14

Local Methods for Supporting Grounded Communication in Robot Teams

Nathan Wiebe and John Anderson

Autonomous Agents Laboratory, Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada R3T2N2
andersj@cs.umanitoba.ca

Abstract. For a mobile robot to be able to communicate usefully with others in a group, the references it makes to points in space must be grounded in concepts that are shared among the group. In the past it has been common to hand-construct a complete set of such groundings, either by individual enumeration or by enforcement of a common coordinate system and origin among all team members. Such assumptions remove the ability to add new robots with no knowledge of the environment in an ad hoc manner, and also require knowledge which may not be available. In an urban search and rescue (USAR) setting, for example, robots may be released into rubble from a collapsed building with no shared starting point for an origin, under conditions where GPS reception is disrupted. Preconstructed groundings are also anthropocentric in that they are a best guess by humans as to what is useful from their perspective, and may be nothing like what robotic agents would come up with on their own. This chapter describes the an approach that allows a group of robotic agents to develop consistent shared groundings for useful locations in an environment over time, using only local communication and interaction. This approach is thus suitable for domains in which broadcast communication may be sporadic, such as USAR, or jammed, such as military applications. The evaluation of this approach, which compares several different grounding techniques, shows that a consistent set of shared groundings can be developed effectively by a team of robots over time using only local interactions, and that these improve the effectiveness of communication in a multi-robot setting.

14.1 Introduction: Grounded Communication in Multi-robot Systems

In order to realize the benefits of deploying a team of robots, individuals are often required to communicate to coordinate their activities. While the benefits of communication to everyday human cooperative activities are easily observed, communication has also been empirically shown to be of great benefit when solving problems in multi-agent systems. Matarić [16] showed that communication was necessary to deal with credit assignment problems in multi-agent learning, for example, while Balch and Arkin [3] showed the advantages of employing communication in team-based foraging tasks. Even in domains such as robotic rescue, where interference and disruption of infrastructure may limit communication to short ranges or line-of-sight, communication can still be effective in improving the group navigation and mapping of a robot team.

Communication, however, can only be of use to the degree that the symbols used for communication are associated with (*grounded* to) concepts in the environment

D. Liu, L. Wang, and K.C. Tan (Eds.): Design and Control of Intel. Robotic Systems, SCI 177, pp. 279–301.
springerlink.com

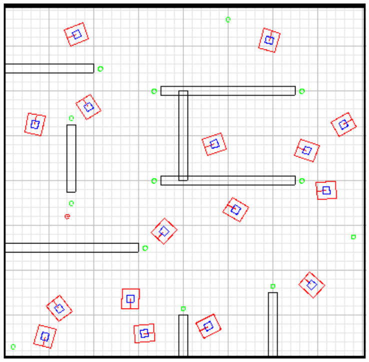

Fig. 14.1. 16 robots in an 8m x 8m domain

consistently among communicating agents. The problem of creating and maintaining these groundings is known as the *symbol grounding problem* [14]. In most multi-agent domains, groundings across a population of agents are hand-constructed. While this is certainly possible to do for small environments, it becomes unwieldy very quickly as the complexity of the environment increases, and simply will not scale to environments of any reasonable size [15]. Such hand-constructed groundings are also biased to the perspective of the humans creating them, as opposed to what might be arrived at by the agents in the domain themselves [15].

The problem of scale in providing groundings becomes much more significant when agents are mobile. In mobile settings, robotic agents must typically communicate spatial information, and thus have potentially a very large number of grounded physical locations that they might wish to communicate to others. Typically a shared coordinate system and the ability to localize well within it are provided by system designers, so that all references to locations become absolute. Robots entering a collapsed building from different starting points, for example, might share a common GPS coordinate system and know one another's starting location to exchange partial maps in a setting such as that shown in Fig. 14.1.

When such a shared coordinate system is unavailable, being able to reference spatial locations in communication between members of a group in a sensible way becomes a very difficult problem. If GPS were unavailable, for example, which could happen due to interference (e.g. in a disaster setting), locale (e.g. underground) or nonexistence (e.g., in a military situation where signals may be jammed, or in space exploration applications), each robotic agent would have to begin with its own coordinate system. Common references would then have to be built up across a population over time, so that an agent could refer to a location by name and have others understand it in terms of their own internal coordinate systems. The difficulty inherent in requiring multiple agents to learn groundings is that in isolation, different agents will naturally ground locations using different symbols, which is obviously not suitable for the purposes of shared reference. Moreover, when agents begin at different coordinates and orientations, there is no way to immediately relate shared groundings. Consider Fig. 14.2, which shows

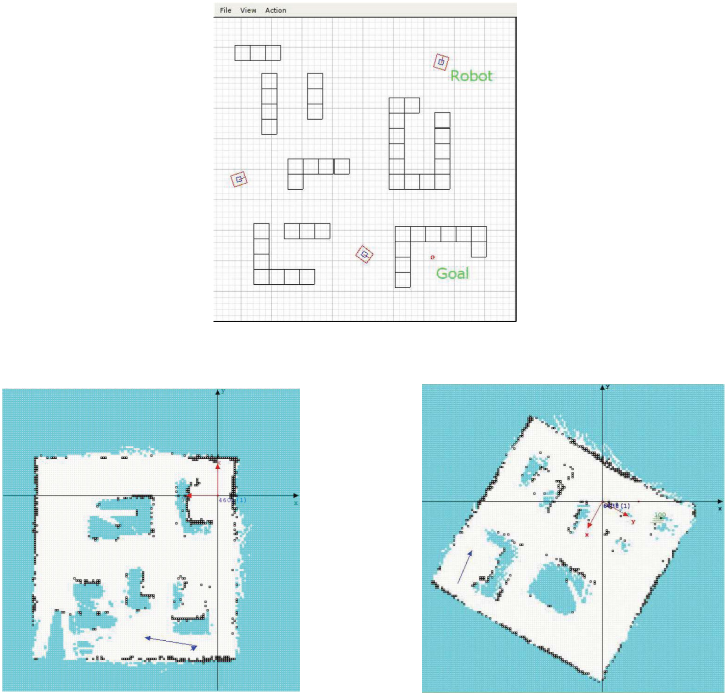

Fig. 14.2. Three robots in an office environment in our implementation. Left, objective view; Center/Right, internal maps of two robots, each with a different internal coordinate origin and a different orientation from the objective (90° and 210°).

an objective domain in our implementation, and the internal world maps of two robots. Each has a different starting point (cartesian plane origin) and begins at a different orientation, and the same world is represented very differently in each.

What is needed is a mechanism to develop shared groundings and reconcile these into a *consistent* set across the population, in spite of individual differences between agents. The bounded resources at each agent's disposal preclude simply sharing each and every grounding among all agents. Communication to agree on consistent symbol grounding in any complex domain would be combinatorially more significant than the amount of communication necessary for problem-solving itself, and in domains where broadcast communication is sporadic, would be physically impossible. Broadcasting all groundings would also require some means of dealing with the differences in perspective between agents illustrated in Fig. 14.2 as well.

However, building up common-sense shared references to locations (and other entities) over time is a much more human-like approach to the problem, and one that serves as a more generally intelligent solution. A team of robots using such an approach would be able to function anywhere, and would be able to be developed independently of one another and form teams in an *ad hoc* manner. The ability to form teams without a great deal of prior knowledge assumed between individuals is an important element in developing robot teams broadly adaptive to many applications.

Beyond these issues, another motivation for allowing the development of shared groundings by a population of agents is avoiding developer bias. The hand-constructed groundings that have been used in most systems thus far reflect *anthropocentric* (human-centered) categorizations, which can limit the resulting system in ways that would not occur if those categorizations were made by the system itself [15]. In designing a behaviour-based system [2], for example, designers of place linguistic labels on the robot's internal behaviours, such as *follow-wall*. These have meaning to the system designer and affect the way the designer thinks about the system. However, the agent performing the behaviours carries none of the label semantics used by the designer: internally the complex interactions between a collection of such behaviours may be operating in a very different manner than might be assumed given their labels. This leads to misunderstandings and errors on the part of developers who use these labels to understand the system. A rift forms between what the designer intended and what the system actually does [15].

Similar issues exist with symbol groundings. Choosing the entities worth grounding may be obvious when one is designing a small system. As systems become larger and more complex, however, not only does selecting all groundings become impossible, but the designer-system rift comes into play. The groundings chosen by the designer are based on his or her own knowledge of only the isolated elements of that system, and of how the agents are *expected* to perform. Were agents to evolve their own set of groundings, those might be very different from the groundings envisioned by the human designers. They would arguably be more effective as well, since unneeded groundings that designers might supply would not develop, and groundings would be tailored to the activity at hand [15].

This chapter describes the an approach by which an ad hoc team of robots may explore an environment and develop a consistent set of shared spatial groundings for the purpose of useful communication. A number of strategies for creating groundings are presented, and the development of a consistent set of groundings across a team is achieved using the local interactions that occur in the course of performing useful work, without sharing a coordinate system.

A location grounding, for our purposes, is a symbol that is associated with an agent's internal spatial coordinates, allowing the agent to make reference to a location in the environment (and ultimately be able to go there or refer to it in reasoning or communication). A shared grounding is one where the same symbol refers to the same absolute location in the world, even though the agents sharing the grounding may have completely different representations of that location internally. The approach presented here begins with no prior knowledge of the environment or shared points of reference, requires no explicit location labelling phase, requires only local encounters between agents rather than broadcast communication, does not flood the environment with groundings, and is demonstrated to be functional with larger populations (up to 16 robots) than previous work (e.g. [15], which uses only 2 robots). This approach is then evaluated in a domain where communication about locations in the environment can improve performance, and is shown to provide significant benefits, indicating that agents using this approach can reap the benefits of grounded communication without predefined groundings. The evaluation compares the alternatives employed under varied environmental conditions and team sizes.

There are a number of ways in which references can be made to entities in the environment in a grounded fashion [9]. An *iconic reference* is an association based on observable features (e.g. seeing a fire and associating that sensor data with that concept). An *indexical reference* associates two icons (e.g.smoke may be an index for fire), while the most powerful is a *symbolic reference*, which is an arbitrary relationship between icons, indices, and/or other symbols. While associating sensory phenomena to internal representations in an iconic manner is a relatively straightforward form of learning, the other two types require much more sophistication. The approach described here demonstrates the use of both indexical and symbolic references.

The remainder of this chapter overviews related work in this area, describes our approach and its implementation, and then describes the results of an empirical evaluation.

14.2 Related Work

Much practical reasoning employs symbols as placeholders for objects in the physical world. The groundings between perceivable objects and the symbols used to refer to them are commonly referred to as *anchors*, and the problem of developing and maintaining anchors for individual mobile agents through perception (a subset of the broader symbol grounding problem) is known as the the *anchoring problem*. Coradeschi and Saffiotti presented a preliminary formal logic solution to the anchoring problem in 2000 [6], defining the major components in predicate logic. They later extended this work to use symbols in actions and plans [7]. While this work provides a formal framework for maintaining anchors to symbols via perception in single agent systems, it does not deal with the multitude of practical issues that arrive when the shared groundings necessary for communication in multi-agent systems are considered, since the primary problems in group are not only deciding whether something is worth grounding, but reconciling the multitude of inconsistencies among the groundings of various agents.

Comparatively little work in grounding has been performed in multi-agent systems, because of the problems of developing and maintaining consistency of groundings among agents. Steels has done work in evolutionary linguistics, examining how individual agents in a group may be able to generate discrimination trees to distinguish one object from other in the environment, without the aid of a teacher [17]. Vogt [18] later improved on this work. These works are more limited than the work we present here, in that they operate with only a very limited number of objects in the environment, and the domain itself exists only to perform groundings as opposed to having groundings acquired over the course of activity. However, these do show that it is possible to allow agents to develop shared symbol associations in coherent ways.

Billard and Dautenhahn [5] studied the benefits of social skills to learning in heterogeneous multi-agent systems. In their environment, robotic agents were provided with a following behaviour, allowing them to keep a similar sensory context to another robot by maintaining physical proximity, and learned to associate words with color descriptions of flooring after receiving demonstrations by following a teacher broadcasting the correct color name. While agents could eventually become teachers, the language used never changed: groundings were never invented, merely spread from an individual teacher through the population. In the approach described in this chapter, all agents are both teachers and learners all of the time, since each is developing useful groundings in the course of its own work, as well as trying to spread those groundings to others in order to be able to communicate about those spatial locations. This is thus suited to unknown environments where an existing competent teacher cannot be assumed.

Jung and Zelinsky [15] describe the implementation of a heterogeneous cooperative robotic cleaning task which benefits from the use of a symbolic references for communication. Two robots were required to clean a laboratory floor: *Flo* sweeps litter into piles, while *Joh* vacuums the piles to remove them. Their system adds three layers of behaviours for coordination between agents, each of which improves performance as it is added. First, Joh is given the ability to track Flo visually (useful here since Flo will likely be near a pile of litter). Next, Flo is given the ability to signal the location of a new litter pile to Joh, in coordinates relative to itself. Joh and Flo use the same wheel encoders, providing a pre-defined shared grounding for the communicated coordinates. Finally, communication using predefined symbolic references is employed. Each reference is communicated in terms of the angle and distance from a reference line connecting two points in the robots' environment, thus allowing a new point to be referred to using two points that are already known. This requires that robots go through an explicit location labelling phase in order to develop a common set of reference points so that communication can be understood. While this work shows that a human-centered coordinate system is unnecessary for navigation and communicating about locations, and that symbolic references are very useful for coordinated activity that takes place across a spatial area, the approach employed is not practical for problems of any significant size. Labelling locations is done by arbitrarily labelling enormous numbers of points (in excess of 500 for a 3.4 x. 5.1 m room) during a phase in which both robots devote their full effort to collectively labelling locations (i.e. there is never a situation where one robot grounds a location independent of the other and then shares that grounding later). The number of points grounded is impractical for any everyday activity, but more importantly in a multi-robot setting, is impractical to make consistent across any significant population. This is unrealistic compared to most situations, where individuals are likely to take note of points that are interesting to them during the course of useful work, and share these with others when it is convenient, as is done in the scheme we present in the remainder of this chapter.

14.3 Developing Grounded Spatial References

Any scheme for developing grounded spatial references among a team of agents requires three distinct policies: deciding when a particular location is worth grounding by an agent (i.e. allocating some label and maintaining an association for the agent's own activities); deciding when such a grounding is worth sharing with others; and deciding how to reconcile groundings when they conflict with those shared by others. We present these in the separate subsections that follow, with specific implementation details provided in Section 14.4. In our implementation, labels are random integers and individual coordinate systems are cartesian planes, but these techniques place no restrictions on the representation used for labels or coordinate systems, and no shared representation for coordinates is assumed.

14.3.1 Grounding Locations

Whether any given location in an environment is worth maintaining a correspondence with a label is largely dependent on the task a robot is performing. If one is cleaning a room, one will consider different locations to be of significance than if one is painting it. During these and many other tasks involving mobility, however, a robot is navigating the room, and so we attempted to develop several approaches that will suit this task and thus indirectly assist in others. The general operation of each scheme is described briefly below, and implementation details of each are found in Section 14.4.1. The relative utility of these schemes in practice is examined in Section 14.5.

The first of the strategies we employ, *label-at-meeting*, creates a shared grounding between a *pair* of agents at the current location when they encounter one another in the environment. This is a basic strategy that takes advantage of chance encounters that will inevitably occur in multi-robot domains. The dynamics of this strategy result in more groundings in areas where robots tend to spend most of their time, since encounters are more common in these areas. When a robot R_1 encounters another robot R_2, it stops and invites R_2 to ground a symbol. If R_2 is amenable (since it may be busy with other things), it requests a label for R_1's current location. R_1 names the location and the new label is subsequently used by both. A delay is built into the implementation (Section 14.4) to stop this from occurring too frequently so that agents are not taken away from their primary task in this environment.

While label-at-meeting is domain independent, it is also limiting in that an agent is restricted to creating a grounding only when another agent is present. The second strategy we developed, *label-spatial-entropy*, is more general in that it tries to predict, in a domain-independent way, which locations in the environment are likely to be useful. To make this prediction, we note the previous use of information theory in categorizing space for path planning. Baltes and Anderson [4] define the entropy of a spatial area (referred to as *spatial entropy* in this work), a metric that uses information theory to measure how mixed a given portion of the environment is in terms of free and occupied space. Spatial entropy is highest where there is the largest mixture of open and blocked areas (i.e. where the environment contains the most information). Here we employ this measure to choose useful locations for grounding: when an entropy threshold is exceeded, a grounding is generated (Section 14.4).

To contrast these general techniques and examine the effect of providing more specific information to agents ahead of time, we also define the *label-environment-feature* technique, where specific phenomena useful to the task at hand are explicitly described to the agent and grounded when these are recognized. This is an anthropocentric attempt to capture useful domain-specific properties, such as doorways and hallways. It also serves as an upper bound for the quality of groundings that can be made in a given domain.

14.3.2 Sharing Groundings

The label-at-meeting strategy is the only one of those above that contains some means of sharing a grounding among agents, and this is limited in that only the two agreeing to make the grounding possess it. To share groundings among agents generally, some means beyond mere communication must be provided. Since each agent may employ its own internal coordinate system, communicating these coordinates would be meaningless to others. Our approach employs the physical demonstration of groundings, to allow other agents to ground a location by perceiving it themselves.

Demonstrations are initiated when agents encounter one another. If a robot R_1 perceives another robot R_2 within a specific distance, and a known grounding is also within a given distance, R_1 will offer to demonstrate it to R_2. If the second agent accepts the offer, it will signal R_1 and begin to follow that agent as it moves to the demonstrated location. When R_1 arrives at the location of the grounding, it will then send a message to R_2 indicating the label it employs to designate the location. If this location is novel to the R_2, that agent will also ground the location to that label, and will signal R_1 that is has done so. This offer-and-acknowledge approach allows either agent to avoid participating in the demonstration if other tasks take priority, and in turn support balancing the importance of the overall task of the team with the benefits that another shared grounding might bring. Further details of our implementation of this strategy may be found in Section 14.4.2.

14.3.3 Consistency among a Group of Agents

An agent may learn many groundings by demonstration, but since each agent is also learning from others and making groundings itself, it will encounter many inconsistencies as well. The most important part of creating a set of groundings among a team of agents is mediating these conflicts appropriately as they are discovered, in a decentralized manner in which the computation is entirely local. Some mechanism is needed so that when groundings are shared, any conflict in labels or locations will be more likely to be resolved toward greater consistency among team members as a whole, rather than away from it.

In our approach, this is done by maintaining a reference count for each grounding within each agent. This reference count indicates, to the best of that agent's knowledge, the number of agents in total that employ this same grounding. This is a heuristic value, because it is maintained based only on local encounters rather than global communication. Many of the agents with which it has been shared will have since gone on to demonstrate the grounding to others, without the originator of the grounding being able to know this. Others which which the grounding has been shared may have since had a conflicting grounding replace this information. When performing a demonstration as described above, the demonstrating agent (R_1) will send its reference count to the encountered agent (R_2) in addition to communicating the symbol used, and will update this count by one if R_2 accepts its demonstrated grounding. In the case of a conflict during a demonstration, reference counts of the symbols involved allow a local determination of which is likely to be more widespread, and thus a means of encouraging less widely used groundings to be discouraged and those more widely used to be propagated.

In practice, there are three types of conflicts to be resolved. This section overviews these at a high level, and leaves the implementation details to Section 14.4.2. The most obvious (Case 1) is that R_2 may know the location grounding already, but under a different symbol, either because it has learned it from a third party or developed it itself. In this case, the agent with the lowest reference count gives up its grounding and adopts that with the higher reference count, and each updates the count accordingly. As this reference count becomes higher, it will eventually propagate to other agents and replace competing labels for the same groundings across the population. In the case of equal reference counts, the grounding of R_2 will be used in both agents: this leaves the two agents consistent in terms of this grounding after an encounter, and serves to add some change to the population and eventually cause one grounding to dominate the other.

The second type of conflict (Case 2) occurs when the R_2 recognizes the symbol communicated, but already uses that symbol for a different location grounding. Here, consistency demands that one of these groundings must ultimately be abandoned, and the reference count is used to preserve the more prevalent of the two. A grounding is lost to one agent, but that grounding is less valuable to the group of agents as a whole than the one that is preserved. If the lost grounding is something of use, it will be rediscovered or relearned from others. It may encountered again with the same symbol, but eventually one of the two competing uses will prevail across the population due to reference counts. The alternative to this–keeping the grounding and choosing yet another new label–is not followed because while it would preserve the grounding in one agent, it would cause even more groundings with redundant labels to ultimately spread across the population, further reducing consistency.

In the final form of conflict (Case 3), both the location and the label are known to the R_2. This may in fact be the same grounding, but recorded slightly differently due to errors in perception or odometry. The implementation described in the next section details the exact methods by which agents judge two locations to be the same. If these locations are not the same, then the viewing agent has both a different symbol grounded to the current location, and the symbol used by the demonstrator in use for another grounding. This amounts to both the two prior cases, and these are resolved as described above.

While conflicts involving different agents independently choosing the same label for different groundings can be somewhat mitigated by using a broad a range of labels to limit the likelihood of duplication, they cannot be guaranteed to be avoided through such means. There will always be some likelihood of duplication, and beyond this, it is possible that errors in demonstrations (e.g. through incorrect localization of the demonstrator or viewer) would lead to the spread of the same symbol with multiple locations, and any scheme to promote consistency must deal with these. The approach presented here is also general in that it allows agents to operate more independently, by not forcing any particular scheme for choosing labels (or even a consistent scheme between agents). In our evaluation (Section 14.5), labels are generated randomly in individuals, allowing such duplication. To the degree that duplication can be lessened, of course, the results shown can be improved.

14.4 Implementation

Since this approach is designed to operate as a group of robots performs some other duty, it could be implemented in any environment. To keep this general, we chose a basic multi-robot exploration domain, and implemented this approach in a behaviour-based robotic control system [2] for Pioneer DX2 robots, using Java 1.4, under Player/Stage [12]. Simulation was chosen to evaluate our implementation because it allows the support of a larger team than would be possible with the physical robots available in our lab. As others have previously noted (e.g. [13, 11]), simulation also lets us run trials much faster and with a much greater level of control and repeatability than the physical world. In addition, Player/Stage has been validated as accurately simulating the behaviour of ActivMedia's Pioneer robots, and code developed under Player/Stage will run under Pioneer robots directly.

The mapping performed by our software employs only the sonar that is standard with the Pioneer DX2, in order to ensure that the approach works with low-resolution mapping and would thus be applicable to much simpler robots as well. A laser rangefinder was added to the robot, but was only used to exploit the ability of this sensor to identify particular markers in the environment. Each robotic agent is given a unique numeric marker detectable by any agent's laser range finder hardware, and this thus serves to uniquely identify each robot. This is not necessary for the approach itself, but is required by this implementation because the simulation software does not provide any direct agent-to-agent messaging options.

In our simulated domain, agents map the world individually as they explore it, and can plot paths to specific locations based on their map. The task environments were 11x11m and 8x8m arenas, containing walls, doors, and obstacles (e.g. Fig. 14.1).

Grounded communication in this domain is employed in the form of goal communication. A goal is placed randomly in the environment for the agents to find. As each agent explores the environment, it creates its own groundings using the methodologies described in Section 14.3, and shares these as other agents are encountered during the course of exploration. When an agent finds the goal, its location is broadcast to all agents in a grounded manner (this is the only broadcast communication used, and is used as a means of evaluating the efficacy of groundings in this domain). The subset of agents with the shared grounding(s) necessary for the goal location to have meaning are then able to plan a path and navigate to the goal. As greater consistency develops among a team of agents over time, all agents in the group should be able to arrive at the goal location more and more quickly.

Two different techniques are used to describe the goal location. The first is simply broadcasting the label for the grounding the discovering agent has nearest to the goal (an indexical reference). This location will not necessarily be at the goal itself, but it is the closest location to it that can be understood by others directly.

A second technique implemented was to specify the goal location using two grounded locations and a more complex symbolic reference. The symbolic reference is specified as (p_1, p_2, a, d). The indicated position is found by drawing a reference line from p_1 to p_2 and then, starting at p_2, traveling d times the distance from p_1 to p_2, at angle $a \times 360^o$ to the reference line. This is similar to the symbolic reference used in [15]. Understanding these two grounded locations allows the basis of a shared coordinate system between communicator and receiver, and allows an agent to specify a goal location precisely even if it has no grounding near the actual goal. This flexibility comes at the cost of requiring two shared groundings instead of one.

Robot control is behaviour-based, and robots are given basic navigation and exploration behaviours (e.g. wandering, obstacle avoidance) and communication behaviours. Higher level behaviours support mapping and path planning. Mapping uses an occupancy grid approach [10], with each robot maintaining a cartesian coordinate system based on its occupancy grid, with the $0,0$ origin marking its own random starting location, and a 0 heading marking its own random orientation. This results in each agent having its own private coordinate system, offset and rotated from the real world (i.e. the Stage coordinate system). An agent has no knowledge of the relationship between its own coordinate system and the coordinate system of others: it is thus not able to communicate meaningful spatial information by using raw coordinates and headings, and must use labels for commonly grounded locations. Agents also make no assumptions about the size of the environment, thus requiring internal mapping to dynamically resize its world map as new areas are discovered. In our implementation, we begin with 20 grid cells (of 10cm x 10cm each, the atomic unit size we employ for mapping) in each direction from the robot's origin, and increase this by 5% along the appropriate axis as new areas of the environment are discovered, and cells are marked as blocked or free with a likelihood based on repeated sonar reflections or lack thereof.

Path planning follows a quad-tree-based approach, with path improvements based on the technique of [8]. This simplification is based on the realization that some nodes along a path planned through a quad-tree decomposition are unnecessary. For example, if a planned path goes from node *a* to node *b* to node *c*, it may be possible to remove the middle node, node *b*. This is possible when there is an obstacle-free straight line path from node *a* to node *c*, then node *b* is unnecessary and may be eliminated. This simplification results in shorter paths.

14.4.1 Location Grounding Behaviours

Behaviours are implemented to encompass the three grounding generation strategies described in the previous section. Only one of these grounding strategies will be active at a time. All grounding strategies label new groundings with a randomly generated integer in the range 0 to 10000. When an agent generates a new label, if a randomly chosen number is already in use to ground a location within the agent's knowledge base, a new random label is selected.

The label-at-meeting strategy involves an interaction between two robots, and a series of communications as a result of that interaction. These are implemented in a behaviour whose operation is summarized in Figure 14.3.

When a robot R_1 senses another robot R_2 within 4m, R_1 stops in place, and sends a message (LABEL_MEETING_START, an invitation to begin a grounding creation) to R_2. If R_2 is receptive, it begins the process of creating an acceptable label as described below. If R_2 is not receptive, it responds negatively to R_1 and both agents continue to explore the environment, performing useful work (in our implementation, searching for a goal). R_1 will not try to initiate another label-at-meeting conversation for 20 seconds after receiving a negative response.

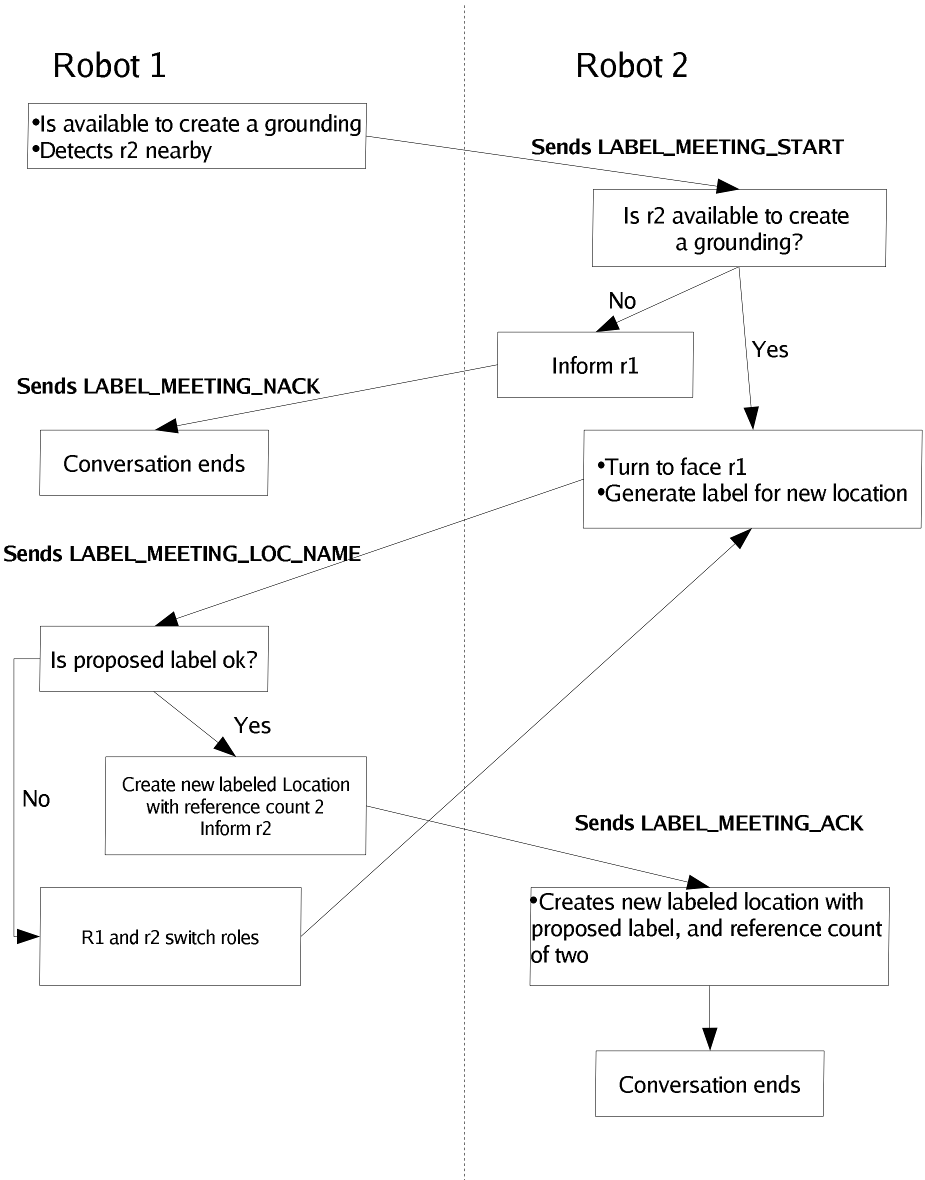

Fig. 14.3. Label-at-meeting agent interaction

If R_2 is receptive, it indicates this receptiveness by responding with a proposed label l_p. This proposed label will ground *the location halfway between them*, and is sent to R_1 (as an argument of a LABEL_MEETING_LOC_NAME message). R_1 then evaluates the proposed location label, and if it is not already used in R_1, then l_p is accepted. R_1 signals this to R_2 with an acknowledgement, and both agents then know the location halfway between them as l_p. l_p starts with a reference count of two, since there are two agents which know the location as l_p. In the event that R_1 already has a grounded location by the label of l_p, the role of choosing a grounding label is reversed, and R_1 sends a message to R_2 with a different proposed label for the location. R_2 processes this message in the same manner that R_1 did. This process goes back and forth until a novel label in both R_1 and R_2 is found, or the conversation times out after 15 seconds.

Recall that the label-at-meeting strategy is the only one where the location of a grounding is decided upon through the interaction of two agents. The other strategies require some judgement as to whether it is useful to ground the location a single agent currently occupies. The behaviour implementing the label-spatial-entropy strategy performs this judgement by considering the fifteen occupancy grid cells that surround the agent's current location every ten seconds, calculating the spatial entropy of these cells as the robot moves. A grounding is made if the spatial entropy threshold exceeds the standard value of 0.75 (a value arrived at by preliminary experimentation in the domains employed). If the agent is near the edge of the environment, the box of adjacent occupancy grid cells would extend outside the environment, and it is moved to be entirely within the environment.

Implementing the label-environment-feature strategy in a behaviour requires recognizing locations that are useful in a domain-specific manner, which can be arbitrarily complex. Because the focus of this work is not on robotic perception, in our implementation we employed feature labelling by placing perceivable feature markers that could be easily recognized. Each marker was given a unique identifier in the same way that agents are uniquely identified – these thus allow the unique identification of environmental features when they are discovered. An agent using this strategy creates a grounding when it perceives a marker within 2m. While the use of uniquely perceivable identifiers could easily allow these to serve as artificial aids to localization compared to more generic phenomena such as doorways and hallways perceived by sonar, at no time was this done in our implementation.

In all labelling strategies, we avoid flooding the environment with groundings by removing any groundings with a reference count of 1 within 1.5m whenever a new grounding is produced. Redundant groundings serve no purpose and since they have not been shared their removal has no impact on the rest of the population.

14.4.2 Sharing Groundings

When sharing groundings, a fundamental question that arises is whether a location being demonstrated or referred to by one agent is identical to that used by another. Due to errors in both perception and localization, determining the spatial equality of two locations is more complex than simply comparing numeric coordinates for equality, even if disparate coordinate systems could be reconciled. For the purpose of implementing the approach described in Section 14.3, a tolerance factor ε is employed. If two locations are physically within ε range (50 cm in this work, a value arrived at through preliminary experimentation in the domains employed), they are considered the same.

Figure 14.4 illustrates the detailed interaction between two agents attempting to develop a shared grounding, in a behaviour known as the *location exchanger*. Note that this behaviour is the same irrespective of the method used to originally create the grounding being shared. When an agent R_1 detects another agent R_2 nearby (by default 5m), and has grounded a nearby location (within 2.5m of its current position), R_1 may attempt to share a grounding with R_2. The interaction is initiated by R_1 sending a DEMO_LOC message to R_2. If R_2 is available it turns to face R_1 and replies with a WILL_FOLLOW message, indicating that it will follow R_1 to the location to be demonstrated. R_2 may not be available because it is already creating a shared grounding with another agent or has just completed doing so (our implementation requires an agent to wait 30 seconds between sharing groundings).

If available, R_2 begins to track R_1. Once R_1 has arrived at the location to be demonstrated ($demo_{coordinates}$ in the objective world, but recorded as ($R_{1location}$ in R_1's own coordinate system), it stops and sends a THIS_IS_LOC message to R_2. This message contains R_1's label (R_{1name}) and reference count for this location. In addition to the perceivable location, and the data communicated by R_1, R_2 also may have a grounding R_{2name} using the same label (which may be at another location, $R_{2labelled\_location}$ in its own coordinate system).

To determine R_2's response to this demonstration, R_2 needs to determine whether or not it has a grounding fitting $R_{2labelled\_location}$ (i.e. knows a different grounding within ε tolerance), and whether or not it has a grounding fitting R_{2name} (i.e. already uses the label R_{1name} elsewhere). The cases that result from these two decisions have already been explained in Section 14.3.3. Here, we explain what occurs in each case from an implementation perspective.

If R_2 knows neither $demo_{coordinates}$ nor $demo_{name}$, there is no conflict. R_2 adopts $demo_{name}$ for $demo_{coordinates}$, and replies to R_1 with a LOC_LABELLED message. Both R_1 and R_2 now know R_1's current location as $demo_{name}$. Each agent also increments the reference count to reflect that one additional agent knows this grounding.

There are also three possible types of conflicts, as described in Section 14.3.3. In case 1, R_2 knows $demo_{coordinates}$, but not $demo_{name}$, and must determine whether to use its own label for this location, or R_1's. If the label that R_2 is currently using has a higher reference count of the two, this is the more consistent label and R_2 tells R_1 its label and reference count for R_1's current location via a LOC_OVERRIDE message. If R_1's label has the higher reference count, R_2 adopts the supplied label for the supplied coordinates, and acknowledges this via a LOC_LABELLED message, allowing both to update reference counts.

In case 2, R_2 knows $demo_{name}$, but not $demo_{coordinates}$ and R_2 must determine whether the location currently grounded by $demo_{name}$ is more useful than this new proposed grounding. If R_2's label has a higher reference count than R_1's, R_2's location is more valuable. However, since the location is an arbitrary one in the environment, and would require travel to demonstrate, the best option is for R_1 to simply forget about $demo_{name}$. To effect this, R_2 sends a LOC_FORGET message to R_1. If R_1's label has a higher reference count, R_2 forgets its current information about $demo_{name}$ and grounds $demo_{name}$ to $demo_{coordinates}$. R_2 then sends a LOC_LABELLED message to R_1, allowing both to update reference counts.

In case 3, R_2 knows both $demo_{name}$ and $demo_{coordinates}$. In this case R_2 must first determine if it employs both of these in the same grounding. If so, R_1 and R_2 already share a grounding for this location, and the reference counts must be made consistent.

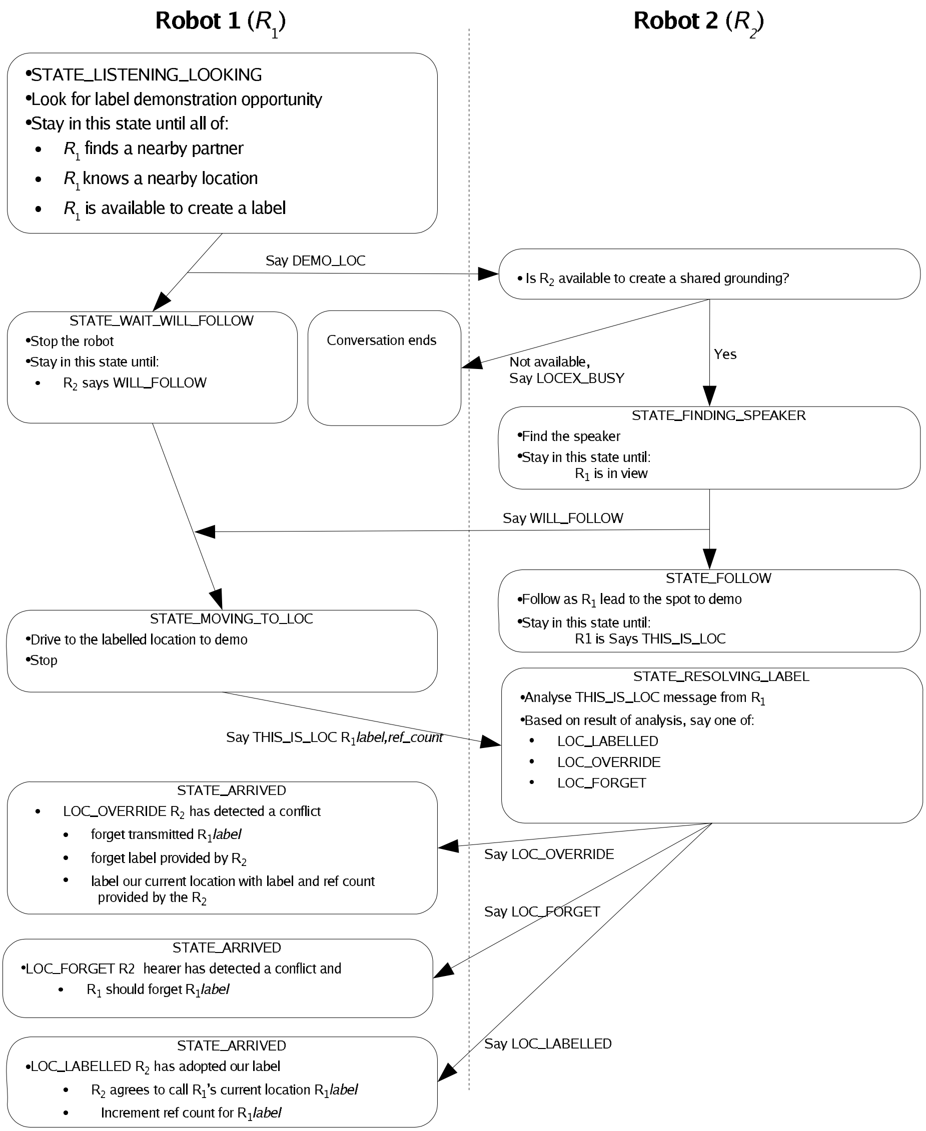

Fig. 14.4. Agent interaction during development of a shared grounding

R_2 corrects the reference count in R_1 to the higher of the two using a LOC_OVERRIDE message (LOC_LABELLED is used only to mutually increment the reference counts).

If R_1 and R_2 do not share the same grounding for $demo_{coordinates}$, R_2 must determine which location should be kept based on the reference counts. If the reference count provided by R_1 is higher than that for R_2's grounding of the label $demo_{name}$, R_2 adopts R_1's label and reference count. R_2 then sends a LOC_LABELLED message to R_1, allowing both to update reference counts. If R_2 has a higher higher reference count than R_1, R_1 should forget its grounding for $demo_{name}$, so R_2 sends a LOC_FORGET message to cause this to occur.

The behaviours described in this section summarize the core of our implementation. These encompass a number of lower-level behaviours (e.g. a behaviour to move to a nearby grounding to demonstrate it, follow another robot, etc.). Additional behaviours are also required that are not described fully here for reasons of brevity and the fact that these behaviours are reasonably obvious. For example, a goal seeking behaviour is implemented that attempts to perceive if the robot has reached a given goal, and listens to goal locations broadcast in terms of grounding labels by other robots. Further details on all of these behaviours are available in [19].

14.5 Evaluation

The evaluation of this approach was done in simulation using Player/Stage [12], using six Linux-based PCs organized in three pairs. The first computer in each pair was used to run the Stage simulator, while the second was used to run all the of the simulated robots.

The evaluation performed involved examining the efficacy of the grounding schemes described in Section 14.3, in the domain described in Section 14.4. Agents are place in the domain with random locations, and headings, and begin with no groundings. Agents then construct, demonstrate, and resolve conflicts between groundings, using the methods described previously, as they explore the environment. Whenever the goal is found by an agent, the location is broadcast using an indexical or symbolic reference as described in Section 14.4. This allows the goal more quickly by any others that share the groundings necessary to understand the reference. If an agent has not found the goal after 10 minutes of searching, it gives up. After all agents have found the goal or given up, one iteration of a trial is complete. During this iteration, agents collectively created a set of groundings which should be useful to improve their performance in this environment in future. These groundings should become more extensive and more consistent as agents gather more experience in this environment. In order to examine this effect, a full experiment consists of a series of 200 of these iterations, where agents maintain their groundings between iterations and are placed in new random locations (with a new, randomly located goal) for each iteration.

Experimental trials were conducted in many different configurations. The three grounding creation strategies were explored independently, as were the two goal communication mechanisms. The size of team was varied (2, 4, 8, and 16 robots), as was the size of the domain (8x8m and 11x11m). We also attempted to vary the configuration of the space, since domains of a given size can have very different complexity, by defining four different layouts: open space for a baseline, a general partitioned domain meant to be typical of an office (Fig. 14.1), a *split* domain (Fig. 14.5) where barriers more strongly restrict travel from one end to the other, and a *hallway* domain with areas partitioned off either side. The distance at which a demonstration would be performed and the ε constant were 5m and 50cm respectively (additional experiments also examined varying these [19]).

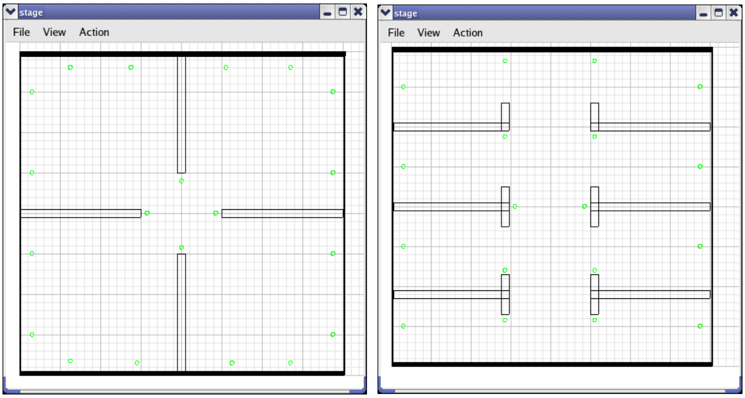

Fig. 14.5. The split (left) and hallway (right) 8x8m environments

These experiments all tracked group performance over trials as shared groundings were developed, and also examined both the final density of groundings produced and the percentage of consistency in groundings among agents that was ultimately achieved. The latter was computed by gathering all individual groundings after an experiment into a full set, removing duplicate labels for the same locations (where two robot widths - 80cm - was used as threshold for equality) and keeping only the most commonly used as the set of globally ground locations. The percentage in each agent that was consistent with this global grounding set was tallied, and these were averaged across all agents to produce an average consistency. This metric is a pessimistic one, in that by throwing multiple values away, we are discounting the partial consistency that does exist in the removed values (e.g. 49% of agents may have a consistent label for a location that is still useful to this subset, but this is removed in favour of the 51% that have a different label). This measure is also biased against larger populations, because unique individual groundings affect the measure more in larger populations. For example, if there is only a single globally ground point, and one agent that possesses that grounding, this amounts to a 50% global consistency in a 2-robot team, but only a 6.25% consistency in a 16-robot team.

In all cases, results are presented by comparing the team performance against a control group consisting of the same configuration but without the ability to develop and communicate with shared groundings, and expressed as a percentage of improvement over this baseline. Here we present a representative subset of the results obtained, focusing only the larger environment size, and the reader is referred to [19] for further results of interest.

Over all domain types, in the larger domain size, the average improvement using a goal communicated as a single grounding (an indexical reference) is shown in Fig. 14.6. This showed good improvement as team sizes increased up to 8 robots, followed by similar performance when doubling the population to 16 robots. In an 8x8m domain, the performance dropped after increasing the team size from 8 to 16 robots. In both domain sizes this is showing the point at which overcrowding and the resulting interference

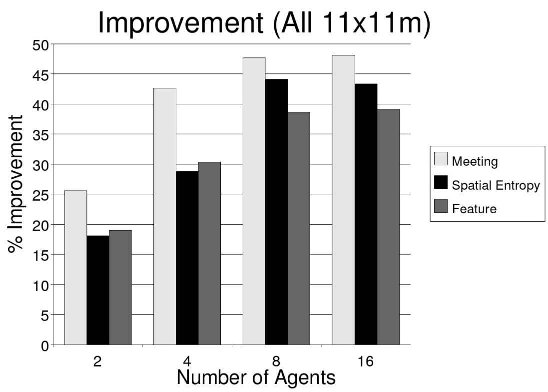

Fig. 14.6. Average percentage improvement by grounding strategy in 11x11m environments

between agents (e.g. collisions, more time spent avoiding obstacles) tended to limit the functionality of a larger team. Nonetheless, the techniques described here produced observable performance gain.

The label-at-meeting strategy had the largest performance improvement. This is attributed to already having a shared grounding between two agents when the grounding is made, leading to an immediate increase in the ability to spread that grounding quickly. This is especially significant for 2-agent teams, since a shared grounding is already globally consistent. Looking at consistency (Fig. 14.7) in these environments illustrates this. In this task, meetings were naturally distributed across the environment, leading label-at-meeting to produce wide coverage in groundings locations that led to a greater likelihood that a goal could be usefully described to most agents. In situations where meetings were infrequent (e.g. a small team over a large domain), this would be less useful, but in such a situation there is also a much stronger likelihood of agents working independently in different areas, and arguably less of a need for grounded spatial communication than in closer confines.

Label-spatial-entropy was generally the second most successful approach. This was despite the fact that it produced significantly more groundings than the other two strategies, especially with large numbers of agents (Fig. 14.8). This increase is still not large, and the number of points grounded by 16 robots is far fewer than the approach of [15] makes for 2 robots. For 2 and 4 agent populations, agents are able to exchange enough locations to maintain a reasonable level of consistency. With larger population sizes, the number of groundings exceeds the number that can be reasonably shared, leading to lower consistency. This does not rule out spatial entropy as an valuable alternative: a domain could be adjusted to use a higher entropy threshold and still use this reasonably domain-independent technique.

It may seem surprising that grounding random encounters performed better than a technique that was specifically designed to ensure useful areas for navigation were grounded. However, not only is there greater sharing built into the label-at-meeting strategy, the points being grounded by label-spatial-entropy are not being exploited in this domain as strongly as they might be. Despite these groundings being interesting

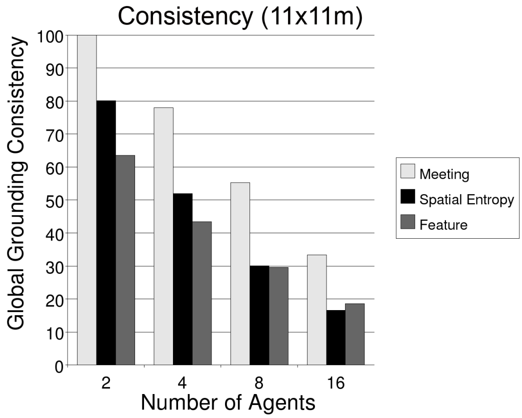

Fig. 14.7. Average global grounding consistency by grounding strategy (indexical reference) averaged over all 11x11m environments

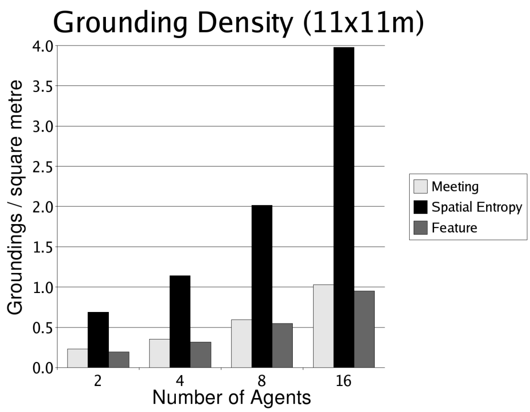

Fig. 14.8. Grounding density by grounding strategy (indexical reference) averaged over all 11x11m environments

from the standpoint of navigation, the grounded communication employed references only the goal location. In that respect, these points are no better than other random locations. We anticipate that if an environment were to be set up where the grounded communication was more navigationally-oriented, spatial entropy would be more successful. An example of this would be an environment where information about waypoints to a goal were communicated instead of the goal itself.

Labelling environment features produced the lowest overall improvement, mainly because there were too few locations to be grounded (as indicated by grounding density). Having few locations to ground seems like a factor that should encourage high consistency. However, it also limits the opportunities for an agent to demonstrate an

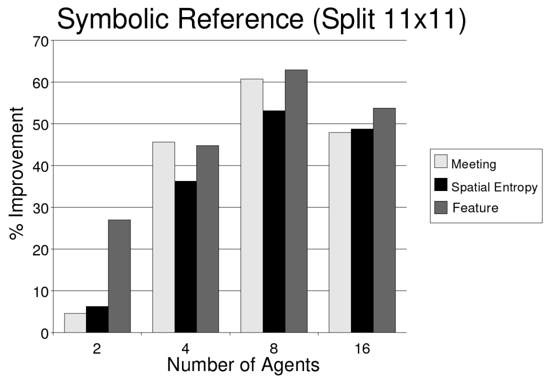

Fig. 14.9. Average improvement by grounding strategy (symbolic reference) in the split 11x11m environment

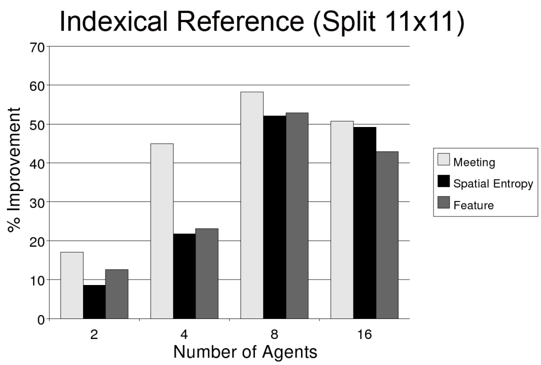

Fig. 14.10. Average improvement by grounding strategy (indexical reference) in the split 11x11m environment only

already-grounded location to others, since these locations are only demonstrated when agents encounter one another with a grounding nearby.

Because of the large numbers of trials demanded by the breakdown of all the controlled variables being evaluated, and the time taken to run trials (One trial of 200 iterations with 16 agents in one environment, for example, took six days using 2 computers), we examined some factors only in subsets of these environments. In particular, the effect of using symbolic references involving two groundings rather than using a single grounding as an indexical reference, was examined only in the split environment configuration. This was selected because it showed a high average improvement in the baseline experiments, and it also had high absolute times, and so we should expect a more sophisticated technique to be just as well-received. Results with goals communicated to team members symbolically using two grounded locations as described in Section 14.4 in an 11x11m split configuration are illustrated in Fig. 14.9. Because this affects only how goals are communicated, density and consistency information should not differ the results presented earlier. To provide a baseline for comparison, the results shown in Fig. 14.6, disaggregated to show only the split domain, are shown in Fig. 14.10.

From these Figures, it can be seen that the label-at-meeting grounding strategy using an indexical reference to communicate a goal performs as well or better than using a symbolic reference. The dynamics of the strategy provides good distribution of shared groundings over the environment, with high consistency, allowing the goal's location to be specified accurately enough with the single reference. This is coupled with the fact that a symbolic reference requires twice as many shared groundings for the reference to be useful, leading to approximately equal performance between the two references.

The label-spatial-entropy grounding strategy shows little difference in performance across all agent populations, using either form of goal reference. The increased flexibility of a symbolic reference seems to be mitigated by the need to share twice the number of groundings in order to make use of it. Future work could explore this by selecting a domain in which a simple reference is less likely to be useful.

The label-environment-feature grounding strategy fared better than all others for an 11x11m split environment when using symbolic references. This illustrates the advantage of using symbolic references over indexical references when sufficient groundings are available to support them. Since agents using the label-environment-feature grounding strategy are limited to grounding certain locations, a single grounded reference (used when indexical references are communicated) has limited ability to specify the goal's location. This is in contrast to a more complex symbolic reference, which gives label-environment-feature the ability to specify an arbitrary point in the environment, and hence the goal's location, more accurately.

14.6 Conclusions

In this chapter, we have presented an approach to developing consistent shared groundings over time that uses local interaction between agents to bring consistency to a population. We have described a number of alternatives for creating groundings, detailed the implementation of these techniques, and presented an evaluation comparing them. In comparing techniques for grounding, label-at-meeting was the most successful, because the implicit sharing when agents create a new grounded location leads to a high global location consistency, which improves the effectiveness of communication about the goal. This approach allows agents to use their own independent spatial representations and coordinate systems internally, and develop the means to function together as a team without assuming a shared coordinate system or external facilities such as GPS. This in turn leads to more general agents. It also allows groundings to be developed that are naturally useful from the point of view of the agents themselves, rather than the viewpoint of the agent developers. Such non-anthropocentric mechanisms are an important part of developing truly autonomous agents that can function across a broad range of environments.

In this approach, the symbols used are initially undefined, as opposed to known by an initial teacher, and assumes no fixed reference points or common coordinate system, as opposed to the approach presented in [5]. Unlike the approach presented in [15], this approach is carried out during other work as opposed to during a labelling phase, grounds a practical number of points, and is shown to work with greater than 2 agents.

We are currently exploring a number of avenues of future research. The most obvious of these is evaluating these techniques in physical environments. The major potential discrepancy is in the error and noise associated with the real world as opposed to that supported by Player/Stage. While there is the possibility of localization and perception problems in Player/Stage, these significantly underestimate those of most complex real-world domains. Greater error in the real world will lead to mistaken groundings which the techniques described here, or adaptations of them, will have to be powerful enough to overcome. For example, a robot could misperceive a feature and ground an incorrect location, or demonstrate a grounding based on incorrect assumptions about its localization. When this occurs, the process of learning and sharing groundings must be used to correct these problems.

Our laboratory is currently performing an evaluation of these techniques in the real world, using pioneer robots in an office environment, under conditions of sensor noise and imperfect localization using a particle filter. Here groundings can also serve a dual purpose, for communication as described in this work, but also as aids to localization itself. As part of this work, we are also adapting these techniques to heterogeneous agents. Heterogeneity adds great complexity and potential error to this work, since one agent may try to describe a grounding to another that cannot actually perceive the same phenomena. In this case, agents must attempt to ground new locations they cannot perceive in terms of existing groundings that are perceivable [1].

There are also numerous additional areas for potential future research. For example, the specific techniques we developed for choosing groundings were meant to be examples of fully domain-independent, task-specific (navigation), and domain-specific (office navigation), in order to examine agent performance against the amount of knowledge supplied by human developers. A broad range of other techniques (or combinations of techniques) are possible. The application of machine learning or evolutionary approaches to this would also be very interesting, so that agents could learn appropriate grounding techniques for the domain at hand.

As a final note, it is noted in [19] that a bug in recognizing the second category of conflict detailed in Section 14.3.3 may have negatively influenced the results obtained. These experiments were entirely re-done with this bug fixed, and resulted in only a very small (1%) difference.

References

1. Abdellateef, A.K.E.: Grounded communication for heterogeneous multi-robot systems. Master's thesis, Department of Computer Science, University of Manitoba, Winnipeg, Canada (2008)
2. Arkin, R.: Behavior-Based Robotics. MIT Press, Cambridge (1998)
3. Balch, T., Arkin, R.: Communication in reactive multiagent robotic systems. Autonomous Robots 1(1), 27–52 (1994)
4. Baltes, J., Anderson, J.: Flexible binary space partitioning for robotic rescue. In: Proceedings of IROS 2003, Las Vegas, pp. 3144–3149 (2003)
5. Billard, A., Dautenhahn, K.: Experiments in learning by imitation – grounding and use of communication in robotic agents. Adaptive Behavior 7(3/4), 415–438 (1999)

6. Coradeschi, S., Saffiotti, A.: Anchoring symbols to sensor data: preliminary report. In: Proceedings of AAAI 2000, Austin, pp. 129–135 (2000)
7. Coradeschi, S., Saffiotti, A.: An introduction to the anchoring problem. Robotics and Autonomous Systems 43(2-3), 85–96 (2003)
8. Davis, I.: Warp speed: Path planning for star trek: Armada. In: Proceedings of the AAAI Spring Symposium on AI and Interactive Entertainment (2000)
9. Deacon, T.: The Symbolic Species: the co-evolution of language and the human brain. Penguin Books, Toronto (1997)
10. Elfes, A.: Occupancy grids: a probabilistic framework for robot perception and navigation. PhD thesis, Department of Electrical and Computer Engineering, Carnegie-Mellon Univesity (1989)
11. Etzioni, O., Segal, R.: Softbots as testbeds for machine learning. In: Proceedings of the Machine Learning Workshop at AI/GI/VI 1992 (May 1992)
12. Gerkey, B., Vaughan, R., Støy, K., Howard, A., Sukhtame, G., Matarić, M.: Most Valuable Player: A Robot Device Server for Distributed Control. In: Proceedings of IROS 2001, Wailea, Hawaii, pp. 1226–1231 (2001)
13. Hanks, S., Pollack, M.E., Cohen, P.R.: Benchmarks, test beds, controlled experimentation, and the design of agent architectures. AI Magazine 14(4), 17–42 (1993)
14. Harnad, S.: The symbol grounding problem. Physica D 42, 335–346 (1990)
15. Jung, D., Zelinsky, A.: Grounded symbolic communication between heterogeneous cooperating robots. Autonomous Robots 8(3), 269–292 (2000)
16. Matarić, M.: Using communication to reduce locality in distributed multiagent learning. Journal of Theoretical and Experimental AI 10(3), 357–369 (1998)
17. Steels, L.: Constructing and sharing perceptual distinctions. In: van Someren, M., Widmer, G. (eds.) Proceedings of the European Conference on Machine Learning, pp. 4–13. Springer, Heidelberg (1997)
18. Vogt, P.: Symbol grounding in communicative mobile robots. In: Proceedings of the AAAI Fall Symposium on Anchoring Symbols to Sensor Data (2001)
19. Wiebe, N.: Developing grounded communication in multi-agent systems. Master's thesis, Department of Computer Science, University of Manitoba, Winnipeg, Canada (January 2006)