



# JSwarm: A Jingulu-Inspired Human-AI-Teaming Language for Context-Aware Swarm Guidance

Hussein A. Abbass<sup>1\*</sup>, Eleni Petraki<sup>2</sup> and Robert Hunjet<sup>3</sup>

<sup>1</sup>School of Engineering and IT, University of New South Wales, Canberra, NSW, Australia, <sup>2</sup>Faculty of Education, University of Canberra, Canberra, ACT, Australia, <sup>3</sup>Defence Science and Technology Group, Canberra, NSW, Australia

Bi-directional communication between humans and swarm systems begs for efficient languages to communicate information between the humans and the Artificial Intelligence (AI)-enabled agents in a manner that is most appropriate for the context. We discuss the criteria for effective teaming and functional bi-directional communication between humans and AI, and the design choices required to create effective languages. We then present a human-AI-teaming communication language inspired by the Australian Aboriginal language of Jingulu, which we call JSwarm. We present the motivation and structure of the language. An example is used to demonstrate how the language operates for a shepherding swarm guidance task.

## OPEN ACCESS

**Keywords:** human-AI teaming, human-swarm teaming, teaming languages, jingulu, human-swarm languages

### Edited by:

Jaelle Frere Lawless,  
Paine College, United States

### Reviewed by:

Jaelle Scheuerman,  
United States Naval Research  
Laboratory, United States  
Michael Mylrea,  
University of Miami, United States

### \*Correspondence:

Hussein A. Abbass  
hussein.abbass@gmail.com

### Specialty section:

This article was submitted to  
Interdisciplinary Physics,  
a section of the journal  
Frontiers in Physics

**Received:** 14 May 2022

**Accepted:** 16 June 2022

**Published:** 14 July 2022

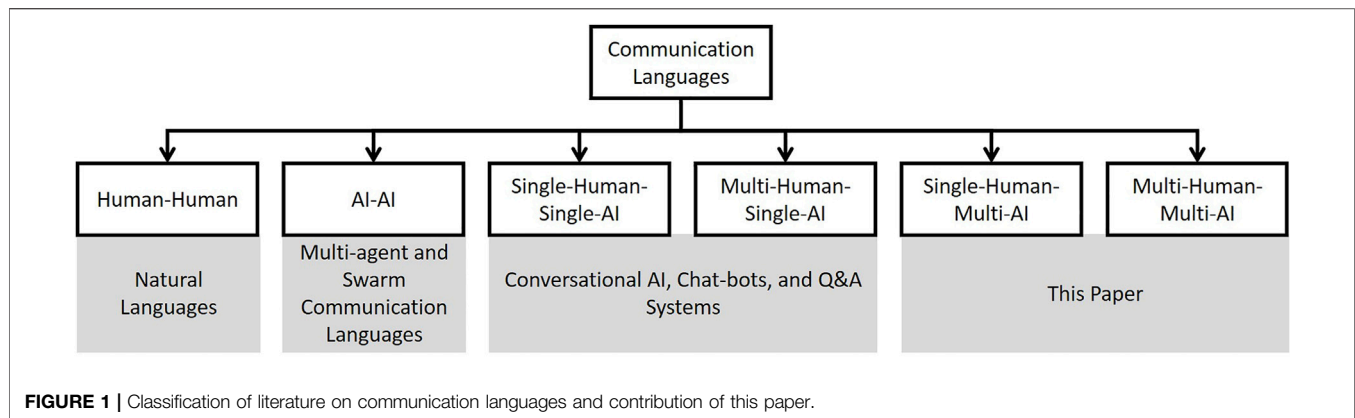
### Citation:

Abbass HA, Petraki E and Hunjet R  
(2022) JSwarm: A Jingulu-Inspired  
Human-AI-Teaming Language for  
Context-Aware Swarm Guidance.  
*Front. Phys.* 10:944064.  
doi: 10.3389/fphy.2022.944064

## 1 INTRODUCTION

Natural languages are very rich and complex. Human languages have been in place for around 10,000 years and have served humans effectively and efficiently. Even within a single language, there could be many different varieties or codes, used by specific speaker groups, or maintained for particular contexts. English for specific purposes is a discipline of language teaching that focuses on the teaching of English for various professional or occupational contexts, such as English for nursing or English for legal purposes. These natural languages could be too inefficient for an artificial intelligence (AI) enabled agent designed for a particular task or use, due to the languages being highly-complex, thus, creating a space of ambiguity or unnecessary complexity. There is a significant amount of research in computational linguistics that could help and guide the design of human-friendly languages for distributed artificial intelligence (AI) systems to enable humans and AI-enabled agents to work together in a teaming arrangement. Each relationship-type among a group of agents shape the subset of the language required to allow agents to negotiate meanings and concepts associated with the particular domain where the relationship-type belongs. Moreover, understanding the principles for computational efficiency in natural languages has been the subject of inquiry by computational linguists. By identifying the minimum set of rules (ie grammar) governing a language, linguists discover the DNA-equivalent of, and morphogenesis for, human languages. Be it through learning or direct encoding of this minimum set, the concept of computational efficiency can contribute to an assurance process for a proper coverage of the semantic space required for, and requirements to reduce impermissible sentences during, an interaction.

Unsurprisingly, culture shapes language and vice-versa [1, 2]. This has led to the diversity of human languages available today, which vary in grammar, vocabulary and complexity of meaning. Similar to human systems, in artificial systems language also reflects cultural and social networks. To situate the contribution of this paper in the current literature, **Figure 1** depicts a high-level



classification of the research done on human-AI languages. The figure presents six dimensions or lenses that one can see the literature. These dimensions are formed from the perspective of who is interacting with whom. We will discuss below three research directions with particular relevance to the current work and focus particularly on human-designed languages for different forms of human-machine interaction. The discussion together with the figure attempts to compress the wide variety of contributions made in this space for centuries.

Human-Human languages have a very long history, with studies that could be traced back to the ancient Greeks. AI-AI languages have its roots in recent literature, and have been a fruitful research area, whereby the languages could be emerging or designed. The literature on the languages required at the interface between humans and machines, or more specifically in this paper human-AI systems, has witnessed different categories of methods to approach the topic. The current literature can generically be categorised into three research directions, each with their own cultural traits. The first research direction aims at designing “computer programming” languages (see for example [3]), aiming at affording a human with a language to program a group of robots. This class of languages allow a human to encode domain knowledge in algorithmic form for robots to function and can be seen as human’s means to communicate to the machine. The second research direction focuses on languages required for communication between an AI and another, or between an AI and a human. In this branch, work on designing languages to allow communication among a group of artificial agents has been primarily dominated by the multi-agent literature [4] and more recently the swarm systems literature [5]. When a human interacts with an AI, conversational AI [6, 7], chatbots and Questions and Answer (Q&A) systems [8] dominate the recent literature using data-driven approaches and neural-learning [9]. The third research direction shifts focus away from human-design of the communication language to the emergence of communication and language in a group of agents. A reasonably large body of evolutionary and developmental robotics literature [10] has dedicated significant efforts into this research direction. These three research directions could carry some relevance across all dimensions in **Figure 1**, but clearly the amount of relevance is not uniform.

The focal point of this paper is human-AI teaming, especially within the context of distributed AI systems capable of synchronising actions to generate an outcome, or what we call AI-enabled swarm systems. In particular, our aim is to design a computationally efficient human-friendly language for human-AI teaming that is also appropriate for human-swarm interaction and swarm-guidance. The design is inspired by the Jingulu language [11], an Aboriginal language spoken in parts of Australia and demonstrated on a swarm guidance approach known as shepherding [12]. Briefly, the shepherding (teaming) system comprises of a swarm (analogous to sheep) to be guided, an actuator agent (analogous to a sheepdog in biological herd mustering) with the capacity to influence the swarm, an AI-shepherd (analogous to sheepdog cognition) with the capacity to autonomously guide the actuator agent (sheepdog body) to achieve a mission, and a human-team (analogous to farmers) with the intent to move the swarm. To achieve this goal, the human team interacts with the AI-shepherd and is required to monitor, understand, and command it when necessary, as well as take corrective actions when the AI-shepherd deviates from the human team’s intent. We assume that the AI-shepherd is more clever than a sheepdog and is performing the role of the human-shepherd in a biological mustering setting. Biological swarms such as sheep herds have been shown to be appropriately modelled by attraction and repulsion rules amongst the swarm members. The special characteristic of the Jingulu language is that the language has only three main verbs: do, go and come. Such a structure is most efficient for communication in attraction-repulsion equations-based distributed AI-enabled swarm systems as this paper shows.

An introduction to Jingulu and swarm shepherding is presented in **Section 2**. We then discuss the requirements for computational efficiency when designing human-AI teaming languages and propose an architecture in **Section 4**. A computationally-efficient human-friendly language for human-AI swarm teams is then presented in **Section 5**, followed by a discussion on the assurance of human-AI teaming language in **Section 6**. Conclusions are drawn and future work is discussed in **Section 7**.

## 2 BACKGROUND

### 2.1 Aboriginal Languages

Aboriginal people are the traditional owners of the Australian land. Different tribes occupy different parts of the island. Australian Aboriginal languages display unique syntactic properties, and one property in particular is called the non-configurationality free word order [13]. We offer a simple introduction to some basic linguistic features to explain this property.

Syntactic relations describe the minimal components of a simple sentence that usually consists of subject-verb-object, which we will abbreviate as SVO. The presence of SVO is a universal property in the organisation of sentences despite the existence of variation in the order. Some languages follow SVO, others VSO, and others SOV. SVO are syntactic positions that are occupied by noun phrases (NPs), verb phrases (VPs), and NPs, respectively. English is an SVO language.

[My daughter] [ate] [her ice cream]  
 [Subject] [Verb] [Object]  
 [Pronoun Noun] [Verb] [Pronoun noun]

Non-configurationality describes a principle that applies to languages whose sentence structure imposes fewer restrictions in the order of syntactic relations. Greek is a non-configurational language that allows SVO swapping in a sentence.

[E kore mou] [efage] [To pagoto ths]  
*[The daughter my] [ate] [The ice cream her]*  
 [Article, noun, pronoun] [Verb (tense/person)] [Article noun pronoun]  
 [Subject] [Verb] [Object]  
 [efage] [E kore mou] [To pagoto ths]  
 [V] [S] [O]  
 [To pagoto ths] [E kore mou] [efage]  
 [O] [S] [V]

Most configurational and non-configurational languages impose restrictions on the constituent order, that is the order of words that forms the subject or the verb phrase or the object phrase. For example, in English, the order of the subject [my daughter] needs to follow [pronoun + noun] order and not vice versa. This group of words always moves together as a phrase (constituent) and cannot be separated.

However, many Aboriginal languages are not only non-configurational but also display free word order within the constituent phrases. This means that a noun phrase, that is a group of words that might fill the position of the subject, for example, or the object, can be split in the sentence. Such languages express meaning, using inflectional morphology such as prefixes and suffixes that might indicate, person, gender, tense, aspect, which are limited in the English language.

This flexibility has also been found in Jiwari and Walpiri, two heavily studied Aboriginal languages. Jiwari language (no longer spoken), used to be part of the Pilbara region in Western

Australia. Walpiri, is an Aboriginal language spoken in the Northern Territory:

Example from Walpiri [14]:  
 [Kurdu-jarra-rlu] [ka-pala] [maliki wajili-pi-nyi] [wita-jarra-rl]  
 [child-(two)-] [(Present)] [Dog chase] [Small]

*Two small children are chasing the dog. OR Two children are chasing the dog and they are small.*

### 2.2 The Jingulu Language

The Jingili people live in the western Barkly Tablelands of the Northern Territory in the town of Elliott. We consulted the grammar of Jingulu (the language of the Jingili's people) in Pensalfini's dissertation and subsequent book written on the grammar of the Jingulu language [11].

Similar to many Aboriginal languages, Jingulu displays free constituent order.

1. *Uliyija-nga nglnja-ju karalu.* (SVO)  
 sun-ERG.f burn-do ground  
 The Sun is burning the ground.
2. *Uliyijanga karalu ngunjaju.* (SOV)
3. *Ngu njaju uliyijnnga karalu.* (VSO)

Jingulu shows also free word order within the Noun Phrases as seen below.

The Noun phrase in English 'that stick' would never be separated, however in Jingulu, they seem to can be separated.

[Ngunll] [maja-mi] [ngnrru] [darrangku.]  
 [that] [get] [(Me)] [stick]

Get me that stick.

Jingulu has many interesting features that we will not cover in this paper. Instead, we will focus on the most prominent feature of Jingulu, that it is a language with only three primary verbs: do, go, and come. We will refer to them as light verbs. We are not aware of any other Aboriginal languages that display this structure and as such this is perhaps unique for Jingulu.

We argue that this feature makes Jingulu an ideal natural language for representing spatial movements between entities and the exchange of communication messages, including commands, among the agents. To explain this further, we borrow three examples from [15]. The use of FOC in the following text indicates 'contrastive focus', which is a linguistic marker to represent where focus is placed in a sentence. This is not a feature of Jingulu per se, it is part of the linguistic characterisation linguists use to mark attention in sentences.

Example 1 [15][p.228]

Kirlikirlika darra-ardi jimi-rna urruju-ni.  
 galah eat-go that(n)-FOC galah\_grass-FOC  
 "Galahs eat this grass."

Example 2 [15][p.229]

Aja(-rni) ngaba-nya-jiyimi nginirniki(-rni)?  
 what(-FOC) have-2sg-come this(n)(-FOC)  
 “What’s this you’re bringing.”

Example 3 [15][p.229]

Ngaja-mana-ju.  
 see-3MsgIO-do  
 “He is looking at us.”

In the previous examples, the use of do, go and come represents stationarity at current location, and departure away from and arrival to current location, respectively. In the first example, eating the grass indicates that the subject needs to move away from the subject’s current location by “going” to the grass. In the second example, questioning what a person brings depicts a picture of something coming from the subject’s current location to our location. In the third example, “looking at us” does not require any movements, the act could be performed without a change of location. Despite this simplicity, what is truly powerful in this representation is the acknowledgement of the abstract concept of a space, that does not have to be physical in nature. For example, the space could be a space of ideas where an idea might come to a person or a person can go to an idea. The explicit spatial representation is so powerful in the structure of the language.

One distinctive aspect of the Jingulu language is the structure of the verb. Take for example “see-do,” which is the third example above. Prior researchers to Pensalfini explained “do” as an inflectional element representing the final tense-aspect marker. Pensalfini, however, defines it as the “verbal head”; the core syntactic verb. “See” in the sentence is the normalised verb object; a category-less element and root of the verb. The root does not bear any syntactic information, only semantic one. The three semantically bleached light verbs, however, play syntactic and semantic roles. Jingulu has the smallest inventory of inflecting verbs, that is 3, that have complex predicates amongst the northern Australian languages [11].

The verbal structure can, therefore, be described as: *Root (See) + Light-verb (Do)*.

Pensalfini saw the final element as the “true syntactic verb,” which encodes inflectional properties such as tense, mood, and aspect, as well as “distinctly verbal notions such as associated motion. These elements fall into three broad classes, corresponding to the English verbs “come” (3.3), “go” (3.4) and “do/be” (3.5).”

While the language may appear to be complex or primitive, depending on perspectives, from a human-human communication perspective, the above discussion demonstrates very powerful linguistic features in the Jingulu language that we will use for human-AI teaming in a human-swarm context. In particular, the above structure sees the light verb as a semantic carrier; that is, it is the vehicle that carries the meaning created by the root. This vehicle offers spatio-temporal meaning, while the root offers context. These features will be explained later on, in this manuscript.

## 2.3 Shepherding

A swarm is a group of decentralised agents capable of displaying synchronised behaviors despite the simple logic they adopt to make decisions in an environment. Members in a swarm do not necessarily synchronise their behavior intentionally. However, for an observer, the repeated patterns of the coordinated actions they display is synchronised in the behavior space. It is this synchronisation that creates observable patterns in the dynamics that allow members in the swarm to either appear in certain formations or act to generate a larger impact than the impact that could have been generated by any of the individuals in isolation.

The Boids (Bird-oids or Bird-Like-Behavior) model by Reynolds [16] is probably the most common demonstration of swarming in the academic literature. The model relies on three simple rules, whereby each agent is (rule 1) attracted to and (rule 2) aligns its direction with its neighbor, and (rule 3) repulses away from very nearby agents. Following these three simple attraction-repulsion equations, the swarm displays complex collective dynamics.

While the collective boids can swarm, real-world use of swarming calls for methods to guide the swarm [17–22]. There are several ways to guide the swarm from the inside by having an insider influence [23] with a particular intent and knowledge of goals. Another approach is to leave the swarm untouched and to guide them externally with a different agent that is specialized in swarm guidance. This approach mimics the behavior of sheepdogs, where a single sheepdog (the guiding agent) can guide a large number of sheep (the swarm). Indeed, the sheep are modelled with two of the boids rules (attraction to neighbors and repulsion from very nearby agents), with the addition of a third rule to repulse away from sheepdogs. A number of similar models exist to implement this swarm-guidance approach [24–26].

Multiple sheepdogs could be used to herd the sheep, and they could themselves act as a swarm leading to a setup of swarm-on-swarm interaction. The implicit assumption in these models is that the number of sheepdogs is far less than the number of sheep; otherwise the problem could become uninteresting and even trivial. In the simplest single-sheepdog model, the sheepdog switches between two behaviors: collecting a sheep when a sheep is outside the cluster zone of the herd and driving the herd when all sheep are collected within the cluster zone of the herd. When more sheepdogs are used, rules to spread them into formations and/or coordinate their actions are introduced [24, 25, 27–31].

One aim of different shepherding models is to increase the controllability of the sheepdog as demonstrated by the number of sheep it can collect. The two most recent models in the literature are the one by [26] and an improvement on it by [32]. Both models use attraction and repulsion forces and smooth movements by adjusting the velocity vector in the previous timestep with new intent. The latter model improved the former with a number of adjustments. One is to skill the artificial sheepdog to use a circular path to reach a collection or a driving point and to avoid dispersing sheep on the way. Sheep have more realistic behaviors when they do not detect sheepdogs.

In nature, the sheep do not continue to group in the absence of a sheepdog. Instead, they continue to do whatever their natural instinct motivates them to do (eating, sleeping, etc); thus, the latter model does not introduce a bias of continuous attraction to neighbors in the absence of a sheepdog effect. The latter model also defined a sheep’s neighborhood based on different sensing ranges. This is more realistic than the former model which fixes the number of closest sheep; ensuring every sheep always having a fixed number of neighbors regardless of where these neighbors are. Other changes were introduced, which overall improved the success rate of the guidance provided by the sheepdog. In the remainder of this paper, we use El-Fiqi et al. [32]’s model.

A fundamental principle in modelling the shepherding problem is the cognitive asymmetry of agents, where sheep are the simplest agents cognitively with simple survival goals. Sheepdogs have more complex cognition than sheep as they need to be able to autonomously execute a farmer’s intent. The cognition of farmers/shepherds, however, is more complex than sheepdogs due to their role which requires them to have a higher intent with abilities to understand the capabilities of sheepdogs and commanding them to perform certain tasks. In our shepherding-inspired system, we separate between the sheepdog as an actuator and the cognition of a sheepdog. We call the latter the AI, representing the cognitive abilities of a sheepdog to interact and execute human intent.

We present the basic and abstract shepherding model introduced by [32]. While this model has evolved in our own research to more complex versions, it suffices to explain the basic ideas in the remainder of the paper. We will use our generalised notations for shepherding to be consistent with the notations used in our group’s publications.

The set of sheep are denoted as  $\Pi = \{\pi_1, \dots, \pi_i, \dots, \pi_N\}$ , where  $N$  is the total number of sheep, while the sheepdog agents are denoted as  $B = \{\beta_1, \dots, \beta_j, \dots, \beta_M\}$ , where  $M$  is the number of sheepdogs. The agents have a set of behaviours available to choose from; the superset of behaviours is denoted as  $\Sigma = \{\sigma_1, \dots, \sigma_K\}$ , where  $K$  is the number of behaviours available in the system. Agents occupy a bounded squared environment of length  $L$ . Each sheep can sense another sheep in the sensing range of  $R_{\pi\pi}$  and can sense a dog in the sensing range of  $R_{\pi\beta}$ . The global centre of mass (GCM) for sheep is denoted by  $\Gamma_{\pi_i}^t$ . The flock of sheep in this environment has two states; they are either collected or not. Sheep are collected when all sheep are located within a radius  $f(N)$  of their GCM. The radius is calculated as:

$$f(N) = R_{\pi\pi} N^{\frac{2}{3}} \tag{1}$$

If all sheep are within distance of  $f(N)$  of their GCM, then they are collected and are ready to be driven as a herd to the goal. The sheepdog moves to the driving point which is located behind the herd on the ray from the goal to the GCM. If the sheep are not collected, the sheepdog needs to identify the furthest sheep to GCM and move to a collection point to collect that sheep by influencing it to move towards the GCM of the herd. By alternating between these two behaviours, in a simple obstacle-free environment, the sheepdog should be able to collect the sheep successfully.

All actions in the basic and abstract shepherding model are represented using velocity vectors; however, these vectors are called force vectors due to the fact that if the agents are actual vehicles, the desired velocities need to be transformed into forces that cause agents to move. For consistency with the shepherding literature, we will call them (proxies of) force vectors. Below is a list of all force vectors used in this basic model.

- Sheep-Sheepdog Repulsive Force  $F_{\pi_i\beta_j}^t$ : repulsion of  $\pi_i$  agent away from  $\beta_j$  agents at time  $t$ .
- Sheep-Sheep Repulsive Force  $F_{\pi_i\pi_{-i}}^t$ : repulsion of  $\pi_i$  agent away from other  $\pi_{k\neq i}$  agent at time  $t$ .
- Sheep Attraction to Local Herd  $F_{\pi_i\Lambda_i^t}^t$ : attraction to Local Centre of Mass for the neighbours of a  $\pi_i$  agent at time  $t$ .
- Sheep Local Random Movements  $F_{e\pi_i}^t$ : jittering movements by the  $\pi_i$  agent at time  $t$ .
- Sheep Total Force Vector  $F_{\pi_i}^t$ : movement vector of the  $\pi_i$  agent at time  $t$ .
- Sheepdog Attraction to Driving Point  $F_{\beta_j d}^t$ : driving vector of the  $\beta_j$  agent at time  $t$ .
- Sheepdog Attraction to Collecting Point  $F_{\beta_j c}^t$ : collection vector of the  $\beta_j$  agent at time  $t$ .
- Sheepdog Local Random Movements  $F_{e\beta_j}^t$ : jittering movements by the  $\beta_j$  agent at time  $t$ .
- Sheepdog Total Force Vector  $F_{\beta_j}^t$ : movement vector of the  $\beta_j$  agent at time  $t$ .

The total forces acting on the sheep and sheepdog, respectively, are formed by a weighted sum of the individual forces. The weights are explained in **Table 1**. The equations for total forces are included below.

$$F_{\pi_i}^t = W_{\pi\Lambda} \times F_{\pi_i\Lambda_i^t}^t + W_{\pi\pi} \times F_{\pi_i\pi_{-i}}^t + W_{\pi\beta} \times F_{\pi_i\beta_j}^t + W_{e\pi_i} \times F_{e\pi_i}^t + W_{\pi_v} \times F_{\pi_i}^{t-1}$$

$$F_{\beta_j}^t = W_{\beta_j c} \times F_{\beta_j c}^t + W_{\beta_j d} \times F_{\beta_j d}^t + W_{e\beta_j} \times F_{e\beta_j}^t$$

### 3 HUMAN-SWARM TEAMING

Human-human teaming, albeit still a challenging topic, seems natural to the extent that most humans would only focus on the external/behavioural traits required to generate effective teams. The compatibility among humans has hardly been questioned; all humans have a brain with similar structure and while their mental models of the world could be different—thus, requiring alignment for effective teams—the internal physiological machines are similar in the manner they operate. When we discuss teaming among different species, especially when one species is biological (humans, dogs, sheep, etc) and the other is in-silico (computers controlling UGVs, UAVs, etc), some of the factors taken for granted in human-human teaming need to be scrutinised and looked at with a great level of depth.

A particular focus in this paper is the alignment of representation language on all levels of operations inside a machine. Interestingly, in a human, neurons form the nerves



**TABLE 1 |** The Library of behaviors observed or performed by the sheepdog.

Desires	Weight vector
Sheep desire to cluster	$W_{\pi\Lambda}$
Sheep desire to avoid collision	$W_{\pi\pi}$
Sheep desire to avoid sheepdog	$W_{\pi\beta}$
Sheep desire to stay where they are	$W_{e\pi_i}$
Sheep desire to maintain velocity at $t - 1$	$W_{\pi_v}$
Sheepdog desire to collect astray sheep	$W_{\beta^c}$
Sheepdog desire to drive collected sheep	$W_{\beta^d}$
Sheepdog desire to desire to avoid sheep on the way to collection or driving points	$W_{\beta_\pi}$
Sheepdog desire to rest	$W_{e\beta_j}$

that sense, the nerves that control the joints and actuators, and the basic mechanical unit for thinking. While these neurons perform different functions, they work with similar principles. It seems within humans, the representational unit in a nervous system, the neuron, is the unified and smallest representational unit for sensing, deciding and acting. The representational unit of concern in this paper sits at a higher level than the physiological neuron; it sits on the level of thinking and decision making, where it takes the form we called ‘force vectors’.

We will differentiate between three representations as showing in **Figure 2**. The first representation, we call *control-representation*, is one where the action-production logic for an agent is ready for hardware/body/form/shape execution. Control systems at their lower level are executed in a CPU, GPU, FPGA or a neural network chip where the output gets transmitted to actuators. They mostly come in calculus forms. The mathematics of the control system, even for simple ones, are not necessarily on the level to be explainable to a general user. What is important on this level is computational efficiency from sensing to execution to ensure that the agent acts in a timely manner, and the assurance of performance to ensure that the agent acts correctly.

The second representation, we call *reasoner-representation*, is where the agent needs to make inferencing to connect its high level goals and low-level control, create an appropriate set of actions, and select the right courses of actions to achieve its goals or purpose. On this level, the agent performs functions that govern the overall logic that connects its mission to its actions.

The third representation, we call *communication-representation*, is where the agent needs to transform its internal representation for reasoning and action production to external statements to be communicated to other agents, including humans, in its environment. This representation is key for agents to exchange knowledge, negotiate meaning, and be transparent to gain trust of others in their eco-system.

Take for example the artificial *sheepdog*, which we will assume to be a ground autonomous vehicle. Its objective is to collect all dispersed sheep outside the paddock into the paddock area. It senses the environment through its onboard sensors and/or through communication messages received from the larger system it is operating within. Through sensing, it needs the following information to be able to complete its mission: location of sheep, location of goal (paddock), location and size of obstacles in the environment, and location of other *dogs* in the

environment. The *dog* may receive all information about all entities in an accurate and precise form as it is the case in a perfect simulated world, or it may receive incomplete or ambiguous information with noise as is the case in a realistic environment.

On a cognitive-level, the *dog* needs to decide which sheep it needs to direct its attention to, how it will get to them, how it will influence them to get them to move, where to take them, and what to do next until the overall mission is complete. The timescale on which the cognitive level works on is moderate. We will quantify this later in the paper. Meanwhile, whatever the cognitive-level decides, it needs to be transformed into movements and actions.

The control representation takes the relevant subset of the sensed information and the immediate waypoints the *dog* needs to move to and generates control vectors for execution. It needs to transform the required positions decided on by the cognitive level into a series of acceleration and orientation information steps that get transmitted to its actuators (for example, joints and/or wheels). The time scale this level operates on is shorter than what the cognitive-level operates on.

Additionally, the dog may need to communicate with its (human or AI) handler, explaining what it is doing and/or obtaining instructions. The timescale in which the communication operates could vary, and the system needs to be able to adjust this time scale based on the cognitive agents it is teaming with. For example, the communication could occur more frequently if the dog is interacting with another AI than if it was interacting with a human.

Each representational language defines what is representable (capacity), and thus, what is achievable (affordance), using such representation. For example, if the control system is linear, the advantages include: being easy to analyse and being easy to prove/disprove its stability. We equally understand its disadvantages for example in requiring a complete system identification exercise prior to the design of the controller and its inability to adapt when context changes. When two or more representational languages interface with one another, their differences generate challenges and vulnerabilities. We will illustrate this point with the three representational languages discussed for the artificial sheepdog.

The reasoner-representation relies on propositional calculus. A set of propositions can be transformed to an equivalent binary integer programming problem. If the communication-representation is in unrestricted natural language, clearly many sentences exchanged at the interface level will not be interpreted properly by the reasoner. In the same manner, when the control-representation is a stochastic non-linear system, some actions produced by the controller may not be interpretable by the reasoner. This requirement for equivalence at the interface between the three representations impose constraints on which representation to select. Meanwhile, it ensures that actions are interpretable at all levels; thus, what the agent does at the control or cognitive levels can be explained to other agents in the environment, and requests from other agents in the environment can be executed as they are by the agent. Moreover, due to the equivalence in the capacity of the

representation language at each level, mappings between different representations are direct mappings; thus, they are efficient.

Last, but not least, once a system is assured on one level, due to the equivalence of the representation language on all three levels, the system can be easily assured on another level. For example, take the case where a system is assured that it will not request or accept an unethical request. Considering that what the agent executes at the control level is equivalent to the request it receives on the communication level, if we guarantee that the mappings between levels are correct and complete, we can induce that the control level will not produce an unethical behaviour.

Before departing from the discussion above, the representation language plays a dual role in a system. On the one hand, it constrains the system's capacity to perform. As we explained above, a linear system can only be guaranteed to perform well under severe assumptions of linearity. On the other hand, the representation language equally constrains affordance. This may not be so intuitive because affordance is the opportunities that the environment offers an individual agent to do. An agent is unable to tap into opportunities where the representation language constrains its action set or the quality of actions. Due to the trade-offs discussed above, the decision on which representation language to use needs to be risk-based to analyse the vulnerability and remedies of the design choices made during that decision.

In the next section, we will present the requirements for the human-machine teaming problem, including a formal representation of the problem, before presenting the Jingulu swarm language in the following section.

## 4 HUMAN-AI TEAMING LANGUAGE REQUIREMENTS

In human-AI teaming, different categories of information assist in the efficiency of the teaming arrangements and the ability of the system to adapt [33]. However, these capabilities will not materialise unless there is a language that allows this information to flow and to be understood by the humans and the swarm. In this section, we focus primarily on the requirements for this language.

### 4.1 Human-AI Teaming Language Requirements

The discussion and example presented in the previous section illustrate the scope of each of the three levels. From this scope, we will draw and justify the requirements for the Human-Swarm Language as follows:

1. **Contextual Relevance:** The representation languages need to be appropriate for the particular mission the agent is assigned to do. The representation needs to represent, and when necessary, enable the explanation of, the sensorial information in the context within which the agent operates, the logic used for action production, the actions produced at a particular level, the intent of the agent,

and the measures that the agent uses to assess its performance. Put simply, the representation serves the context; everything the context requires should be representable by the chosen language at each level.

2. **Computational Efficiency:** The reasoner-representation works in the middle between the control-representation and the communication-representation. The three representations need to allow the clocks of the three levels to serve each other's frequencies. For example, if the reasoner needs to produce a plan on 0.2 Hz (ie a plan each 5 s), and the controller is running on 1 Hz, while the communication system needs to explain the decisions made in the system on 0.05 Hz, the representation on each level needs to be computationally efficient to allow each level to work on these timescales without latencies. In other words, if it takes 25 s to produce a sentence on the communication layer, the agent will not be able to catch up with the speed of action-production. This latter case will force the agent to be selective in what aspects of its actions it needs to explain, which could generate cognitive gaps in the understanding of other agents in the environment.
3. **Semantic Equivalence:** A reasoner that is producing a plan that can't be transformed intact<sup>1</sup> to the communication-representation will put the dog in a situation that the handler can't understand. Similarly, if the control level is relying on highly non-linear and inseparable functions, it could be very difficult to exactly explain its actions through the communication-level. Representation is a language, and the three representational levels need to be able to map the meanings they individually produce to each other. Semantic equivalence is a desirable feature, which would ensure that any meaning produced on one level has sentences on other levels that can reproduce it without introducing new meaning (ie correctness) or excluding some of the meaning (ie completeness).
4. **Direct Syntactic Mapping:** The easier it is to map each sentence on one level to a sentence on a different level, the less time it will take to translate between different levels. The direct mapping of syntactic structures from one level to another contributes to achieving the two requirements of computational efficiency and semantic equivalence.

In the next sub-section, we will present formal notations to demonstrate the mappings from the internals of shepherding and swarm guidance equations to the external transparent representation enabling the Jingulu-Swarm based communication language. These mappings are essential to ensure that the requirements above have been

<sup>1</sup>While we acknowledge that human communication contains and tolerates ambiguity, we argue that given the current state of technological advances in artificial intelligence systems, it is still difficult to allow for ambiguity to prevail in a human-AI interaction. Therefore, we are constraining the space currently to a bounded set of statements, where meaning is exact; thus, interpretation can be done intact if we further assume that the communication channels do not introduce further noise causing ambiguity in received messages.

taken into account during the design and have been met during the implementation phase of the system.

## 4.2 Interpretability, Explainability and Assurance of Human-Swarm Systems

Abbass et al. [34]. presented a formal definition of transparency towards bi-directional communication in human-swarm teaming systems. The concept of transparency was based on three dimensions, interpretability, explainability and predictability. We will quote these definitions here again for completeness and to allow us to expand on them for human-AI teaming.

Let  $\mathbb{L}$  be a set of languages, where each language  $\mathcal{L}_i \in \mathbb{L}$  is a set of sentences,  $\mathbb{S}_\beta$ . In this world view, any sentence in any language will be a member of the superset  $\mathbb{L}$ ; while noting that communicative sentences by an agent are interpretations of internal knowledge statements  $\mathbb{K}_\beta$  of the agent. The mapping from internal knowledge representation  $\mathbb{K}_\beta$  to a sentence  $\mathbb{S}_\beta$  is achieved by a transformation function  $\mathbb{R}$ . The reverse is achieved by  $\mathbb{R}^{-1}$ . Below is the set of definitions quoted from [34].

**Definition 4.1 (Interpretability).**  $\mathbb{I}$  is an interpretation function that maps a sentence in one language to a sentence in a second, potentially the same, language; that is,

$$\mathbb{I}: \mathbb{S}_m \rightarrow \mathbb{S}_n, \mathbb{S}_m \in \mathcal{L}_i, \mathbb{S}_n \in \mathcal{L}_j, \mathcal{L}_i, \mathcal{L}_j \in \mathbb{L} \quad (2)$$

**Definition 4.2 (Explainability).**  $\mathbb{E}$  is an explanation function iff, given a hypothesis  $\mathbb{S}^{c_0 \rightarrow e}$ , there exists,

$$\mathbb{E}: \mathbb{S}^{c_0 \rightarrow e} \rightarrow \{\mathbb{S}^{c_0}, \mathbb{S}^{c_0 \rightarrow c_1}, \mathbb{S}^{c_1 \rightarrow c_2}, \dots, \mathbb{S}^{c_{k-1} \rightarrow c_k}, \mathbb{S}^{c_k \rightarrow e}\} \quad (3)$$

where  $\mathbb{R}^{-1}(\mathbb{S}^{c_i}) \in \mathbb{K}_\beta$

**Definition 4.3 (Predictability).**  $\mathbb{P}$  is a prediction function that takes a subset of axioms and facts, and projects them through induction or abduction onto a different set of axioms or facts; that is,

$$\mathbb{P}: \mathbb{S}^c \rightarrow \mathbb{S}^e \quad (4)$$

**Definition 4.4 (Transparency).**  $\mathbb{T}$  is a transparency function that decides on the level of interpretability, explainability and predictability that will be visible from one agent to another agent, then implicitly or explicitly forms the language it will use to communicate this information to the other agents; that is,

$$\mathbb{T}: \{\mathbb{I}, \mathbb{E}, \mathbb{P}\} \rightarrow \mathcal{L}_j \quad (5)$$

**Figure 3** depicts the coupling of the internal decision making of an agent with the above definitions, leading to a transparent human-AI teaming setting. Without loss of generality, we will assume in our example that agents have complete and certain information. The relaxation of this assumption does not change the modules in the conceptual diagram in **Figure 3**, but rather, it

changes the design choices and complexity of implementing each module.

In **Figure 3**, an agent senses two types of states, its internal agent states reflecting its self-awareness, and the environment's states representing the states of other agents and the space it is located within. In herding, the sheepdog needs to sense its own position location (its own state), and the states of the environment, which consists of the position locations of sheep and the goal. The sheepdog needs to decide on its goal. This goal-setting module could choose the goal by listening to a human commanding the sheepdog or the sheepdog could have its own internal mechanism for goal setting as in autonomous herding. The goal could be mustering, where the aim is to herd the sheep to a goal location.

Based on the goal of the sheepdog, its state and the state in the environment, the agent needs to select an appropriate behaviour. The behavioural database contains two main behaviours in this basic model: a collecting and a driving behaviour based on the radius calculated in **Eq. 1**. Once a behaviour is selected, a planner is responsible for sequencing the series of local movements by the sheepdog to achieve the desired behaviour. For example, if the behaviour is to "drive," the sheepdog needs to reach the driving point using a path that does not disturb the sheep then modulate its force vectors on the sheep to drive them to the goal location. The planner will generate a series of velocity vectors for the sheepdog to follow. While the planner has an intended state, in a realistic setup, the desired state by the planner may be different from the actual state achieved in the environment due to many factors including noise in the actuators, terrain, weather, or energy level. The state update function is the oracle that takes the actions of the agents and updates the states of the agent and the environment.

The above description explains how the sheepdog makes decisions. However, an external agent, be it a human or artificial, needs to operate effectively as a teammate. Therefore, the sheepdog needs to be able to communicate the rationale of its decisions. The three factors for transparency mentioned above are critical in this setting [35]. Explainability provides teammates the reasons why certain goals, behaviours, and actions were selected at a particular point of time. Predictability offers the information for teammates to be ready for future actions of the sheepdog; thus, it reduces surprises which could negatively impact an agent's situation awareness and trust. The sheepdog is operating with force vectors as explained above. However, an external teammate may not understand these force vectors or may get overloaded when a sheepdog storms it with a large number of force vectors. This is where the force vectors generated by the explainability and predictability module need to be transformed into a language that the agent can use to interact with other agents. This language needs to be bidirectional; that is, humans and artificial agents need to be able to exchange sentences in this language that they can transform them into their own internal representation. In the case of sheepdogs, the plain English sentences need to be transformed to force vectors and vice-versa.



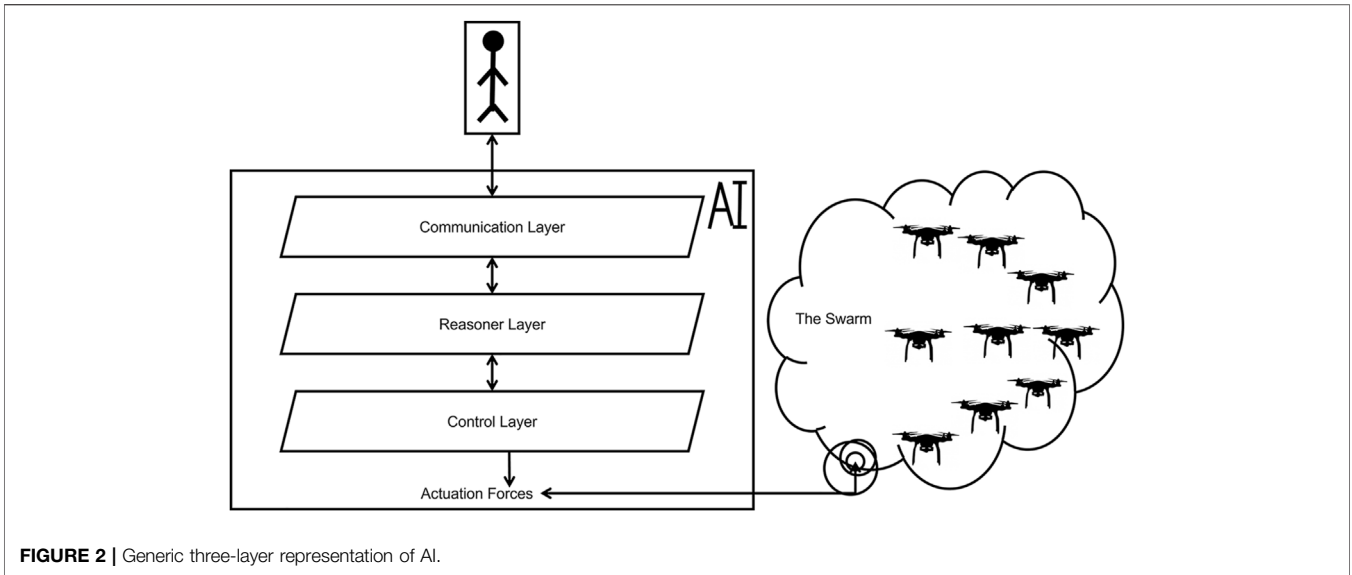


FIGURE 2 | Generic three-layer representation of AI.

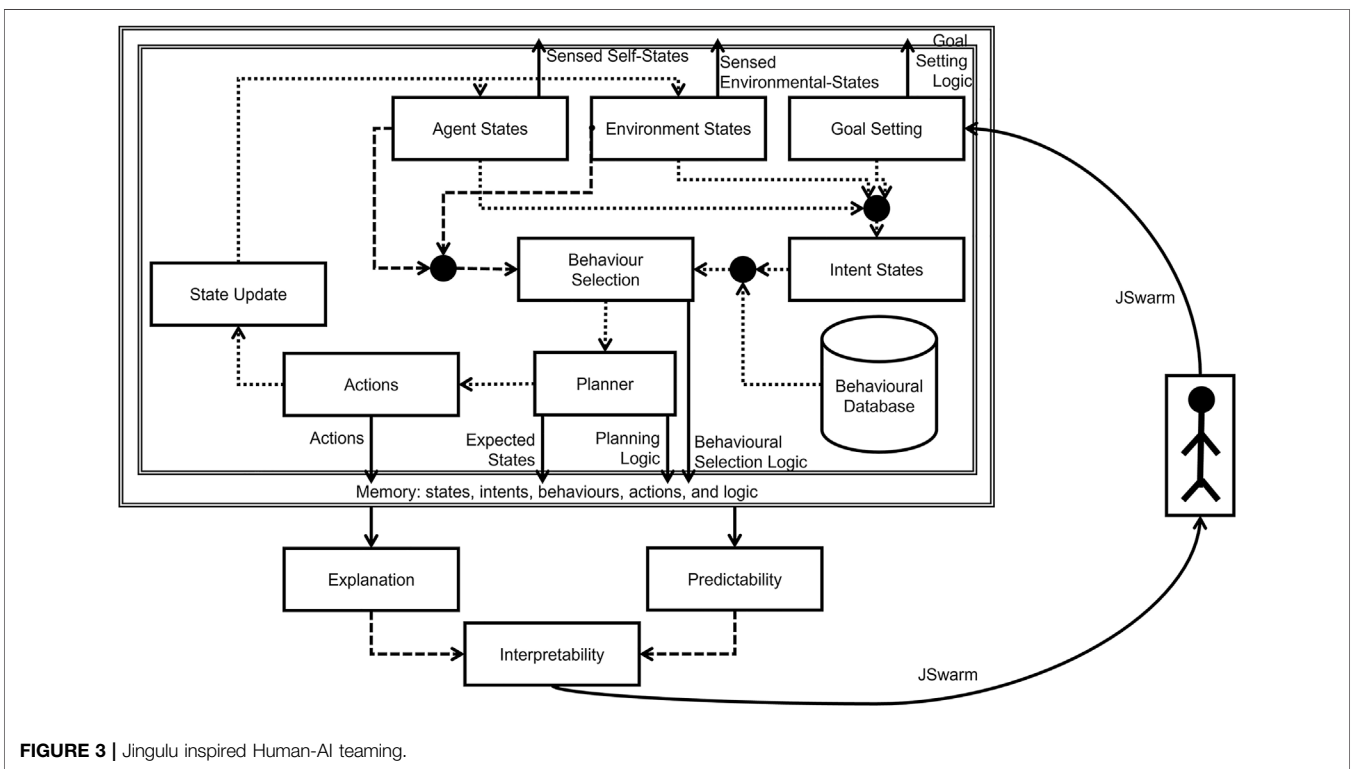


FIGURE 3 | Jingulu inspired Human-AI teaming.

## 5 JSWARM: A JINGULU-INSPIRED HUMAN-SWARM LANGUAGE

Similar to Boids, the herding system transforms all swarm actions into attraction and repulsion equations. In principle, a whole mission can be encoded in this system. Let us revisit sheep herding as a mission example to illustrate the application and

efficiency of the JSwarm language. We recall herding as a library of behaviors that gets activated based on the sheep’s and the dog’s understanding of a situation. We will use the definitions of a context and a situation as per [36], where a context is “the minimum set of information required by an entity to operate autonomously and achieve its mission’s objectives,” while a situation is defined as “a manifestation of invariance in a

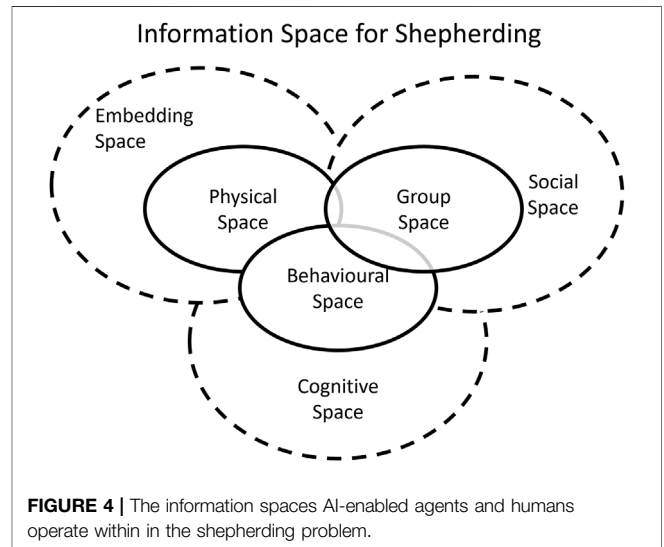
**TABLE 2 |** The Library of behaviors observed or performed by the sheepdog.

Situation	Behavior	Force vector	Sentence
Sheepdog detected	Sheep Attraction to Center of Mass	$F_{\pi, A_i}^t$	COME to CM
Sheepdog detected	Sheep Repulsion from Nearby Sheep	$F_{\pi, \pi_i}^t$	GO away from Nearby Sheep
Sheepdog detected	Sheep Repulsion from sheepdog	$F_{\pi, \beta_j}^t$	GO away from sheepdog
Sheepdog not detected	Sheep perform small local random movements	$F_{\pi, \epsilon}^t$	GO Random Direction and Steps
Always	Sheep move towards velocity at $t - 1$	$F_{\pi}^{t-1}$	GO velocity(t-1)
Astray sheep detected	Sheepdog Attraction to Collection Point	$F_{\beta, C}^t$	COME to Collection Point
Sheep Collected	Sheepdog Attraction to Driving Point	$F_{\beta, D}^t$	COME to Driving Point
Sheep detected on the path	Sheepdog Repulsion from nearby sheep	$F_{\beta, s}^t$	GO away from nearby sheep
Chaos or natural dynamics	Sheepdog remains at current location	$F_{\beta, \epsilon}^t$	DO nothing

subset of this minimal information set over a period of time.” In the herding example, different contexts and situations may activate different behaviors. We will illustrate these by categorizing all behaviors as either attraction, repulsion or no-movements. We will commence our description of the language by mapping all attractions, repulsions and no-movements forces to sentences with the verbs “COME,” “GO,” and “DO,” respectively. We will assume for simplicity that the dog has complete accurate knowledge of all agents in the environment, including how the sheep moves. This is only for convenience for this first iteration of the language to avoid adding unnecessary complexity.

**Table 2** summarises the basic behaviors in shepherding. It is worth noting that any other required behavior could still be explained using the attraction-repulsion system. For example, if the agent wishes to speak, words are directed towards an audience and thus, the audience becomes the attraction point or the agent becomes the repulsion point of the message. When an agent wishes to drop a parcel, the target location of the parcel becomes the attraction point and the agent becomes the repulsion point. When an agent wishes to eat, the food-store becomes the repulsion point of the food and the agent becomes the attraction point. In each of these examples, there is a frame of reference, such as the agent, where the world is seen from that agent’s perspective. In this world view, every behavior in a mission of any type could be encoded as movements in a space. For example, this is the underlying fundamental concept of a transition in a state space, where a transition is a movement from one state/location in a space to another. When modelling flow of ideas, an idea is either sent to an agent (attracted to the agent), created by an agent (the doing of an agent), or brainstormed by an agent (doing of an agent). The attraction-repulsion system assumes that things move in one or more spaces. While the information on space and time are not required in the simple shepherding example, it is very important to include information on space and time in our description of the language when agents operates in different spaces or on different timescales to ensure that the language is general enough to capture these complexities.

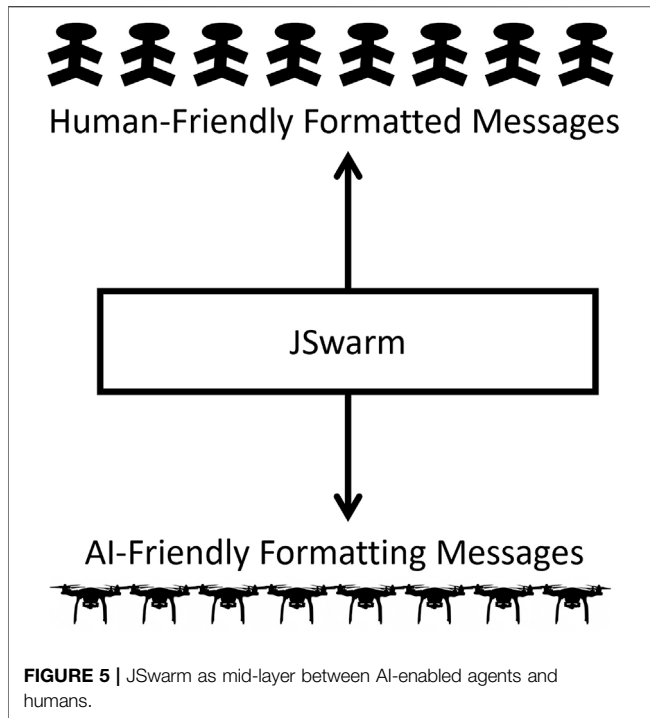
Each agent has its context encoded in a state-vector, representing the super-set of all spaces an agent needs to be aware of. For example, in shepherding, three spaces are



**FIGURE 4 |** The information spaces AI-enabled agents and humans operate within in the shepherding problem.

important, the physical space affording an agent with information on the location of each other agent in the environment, the group space providing information on the state of groups and members in the groups (whether the herd is clustered or not, there is astray sheep or not, if so, how many stray sheep) and the behavioral space representing the type of perceived or real behavior an agent or a group of agents are performing. The previous three spaces are sub-spaces of larger spaces as shown in **Figure 4**. Decisions are made by transforming sensorial information into particular spaces that an agent operate on. We call these the embedding spaces, a concept familiar to AI researchers working with natural language processing. The physical space is a subset of the embedding space. The behavioural space could be seen as the externalisation of actions generated in the cognitive space. The grouping aspects in shepherding are just a subset of the social space that could extend to social ties and relationships.

The cognitive level of an agent may focus on generating movements in the group and behavioral spaces. For example, the sheepdog would want to collect the astray sheep so that the



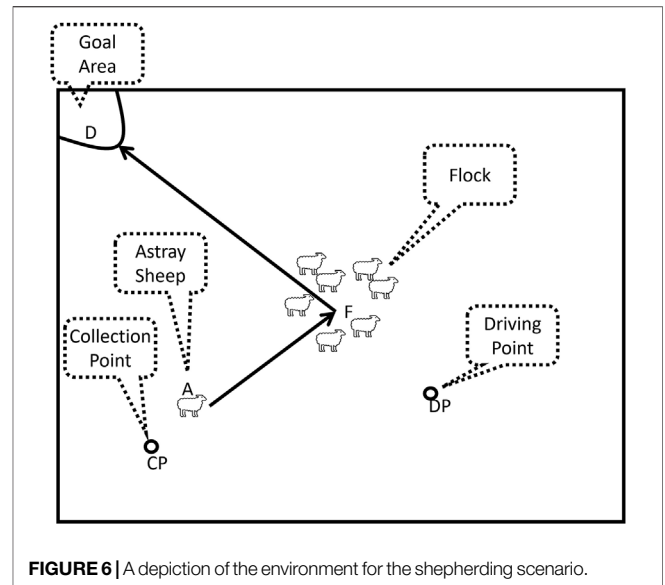
group of sheep is clustered together. These movements can be achieved through a plan that will then generate a sequence of actions requiring the sheepdog to move in the physical space. The plan and its associated actions need to be executed within a particular time-frame; thus, sentences may need to be parameterised with time.

The previous discussion offers the rationale for the design choice of the JSwarm language, which is presented formally next.

JSwarm has three types of sentences: sentences to communicate behaviors, sentences to communicate intent, and sentences to communicate state information. The state sentences are relevant when the AI communicates to humans its states. The behavioural systems are more relevant for action-production. The intent statements are crucial for commanding and communicating goals to the AI. **Figure 5** depicts the position of JSwarm as a formalism that sits between humans and AI agents. Each of the three sentence types has a structure explained below. We describe it in a predicate-logic-like syntax as an intermediate representation for humans and AI-enabled agents.

- Behavior: *Subject.Verb(Space.TimeDelay. TimeDuration).SupportingVerb(Space. TimeDelay.TimeDuration).Object*
- Intents: *Subject.DO.SupportingVerb.Object*
- States: *Subject.DO.SupportingVerb(Space.TimeDuration).Object*

We will present below examples to cover the space of each sentence type mentioned above. We assume all spaces represented in Cartesian coordinates and time in seconds.



- Behaviour: *Subject.Verb(Space.TimeDelay. TimeDuration).SupportingVerb(Space.TimeDelay. TimeDuration).Object*
  - Sheep4.Go((x = 50,y = 70).0.50).Escape(NULL.0.30).Dog1 Now (0 delays), Sheep Sheep4 needs to go to location (50,70) within 50 s and escape dog Dog1 for 30 s.
  - Sheep2.Come((x = 20,y = 15).0.30).Group(NULL.0.10).Herd1 Within 30 s from now, sheep Sheep2 needs to arrive at location (20,15) to group within 10 s with herd Herd1.
  - Dog1.Come((x = 40,y = 40).0.30).Collecting(NULL.0.20). Sheep5 Within 30 s from now, dog Dog1 needs to start moving to arrive at location (40,40) and spend 20 s collecting sheep Sheep5.
  - Dog1.Come((x = 70,y = 20).0.30).Driving(NULL.0.15).Herd0 Within 30 s from now, dog Dog1 needs to start moving to arrive at location (70,20) to drive Herd0 for 15 s.
  - Herd1.Do(NULL.0.0).Sitting(NULL.0.20) Herd Herd1 needs to stay in its current location for 20 s.
  - Dog1.Do(NULL.10.0).Sitting(NULL.0.30) Dog Dog1 needs to wait for 10 s then sit in its location for 30 s.
- Intents: *Subject.DO.SupportingVerb.Object*
  - Dog1.Do.Herd.Herd1 Dog Dog1 needs to herd group Herd1.
  - Herd1.Do.Escape.Dog1 Herd Herd1 needs to escape dog Dog1.
- States: *Subject.SupportingVerb(TimeDuration).Object*
  - Dog1.Do.Collecting(120).Many Dog Dog1 is collecting many sheep for 2 min.
  - Dog1.Do.Collecting(160).Sheep5 Dog Dog1 is collecting sheep Sheep5 for 3 min.
  - Dog1.Do.Driving(60).All Dog Dog1 is driving all sheep for 1 min.
  - Dog1.Do.Driving(30).Herd0 Dog Dog1 is driving herd Her0 for 30 s.

- Dog1.Do.Patrolling(3600).All  
Dog Dog1 is patrolling all sheep for an hour.
- Sheep1.Do.Foraging(1,200)  
Sheep Sheep1 is foraging for 20 min.
- Sheep1.Do.Escaping(60).Herd1  
Sheep Sheep1 is escaping herd Her1 for 1 min.

JSwarm is designed to be a transparent human-swarm language in a manner consistent with Eq. 5 and the definition of transparency given in the previous section. As shown in Figure 5, JSwarm sits at the middle layer between humans and AI-enabled swarms. We provided in this paper the direct mappings from JSwarm to force vectors and vice-versa. This form of interpretability is both sound and complete. The example presented in the following section will demonstrate explainability, where the logs of the JSwarm is a series of expressions providing the intents and actions taken by the AI-enabled swarm. Predictability relies on the mental model formed within an agent's brain (including computers) for an agent to predict another. Thanks to the 1-to-1 mapping in JSwarm, and the simplicity of the abstract shepherding problem presented in this paper, predictability is less of a concern. While we argued that JSwarm is transparent in the abstract shepherding problem, our future work will test this hypothesis in more complex environments.

## 6 HUMAN-SWARM LANGUAGE DEMONSTRATION

The JSwarm language could generate significant amount of sentences due to its ability to work on the level of atomic action. It could also generate very comprehensive sentences due to its ability to work on the behavioural space. The design relies on the definitions provided in the previous section, where interpretability is the mapping from the equations of shepherding to JSwarm syntax presented in this section, and explainability is the outcome of the sequence of expressions produced by JSwarm to explain the sequence of behaviours presented by an agent. In this section, we will present a scenario for shepherding to demonstrate the use of the JSwarm language.

Consider a case of a  $100 \times 100$  m paddock with the goal situated at the top left corner, the sheep are spread around the centre point with an astray sheep at location (20,20), and the dog at the goal location. The dog has complete and accurate situation awareness of the location of all sheep. Its internal logic determines that it needs to activate its collecting behaviour to move around the edge of the paddock to reach the collection point behind the astray sheep. The collection point is at location (15,15), where its location sits on the direction vector from the location of the sheep (20,20) to the location of the centre of the flock (50,50). The characters in this scenario are labelled D for the dog, A for the astray sheep, and F for the flock. The dog needs to

communicate its actions every 5 s or when it selects a different behaviour. Below is a series of messages announced by the dog in JSwarm to indicate its actions and what it perceives in the environment. This environment is depicted in Figure 6.

Dog.Do.Herd.F % Intent communicated that the dog needs to herd the flock.

Dog.Do.Collecting.A % Intent communicated that the dog is collecting astray sheep A

Dog.Come(x = 15,y = 15).Collecting.CP % Dog is on its way to collection point CP for astray sheep.

The above sentence repeats until Dog reaches the collection point, CP.

Dog.Come(x = 50,y = 50).Collecting.A % Dog is collecting astray sheep in the direction of the flock centre

The above sentence repeats until sheep A joins the flock or the dog drifts away from the collection point. We assume the latter, at which point in time, the dog needs to move towards the new location of the collection point.

Dog.Come(x = 30,y = 30).Collecting.CP % Dog on its way to new collection point for astray sheep

Dog.Come(x = 50,y = 50).Collecting.A % Dog collecting astray sheep in the direction of the flock centre

The above sentence repeats until sheep A joins the flock.

Dog.Do.Driving.F % Dog's intent change to driving the flock  
Dog.Come(x = 75,y = 75).Driving.DP % Dog on its way to driving point, DP, for the flock

The above sentence repeats until the dog reaches the driving point.

Dog.Come(x = 100,y = 100).Driving.F % Dog is driving the flock F towards the goal

The above sentence repeats until sheep are at the goal.

Dog.Do.Rest

While in the above example, we focused on the dog communicating its actions, the example could get extended where the dog communicates also its situation awareness, the sheep communicates their actions and situation awareness as individuals, and the AI communicates the sheep flock actions. It is important to notice that in the above example, we did not use the time parameters in the language due to the fact that the simulation for abstract shepherding is normally event-driven rather than clock-driven. We could also decide to represent spatial locations in other formats. The exact representation of the parameters is a flexible user choice.



## 7 CONCLUSION AND FUTURE WORK

Effective human-AI teaming requires a language that enables bidirectional communication between the humans and the AI agents. While human natural languages could be candidates, we explained that the richness of these languages come with a cost of increased ambiguity. The humans and the swarm need to communicate in an unambiguous manner to reduce confusion and misunderstanding. Consequently, we defined four main requirements in the design choice of a language for human-AI teaming; then we presented a language inspired by the Jingulu language, an Australian Aboriginal language.

The JSwarm language is the first of its kind Human-AI Teaming language that is based on direct mappings from the internal logic, including the equation of motion, of an agent to a human-friendly language. The language is designed to accurately reflect the internal attraction-repulsion equations governing the dynamics of a swarm, including states, intent and behaviours. JSwarm allows humans and swarms to communicate with each other without ambiguity and in a form that could be verified. The language separates semantics from syntax, where the supporting verb acts as a semantic carrier. While the light verb impacts syntax, the supporting verb does not affect the syntax of an expression, thus allowing semantics to be associated and de-associated freely. This latter feature could utilise an ontology, and allows the syntactical-layer of the language to remain intact as it gets applied to different domains, while a replacement of the ontology changes the semantic layer. Moreover, the free word order feature in the language could offer a robust communication setting, where meaning is maintained even if the receiver orders the words differently.

We proposed a simple grammar and representation of sentences in the language, which was intentionally selected such that it mimics the structure of a first-order logical representation, while being semantically-friendly to human comprehension. We concluded the paper with an example to showcase how the language could be used to provide a real-time log for a dog to communicate its actions in a human-friendly language.

For our future work, we will extend the design to connect the JSwarm language as it works on the communication layer with the representations used on the control and reasoner layers. We will also conduct human studies to evaluate the efficacy of JSwarm. It is important that the human usability study to evaluate the effectiveness of JSwarm takes place with a complex scenario with appropriate architectures and implementations of the swarm. The risk of testing the concept in a simple scenario is that the human will find the scenario trivial and the need for explanation unwarranted.

## REFERENCES

1. Coulmas F. *Sociolinguistics: The Study of Speakers' Choices*. Cambridge, UK: Cambridge University Press (2013).
2. Holmes J, Wilson N. *An Introduction to Sociolinguistics*. Abingdon-on-Thames, Oxfordshire United Kingdom: Taylor & Francis Group (2022).

While the JSwarm language is explained using a shepherding example to make the paper accessible to a larger readership, the language is designed to be application-agonistic and could benefit any problem where communication between humans and a large number of AI-enabled agents is required. For example, a swarm of nano-robots combatting cancer cells could offer a perfect illustration where the robots have a very simple logic that needs to be transformed to an explanation to the medical practitioner overseeing the operation of the system. The medical practitioner equally needs a language to command the swarm that is simple to match the internal swarm logic and reduces communication load. In these applications, the sheepdog could be a chemical substance that the swarm of nano-robots react to, which is controlled by an external robot that the medical practitioner needs to command to guide the swarm. Other applications include a swarm of underwater vehicles cleaning the ocean or in the mining industry, a swarm of uncrewed aerial vehicles surveying a large area.

## DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

## AUTHOR CONTRIBUTIONS

HA conceptualised and wrote the first draft. EP contributed to the linguistic analysis. RH contributed to conceptualising the impact of the work on human-AI teaming. All authors revised and edited the paper.

## FUNDING

This work is funded by Grant No N62909-18-1-2140 from Office of Naval Research Global (ONRG). Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the ONRG.

## ACKNOWLEDGMENTS

The authors wish to thank Adam Aly for role playing the AI to formulate an example for this paper, and Beth Cardier, Adam Hepworth, and Aya Hussein for providing constructive feedback on the work. We also wish to thank the reviewers for their constructive comments.

3. Plun JY, Wilcox CD, Cox KC. *Swarm Language Reference Manual*. Seattle, Washington: Washington University in St. Louis (1992).
4. Vaniya S, Lad B, Bhavsar S. A Survey on Agent Communication Languages. *2011 Int Conf Innovation, Management Serv* (2011) 14: 237–42.
5. Pantelimon G, Tepe K, Carriveau R, Ahmed S. Survey of Multi-Agent Communication Strategies for Information Exchange and mission Control

- of Drone Deployments. *J Intell Robot Syst* (2019) 95:779–88. doi:10.1007/s10846-018-0812-x
6. Bocklich T, Faulkner J, Pawlowski N, Nichol A. *Rasa: Open Source Language Understanding and Dialogue Management*. arXiv preprint (2017).
  7. Singh S, Beniwal H. A Survey on Near-Human Conversational Agents. *J King Saud Univ - Computer Inf Sci* (2021) 2021. doi:10.1016/j.jksuci.2021.10.013
  8. Almansor EH, Hussain FK. Survey on Intelligent Chatbots: State-Of-The-Art and Future Research Directions. In: *Conference on Complex, Intelligent, and Software Intensive Systems*. Cham, Switzerland: Springer (2019). p. 534–43. doi:10.1007/978-3-030-22354-0\_47
  9. Su PH, Mrkšić N, Casanueva I, Vulić I. Deep Learning for Conversational AI. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorial Abstracts*. Pennsylvania, US: Association for Computational Linguistics (2018). p. 27–32. doi:10.18653/v1/n18-6006
  10. Cambier N, Miletitch R, Frémont V, Dorigo M, Ferrante E, Trianni V. Language Evolution in Swarm Robotics: A Perspective. *Front Robot AI* (2020) 7:12. doi:10.3389/frobt.2020.00012
  11. Pensalfini R. *A Grammar of Jingulu, an Aboriginal Language of the Northern Territory*. Canberra, Australia: The Australian National University (2003).
  12. Abbass HA, Hunjet RA. Smart Shepherding: Towards Transparent Artificial Intelligence Enabled Human-Swarm Teams. In: *Shepherding UxVs for Human-Swarm Teaming: An Artificial Intelligence Approach*. Cham, Switzerland: Springer (2020). p. 1–28. doi:10.1007/978-3-030-60898-9\_1
  13. Austin P, Bresnan J. Non-configurationality in Australian Aboriginal Languages. *Nat Lang Linguist Theor* (1996) 14:215–68. doi:10.1007/bf00133684
  14. Simpson J. *Warlpiri Morpho-Syntax: A Lexicalist Approach*. Berlin, Germany: Springer Science & Business Media Dordrecht (1991).
  15. Pensalfini R. The Rise of Case Suffixes as Discourse Markers in Jingulu-A Case Study of Innovation in an Obsolete Language\*. *Aust J Linguistics* (1999) 19: 225–40. doi:10.1080/07268609908599582
  16. Reynolds CW. Flocks, Herds and Schools: A Distributed Behavioral Model. *SIGGRAPH Comput Graph* (1987) 21:25–34. doi:10.1145/280811.28100810.1145/37402.37406
  17. Isobe M, Helbing D, Nagatani T. Experiment, Theory, and Simulation of the Evacuation of a Room without Visibility. *Phys Rev E Stat Nonlin Soft Matter Phys* (2004) 69:066132. doi:10.1103/PhysRevE.69.066132
  18. Nalepka P, Kallen RW, Chemero A, Saltzman E, Richardson MJ. Herd Those Sheep: Emergent Multiagent Coordination and Behavioral-Mode Switching. *Psychol Sci* (2017) 28:630–50. doi:10.1177/0956797617692107
  19. Paranjape AA, Chung S-J, Kim K, Shim DH. Robotic Herding of a Flock of Birds Using an Unmanned Aerial Vehicle. *IEEE Trans Robot* (2018) 34: 901–15. doi:10.1109/TRO.2018.2853610
  20. Kakalis NMP, Ventikos Y. Robotic Swarm Concept for Efficient Oil Spill Confrontation. *J Hazard Mater* (2008) 154:880–7. doi:10.1016/j.jhazmat.2007.10.112
  21. Cohen D. Cellular Herding: Learning from Swarming Dynamics to Experimentally Control Collective Cell Migration. *APS March Meet Abstr* (2019) 2019:F61.
  22. Long NK, Sammut K, Sgarioni D, Garratt M, Abbass HA. A Comprehensive Review of Shepherding as a Bio-Inspired Swarm-Robotics Guidance Approach. *IEEE Trans Emerg Top Comput Intell* (2020) 4:523–37. doi:10.1109/tetci.2020.2992778
  23. Tang J, Leu G, Abbass HA. Networking the Boids Is More Robust against Adversarial Learning. *IEEE Trans Netw Sci Eng* (2017) 5:141–55.
  24. Lien JM, Rodríguez S, Malric JP, Amato NM. Shepherding Behaviors with Multiple Shepherds. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*; 18–22 April 2005; Barcelona, Spain. New York, US: IEEE (2005). p. 3402–7.
  25. Miki T, Nakamura T. An Effective Simple Shepherding Algorithm Suitable for Implementation to a Multi-Mobile Robot System. In: *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*; 30 August 2006 - 01 September 2006; Beijing, China. New York, US: IEEE (2006). p. 161–5. doi:10.1109/ICICIC.2006.411
  26. Strömbom D, Mann RP, Wilson AM, Hailes S, Morton AJ, Sumpter DJT, et al. Solving the Shepherding Problem: Heuristics for Herding Autonomous, Interacting Agents. *J R Soc Interf* (2014) 11:20140719. doi:10.1098/rsif.2014.0719
  27. Kalantar S, Zimmer U. A Formation Control Approach to Adaptation of Contour-Shaped Robotic Formations. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*; 09–15 October 2006; Beijing, China. New York, US: IEEE (2006). p. 1490–7. doi:10.1109/IROS.2006.281977
  28. Razali S, Meng Q, Yang S-H. Immune-inspired Cooperative Mechanism with Refined Low-Level Behaviors for Multi-Robot Shepherding. *Int J Comp Intel Appl* (2012) 11:1250007. doi:10.1142/s1469026812500071
  29. Bat-Erdene B, Mandakh O-E. Shepherding Algorithm of Multi-mobile Robot System. In: *2017 First IEEE International Conference on Robotic Computing (IRC)*; 10–12 April 2017; Taichung, Taiwan. New York, US: IEEE (2017). p. 358–61. doi:10.1109/IRC.2017.51
  30. Masehian E, Royan M. Cooperative Control of a Multi Robot Flocking System for Simultaneous Object Collection and Shepherding. In: *Computational Intelligence*. Cham, Switzerland: Springer (2015). p. 97–114. doi:10.1007/978-3-319-11271-8\_7
  31. Nalepka P, Kallen RW, Chemero A, Saltzman E, Richardson MJ. Herd Those Sheep: Emergent Multiagent Coordination and Behavioral-Mode Switching. *Psychol Sci* (2017) 28:630–50. doi:10.1177/0956797617692107
  32. El-Fiqi H, Campbell B, Elsayed S, Perry A, Singh HK, Hunjet R, et al. The Limits of Reactive Shepherding Approaches for Swarm Guidance. *IEEE Access* (2020) 8:214658–71. doi:10.1109/access.2020.3037325
  33. Hussein A, Ghignone L, Nguyen T, Salimi N, Nguyen H, Wang M, et al. Characterization of Indicators for Adaptive Human-Swarm Teaming. *Front Robot AI* (2022) 9:745958. doi:10.3389/frobt.2022.745958
  34. Abbass H, Petraki E, Hussein A, McCall F, Elsayed S. A Model of Symbiogenesis: Machine Education and Communication as Pillars for Human-Autonomy Symbiosis. *Phil Trans R Soc A* (2021) 379:20200364. doi:10.1098/rsta.2020.0364
  35. Hepworth AJ, Baxter DP, Hussein A, Yaxley KJ, Debie E, Abbass HA. Human-swarm-teaming Transparency and Trust Architecture. *IEEE/CAA J Automatica Sinica* (2020) 8:1281–95.
  36. Fernandez-Rojas R, Perry A, Singh H, Campbell B, Elsayed S, Hunjet R, et al. Contextual Awareness in Human-Advanced-Vehicle Systems: A Survey. *IEEE Access* (2019) 7:33304–28. doi:10.1109/access.2019.2902812

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Abbass, Petraki and Hunjet. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.