ISSN: 2321-2152
IJMECE
International Journal of modern electronics and communication engineering
E-Mail
editor.ijmece@gmail.com
editor@ijmece.com
www.ijmece.com

IJMECE International Journal of modern electronics and communication engineering

ISSN 2321-2152

www.ijmece.com

Vol 11, Issue 2, 2023

https://zenodo.org/records/14505395

An Introduction to Distributed Intelligence and Its Use in Multi-Robot Systems

Dr.Valiki Vijaya Bhasker[1], Professor[1], Department of ECE, Siddhartha Institute of Technology & Sciences, Telangana, India

Rambabu Bommisetti[2], Assistant Professor[2], Department of CSE, Siddhartha Institute of Technology & Sciences, Telangana, India.

Abstract

This article provides an overview of the principles behind distributed intelligence and the justifications for further research in this area. Since the nature of the interaction is relevant to the solution paradigm, we proceed to categorise prevalent DIS systems in light of the interactions they often display. We describe three prevalent paradigms for distributed intelligence and provide examples of their use in multi-robot systems. These paradigms include bio-inspired, organisational and social, and knowledge-based, ontological. We next examine the issue of work allocation, which arises often in multi-robot systems, and demonstrate how the strategy to solving it varies greatly depending on the paradigm used to abstract the problem. We conclude that the paradigms are not equivalent, and that choosing the right one depends on the particular restrictions and needs of the application at hand. More research is required to help system designers choose the best abstraction (or paradigm) for a specific challenge.

Introduction

intelligence that is spread out throughout a network Systems of creatures working together to reason, plan, solve problems, think abstractly, grasp concepts and language, and learn are referred to as having "distributed intelligence." In this context, we use the term "entity" to refer to any self-aware thing, whether it a person, a robot, a piece of software, a piece of hardware, or anything else with a brain. In such setups, several actors often focus on certain facets of the overall operation. Humans have evolved to work in groups, so we're all used to sharing knowledge across a variety of minds. Chief Executive Officer, Chief Operating Officer, Chief Financial Officer, Chief Information Officer, and so on are all examples of titles held by members of corporate management teams. Specialists in medical oncology, surgical oncology, plastic and reconstructive surgery, pathology, and other related fields make up oncology patient care teams. The military also makes use of distributed intelligence, as seen in special forces A-Teams.

hone your skills in the areas of warfare, technology, medicine, and communications. Personnel aboard a military aircraft carrier, for instance, may be divided into subunits such as catapult crew, landing signal officers, ordnance men, plane handlers, etc. Humans have obviously realised that these teams, by using experts who work together effectively, can perform complicated jobs extremely rapidly. The goal of distributed intelligence in computer science and related subjects is to design systems that include software agents, robots, sensors, computers, and even humans and animals (like search and rescue dogs) that can collaborate as effectively as human teams. There is little doubt that such systems have the potential to solve several significant problems, such as urban search and rescue, military network-centric operations, gaming technologies and simulation, computer security, transportation and logistics, and many more.

Distributed Intelligence and Its Domain

Researchers are finding a wide variety of paradigms that might be used to successfully implement distributed intelligence. Some forms of distributed intelligence are not suited to the aforementioned paradigms. Therefore, it is crucial to learn about the diverse forms of dispersed intelligence that might emerge in distinct contexts. The different possible interactions between entities in a distributed intelligence system may be used to get a better knowledge of the domain space. We find it useful to consider interactions along three axes, as shown in Figure 2: the nature of the objectives involved, whether or not the entities involved are aware of one another, and whether or not the entity's activities contribute to the success of the team as a whole. Systems are categorised based on whether or not their constituent parts pursue separate or common objectives. The systems are classified into two groups, aware and unaware, along the dimension of awareness of others. In this sense, "conscious" refers to a capacity for an entity to reflect on the behaviour and motivations of its teammates. Although non-aware robots may detect the presence of nearby objects and adjust their position accordingly, for example, they are unable to reason about their colleagues' intentions or anticipate their next moves. The notion of stigmergy, in which things communicate with one another without exchanging direct messages, underlies the operation of many "un aware" systems. Finally, we classify systems into those in which a person's activities contribute to the success of the group as a whole (yes) and those in which they do not (no) (no). A floor-cleaning robot, as part of a team of floor-cleaning robots, is an example of an entity whose activities further the aims of others. The floor cleaning efforts of one robot assist the other robots in the team avoid having to clean the same area twice. It is clear that these divisions of the domain space are approximations, yet we nevertheless find them useful for learning about the most common interactions in real-world scenarios. This subspace is a representation of the many interactions that may be found in distributed intelligence systems. The following are typical methods of communication:

- Collective
- Cooperative
- Collaborative
- Coordinative

In the following paragraphs we describe these types of interactions in more detail. Perhaps the simplest type of interaction is the collective interaction, in which entities are not aware of other entities on the team, yet they do share goals, and their actions are beneficial to their teammates. An example of this type of intraction in multi-robot systems is the swarm robotics work of many researchers (e.g., (McClurkin 2004; Matara’s 1995;

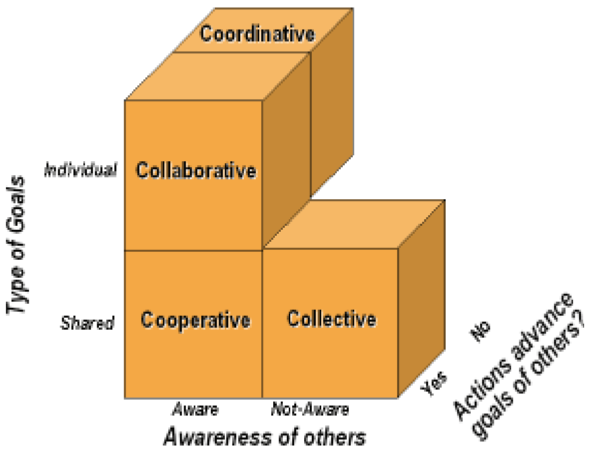

Figure 1: Categorization of types of interactions in systems of distributed intelligence.

Kube & Zhang 1993)). This work focuses on creating systems of robots that can perform biologically-relevant activities such as searching for food, travelling in large groups, herding livestock, maintaining a formation, and so forth. When combined with a greater number of robots, the global aim is generally realised as an emergent aspect of the local interactions, with the robots in these systems often performing relatively basic local control rules. The second kind of interaction is cooperative interaction, which occurs when the involved entities are aware of one another, have similar objectives, and take steps that benefit their teammates. For example, in multi-robot systems, robots could cooperate to move a box (e.g., (Gerkey & Matari'c 2002)), clean up a jobsite (e.g., (Parker 1998)), conduct a search and rescue operation (e.g., (Murphy 2000)), or even explore distant planets (e.g., (Stroup et al., 2006)). In such setups, robots may need to coordinate their use of the shared workspace so that they don't impede one another's progress toward the system's overarching aim. However, the robots' efforts are mostly concentrated on collaborating to accomplish a shared objective.

Robots with their own objectives, who are aware of their colleagues, and whose activities contribute to the success of the team's overall objectives provide a third form of interaction in distributed intelligence systems. Collaborative refers to the subset of the domain space in which entities cooperate to attain their separate but compatible objectives. In this context, we distinguish between the cooperative do main space and the ability of entities to work together to assist others in better achieving their own objectives. We are used to seeing cooperation in human research teams, where each individual has their own area of specialty that contributes to the group's success. While everyone on the team is working toward the same common goal—completing their assigned portion of the research—their efforts will be amplified by the synergistic effect of working with others who bring unique perspectives to the table. Most of these partnerships are also cooperative, and any group may become cooperative by shifting its focus to the bigger picture and revaluating its aims. Collaborative teamwork may be shown by a collection of robots working together to achieve individual goals.

In the event that a robot's sensors prevent it from reaching its destination, it may be able to collaborate with other robots to achieve its objectives by pooling its resources and enhancing the sensory capabilities of each member. Alliances like this have been shown in (Parker & Tang 2006; Vig & Adams 2006). When it comes to distributed intelligence, coordinative interaction is the fourth and last form of interaction. Entities in such systems are aware of one another, but they are not working toward a shared objective, and their activities are not conducive to the success of the team as a whole. These conflicts often arise when many robots are working in the same area. Coordination among the robots is essential if they are to cause the least possible disruption to one another. In these contexts, it is not uncommon to use multi-robot route planning (e.g., (Kloser& Hutchinson 2006; Guo & Parker 2002)) or traffic control (e.g., (Asama et al. 1991; Yuta & Pre mute 1992; Wang 1991)) approaches. Besides, we might have added a third dimension to our domain space to classify systems according to whether they (1) help other entities achieve their objectives, (2) don't influence other entities' ability to achieve their goals, or (3) hurt other entities' ability to achieve their goals. This would allow us to design a novel kind of interaction in which the participants all operate in accordance with their own self-interest, are aware of one another, and yet impede progress toward the objectives of the other participants. This is the essence of the antagonistic sphere, where entities conspire against one another. Many researchers have devoted time and energy to this question in the context of multi robot systems, specifically in the context of multi robot soccer (see, for example, (Kitano et al. 1997; Browning et al. 2005; Veloso, Stone, & Han 1999; Stone & Veloso 1999)). There is no denying the military utility of this kind of cooperation.

Models for Decentralized Intelligence

There are as many different models for creating distributed intelligence as there are different forms of interactions in systems based on distributed intelligence. Each paradigm provides a distinct level of abstraction over the issue space, allowing the system designer to gain insight into effective approaches to solving the challenge. Whether it's the structure of ant colony or human community, these models often draw parallels. Paradigms may be useful tools, but they aren't universally applicable across all interaction dynamics. This section provides an overview of many prevalent distributed intelligence models, with a special emphasis on how they apply to systems with several robots. It is important to keep in mind that a key difficulty shared by all of these paradigms is figuring out how to bring about global coherence via the local interaction of things. Different levels of issue abstraction reveal complementary approaches to resolving this difficulty.

Three commonly used paradigms for building systems of distributed intelligence include:

• Bioinspired, emergent swarms' paradigm,

• Organizational and social paradigms, and

• Knowledge-based, ontological, and semantic paradigms.

We discussed concepts of the bioinspired, emergent swarms' paradigm in the previous section, as part of the description of collective interactions. In this paradigm, the need for communication between entities is greatly reduced by assuming the ability of the entities to sense relevant information in their local environments (i.e., staggery). The application requirements in these problems allow for simple action protocols, or control rules, that are identical on each entity, and that lead to the desired group behaviour. An example local control rule under this paradigm that can cause all the agents/robots to aggregate (as in a swarm) is

```
Aggregate:
  If agent is outside aggregation
      distance
  then turn toward aggregation
      centroid and go.
  Else
     stop.
```

This is an effective paradigm for applications that need the same work to be done in a decentralised environment, where the job doesn't need sophisticated entity-entity interactions and all entities are generic. Both the inverse issue, where we want to derive the local control rules given a desired global behaviour, and the former problem, where we want to anticipate the global behaviour given a set of local control rules, provide formidable research problems. Flocking, schooling, foraging, chaining, searching, sorting, herding, aggregation, condensation, dispersion, confinement, formations, harvesting, deployment, and coverage are just some of the geographically dispersed applications that might benefit from this paradigm. However, more complicated frameworks for solving various kinds of interactions are needed.

Task Assignment in a Multi-Robot Environment: Competing Models

After looking at three different approaches to distributed intelligence systems, we'll quickly contrast how each deal with a typical problem in multi-robot setups: dividing up the work. As was discussed before, job allocation is a common problem in multi-robot applications when the team's goal is broken down into individual tasks. Various robots can tackle different tasks, and vice versa. While it is possible to work on independent activities at the same time, dependent tasks must be completed in a sequential order that accounts for their interdependencies. Once the list of jobs is defined, the next step is to find the optimal way to assign robots to jobs so as to maximise some objective function. This is the issue of dividing up work. As previously shown by Gerkey and Matara's (2004), optimum solutions to the broad work allocation issue are NP-hard. As a result, approximations that are acceptable in practise are often used as solutions to this issue. Consider the multi-robot work allocation problem, and how each of the above paradigms might approach it. To begin, a large number of identical robots would normally be assumed by the bioinspired method of work allocation. Any robot that is nearby and aware of the need of completing a job might volunteer to do so (i.e., the task is allocated to that robot). Robots may utilise staggery to figure out what to do without resorting to direct communication. If a robot fails, it may be swapped out for another one. All robots should follow this idea for best results. Second, much as we discussed before for multi-robot soccer, roles might be used to organise the distribution of tasks. Robots choose positions that are most suited to their capabilities, and each duty includes a number of distinct responsibilities. In this context, robots may have a wide range of sensing, computing, and effector skills; they need not be standardised.

The market-based approach to allocation was also proposed as a different organisational strategy. With these methods, robots negotiate for jobs by openly discussing their capabilities and offering bids based on their predicted contributions. Typically, assignments are established by giving each job to the most efficient robot possible. The Contract Net Protocol (Smith, 1980) is foundational here because it was the first to tackle the issue of how agents might negotiate to collectively accomplish a set of tasks. The M+ architecture was the first to use a market-based method for the purpose of locating tasks for many robots (Botelho & Alami 1999). In the M+ method, each robot makes its own strategy to complete its objective. Next, they employ social norms that allow for the gradual merging of plans as they negotiate with other team members to gradually adjust their activities to best serve the team as a whole. Last but not least, the knowledge-based method is used for work distribution in multi-robot teams by modelling colleague skills. Among the many potential variants is the ALLIANCE technique (Parker, 1998), in which robots simulate the capacity of team members to carry out the duties of the system by watching team member performance and collecting important task quality information, such as the time to task completion. These models are then used by the robots to decide which jobs would be best for the team as a whole. The selection of assigned tasks in this method does not need open dialogue. The use of trained models of teammate skills opens the door to other methods. These job allocation examples show that there are numerous possible solutions to a given issue in multi-robot systems, depending on the abstraction paradigm used. Benefits and drawbacks of each paradigm vary depending on the context. The appropriate paradigm depends on the specific limitations and needs of the application at hand.

Conclusions

In this article, we've introduced several key concepts in distributed intelligence and discussed the many possible interactions between distributed systems as well as some of the most popular approaches to achieving distributed intelligence. We have utilised examples from the area of multi-robot systems to show, compare, and contrast the various interactions and paradigms in order to better understand the difficulties. The takeaway from these debates is that the appropriate paradigm for a given problem depends on the specifics of the application at hand. We also point out that different robot paradigms may be used concurrently in complex systems. An organisational paradigm can be used to define roles for the high-level abstraction, a knowledge-based approach can be taken to multi-robot mapping, a knowledge-based modelling approach can be taken to mobile network deployment, and a bio-inspired approach can be taken when creating a mobile sensor network (Howard, Parker, &Sukhumi, 2006). The task of system designers is to develop and use paradigms that are tailored to the unique requirements of each application.

References

[1] Asama, H.; Ozaki, K.; Itakura, H.; Matsumoto, A.; Ishida, Y.; and Endo, I. 1991. Collision avoidance among multiple mobile robots based on rules and communication. In Proceedings of IEEE/RJS International Conference on Intelligent Robots and Systems. IEEE.

[2] Botelho, S., and Alami, R. 1999. M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In Proceedings of the IEEE International Conference on Robotics and Automation, 1234– 1239. IEEE.

[3] Browning, B.; Bruce, J.; Bowling, M.; and Veloso, M. 2005. STP: Skills, tactics and plays for multi-robot control in adversarial environments. IEEE Journal of Control and Systems Engineering 219:33–52.

[4] Dias, B.; Zlot, R.; Kalra, N.; and Stentz, A. 2006. Marketbased multirobot coordination: A survey and analysis. Proceedings of the IEEE 94(7):1257–1270.

[5] Finin, T.; Labrou, Y.; and Mayfield, J. 1995. Kqml as an agent communication language. In Bradshaw, J., ed., in Software Agents. MIT Press.

[6] Genesereth, M. R., and Fikes, R. E. 1992. Knowledge Interchange Format, Reference Manual. Computer Science Departmenet, Univ. of Stanford.

[7] Gerkey, B. P., and Matari´c, M. J. 2002. Sold! auction methods for multi-robot coordination. IEEE Transactions on Robotics and Automation 18(5):758–768.

[8] Gerkey, B., and Matari´c, M. J. 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. International Journal of Robotics Research 23(9):939–954.

[9] Kloder, S., and Hutchinson, S. 2006. Path planning for permutation-invariant multirobot formations. IEEE Transactions on Robotics 22(4):650–665.

[10] Kube, C. R., and Zhang, H. 1993. Collective robotics: From social insects to robots. Adaptive Behavior 2(2):189–219.

[11] Marsella, S.; Adibi, J.; Al-Onaizan, Y.; Kaminka, G.; Muslea, I.; and Tambe, M. 1999. On being a teammate: Experiences acquired in the design of RoboCup teams. In Etzioni, O.; Muller, J.; and Bradshaw, J., eds., Proceedings of the Third Annual Conference on Autonomous Agents, 221–227.

[12] Matari´c, M. J. 1995. Issues and Approaches in the Design of Collective Autonomous Agents. Robotics and Autonomous Systems 16:321–331.

[13] McLurkin, J. 2004. Stupid robot tricks: Behavior-based distributed algorithm library for programming swarms of robots. In M.S. Thesis, Massachusetts Institute of Technology. Murphy, R. R. 2000. Marsupial robots for urban search and rescue. IEEE Intelligent Systems 15(2):14–19.

[14] Parker, L. E., and Tang, F. 2006. Building multi-robot coalitions through automated task solution synthesis. Proceedings of the IEEE, special issue on Multi-Robot Systems 94(7):1289–1305.

[15] Parker, L. E. 1998. ALLIANCE: An architecture for faulttolerant multi-robot cooperation. IEEE Transactions on Robotics and Automation 14(2):220–240.

[16] Smith, R. G. 1980. The Contract Net Protocol: high-level communication and control in a distributed problem solver. IEEE Transactions on Computers C-29(12).

[17] Stone, P., and Veloso, M. 1998. A layered approach to learning client behaviors in the robocup soccer server. Applied Artificial Intelligence 12:165–188.

[18] Stone, P., and Veloso, M. 1999. Task decomposition, dynamic role assignemnt, and low-bandwidth communicaiton for real-time strategic teamwork. Artificial Intelligence 110(2):241–273.

[19] Stroupe, A.; Okon, A.; Robinson, M.; Huntsberger, T.; Aghazarian, H.; and Baumgartner, E. 2006. Sustainable cooperative robotic technologies for human and robotic outpost infrastructure construction and maintenance. Autonomous Robots 20(2):113–123.

[20] Veloso, M.; Stone, P.; and Han, K. 1999. The cmunited97 robotic soccer team: Perception and multiagent control. Robotics and Autonomous Systems 29(2-3):133–143.

[21] Vig, L., and Adams, J. A. 2006. Multi-robot coalition formation. IEEE Transactions on Robotics 22(4):637–649.

[22] Wang, J. 1991. Fully distributed traffic control strategies for many-agv systems. In Proceedings of the IEEE International Workshop on Intelligent Robots and Systems, 1199– 1204. IEEE.

[23] Yuta, S., and Premvuti, S. 1992. Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 1566–1574. IEEE.

[24] Zlot, R., and Stentz, A. 2006. Market-based multirobot coordination for complex tasks. International Journal of Robotics Research 25(1):73–101.