C2 roles provides execution more resilient to control failures because no single point of failure exists, faster responsiveness to events, as well as faster construction via parallel channels of command and control. However, any parallalization introduces the complexity of coordination (fusion of the subshapes in our case), which might not be possible to conduct and can result in stalling the progress of the assembly. In the military domain, over-relying on self-synchronization might burden the command elements at the key junctions, requiring commanders to align their situation interpretations, developing multi-dimensional interdependencies between the operations of different command elements, and creating potentials for competition.
- *Benefits of dynamically reallocating the roles of particles in improving agility:* Static allocations of active particles to subshape construction, allocation of reactive particles to be controlled by active particles, and allocation of reactive particles to shape plan node roles may become inefficient over time. This is due to uncertainty in the future location in the space of the moving particles, availability of particles, and uncertainty of the success of communication. For example, the movements of particles completely changed the ability of the active particles to communicate with reactive particles, with allocated reactive particles being too far to the particles they needed to join with at the time of instruction execution.
- *Benefits of leveraging randomness:* While randomness can prevent some of the benefits of offline planning, by incorporating shake we were able to create more reliable object construction. Thus, we used the random movements to our advantage similarly to how the random walk algorithms allow the solution to avoid local optimums.

# Conclusions

In this paper, we have described the application of hybrid command and control design for the domain of automated particle shape assembly. Findings of this work will benefit the development of novel agile control and planning strategies for the semi-automated forces, as well as provide the enabling functionality for the distributed agile control of future programmable ensembles. We have identified some of the challenges associated with agile organizational structures and processes, especially in situations with high complexity of the objective and uncertainty in the environment.

During our work, we have discovered several additional potentials to improve our hybrid command and control architecture. First, in the research presented in this paper we have worked with homogeneous particles, and will naturally extend our architecture to particles and units with diverse properties. Second, adapting command and control organization and construction plan over time will improve the efficiency of assembly execution. Third, such adaptive capabilities with addition of more intelligence in the particles will enable the assembly of the shapes that can adapt their form to the situation at hand without the need for reassembly – for example, designing the wrench which can adapt to the task (size of the bolt). This requires modeling the observation of the particles more comprehensively (active particles will provide observations about surrounding of the shape to learn the size of the bolt that the wrench is about to be applied to), as well as modeling the resizing of the shape structure potentially through mechanical or structural means. We are also working on the feedback that active particles may provide about the observed structure of the currently assembled object shape. This is especially useful for material property assessment and diagnostics of the structural topology of unknown substances. In the military applications, this is equivalent to the adaptation of the situational awareness of the commanders given the information collected by the units of the ground or in the air. Finally, we are enhancing our current C2 theories and models for the case of competitive assemblies, where different competing objectives or sides exist that try to achieve their assembly goals, and each active particle may take away the control over the reactive particles. The applications of the competitive assembly can range from the medications that build healthy strands of bacteria to control of robotic forces fighting against adaptive opponents. In military command and control applications, this will represent the existence (of possibly multiple) opposing (and/or non-hostile) sides. When fully developed, this framework will provide the baseline research for designing fully adaptive semi-automated forces capable to respond to varying situations and contingencies.

## Acknowledgement

The authors gratefully acknowledge the guidance and wisdom of Dr. Mitch Zakin of the Defense Sciences Office of the Defense Advanced Research Projects Agency, who funded this research. The opinions expressed here, however, are those of the authors and do not necessarily reflect the official position of the sponsors or the Department of Defense.

## References:

Alberts, D.S., and Hayes, R.E. *Power to the Edge. Command, Control in the Information Age*. CCRP Publications Series, June 2003.

Alstyne, M.(1997), “*The State of Network Organization: A Survey in Three Frameworks*”, Journal of Organizational Computing and Electronic Commerce, 7(2/3), 83-151.

Entin, E., “Optimized Command and Control Architectures for Improved Process and Performance,” Proc. *1999 Command & Control Research & Technology Symposium*, NWC, Newport, RI, June 1999.

Entin, E., F. Diedrich, D. Kleinman, B. Kemple, S. Hocevar, B. Rubineau, and D. Serfaty, “When Do Organizations Need to Change – Part II: Incongruence in Action,” *International Command and Control Research and Technology Symposium*, Washington, DC, June, 2003.

Jacobs, H. O., Tao, A. R., Schwartz, A., Gracias, D. H., & Whitesides, G. M. (2002). Method of Self-Assembly and Self-Assembled Structures. Wordwide Patents Pending.

Jacobs, H. O., & Wei, Z. (2006). Programmable Self-assembly: Heterogeneous Integration of Chip-scale Components on Planar and Non-planar Surface. USA, Provisional Patent Application.

Kleinman, D.L., G.M. Levchuk, S.G. Hutchins, and W.G. Kemple, “Scenario Design for the Empirical Testing of Organizational Congruence,” *Proceedings of the 2003 International Command and Control Research and Technology Symposium*, Washington, DC, June, 2003.

Levchuk, G.M., Y.N. Levchuk, J. Luo, K.R. Pattipati, and D.L. Kleinman, “Normative Design of Organizations - Part I: Mission Planning”, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 32, No. 3, May 2002, pp. 346-359.

Levchuk, G.M., Y.N. Levchuk, J. Luo, K.R. Pattipati, and D.L. Kleinman, “Normative Design of Organizations - Part II: Organizational Structure”, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 32, No. 3, May 2002, pp. 360-375.

Levchuk, G. M., Meirina, C., Pattipati, K.R., & Kleinman, D.L. “Normative design of project-based organizations: Part III - Modeling congruent, robust and adaptive organizations,” *IEEE Trans. on SMC: Part A: Systems and Humans*, 34(3), 337-350, 2004

Levchuk, G., F. Yu , K. Pattipati, and Y. Levchuk “From hierarchies to heterarchies: Application of network optimization to design of organizational structures,” Proceedings of CCRTS, Washington, DC, June, 2003.

Levchuk, G., F. Yu , Y. Levchuk, and K. Pattipati, “Networks of Decision-Making and Communicating Agents: A New Methodology for Design and Evaluation of Organizational Strategies and Heterarchical Structures”, *Proceedings of the 2004 International Command and Control Research and Technology Symposium*, San Diego, CA, June, 2004

Levchuk, G., Levchuk, Y., Weil, S. A., & Pattipati, K. “PERSUADE: Modeling framework for the design of modular Army organizations,” *Proceedings of the 2006 Command and Control Research and Technology Symposium*, San Diego, CA, 2006

Yu, F., F. Tu, and K. Pattipati, “Integration of a Holonic Organizational Control Architecture and Multiobjective Evolutionary Algorithm for Flexible Distributed Scheduling,” *IEEE Transactions on Systems, Man and Cybernetics*, *Part A: Systems and Humans*, 38(5), 1001-1017, 2008

Zheng, W. & Jacobs, H. O. (2006). Self-assembly process to integrate and connect semiconductor dies on surfaces with single-angular orientation and contact-pad registration. Advanced Materials (Weinheim, Germany), 18, 1387-1392.