A Physically Grounded Approach to Coordinate Movements in a Team[1]

Letizia Leonardi[1], Marco Mamei[2], Franco Zambonelli[2]

1) Dipartimento di Scienze e Metodi dell'Ingegneria – Università di Modena e Reggio Emilia
2) Dipartimento di Ingegneria dell'Informazione – Università di Modena e Reggio Emilia
{letizia.leonardi, mamei.marco, franco.zambonelli}@unimo.it

Abstract

This paper focuses on the problems of coordinating the movements of a cooperative team in an environment, and proposes an approach that takes inspiration from the laws of physics. Our idea is to have the movements of team members driven by abstract force fields, generated by team members themselves (i.e., by carried-on devices) and propagated via some embedded infrastructure (or by team members in an ad-hoc way). A globally coordinated and self-organized behavior in team members' movements emerges due to the interrelated effects of team members following the shape of the fields and of dynamic fields re-shaping. A case study in the area of warehouse management is introduced to exemplify the proposed approach.

1. Introduction

The latest trends in communication technology and mobile computing are introducing radically new information technology application scenarios. In particular these new technologies will give people the possibility to communicate and teamwork independently from their location or from the fact that they are in movement [12]. However, while new devices to provide mobile computing are quickly gaining more and more power both in term of computational, storage and wireless-connectivity capabilities (following or even overcoming Moore's law), software engineering practices and current distributed programming models show their inadequacy in providing effective support to teamwork coordination, leading to fragile systems unable to cope with environmental dynamics and self-organization [17], as required by mobile computing scenarios.

One specific problem, which is the focus of this paper, is to enable robust and flexible coordination among the members of a teamwork that have to carry on their tasks by moving in an environment. The goals of their coordination can be various: letting members meet somewhere [4], distribute themselves accordingly to specific spatial patterns [18], or simply move in the environment without interfering with each other and avoiding the emergence of traffic jams [7]. To provide uniform support to all of the above coordination problems, the basic idea of the approach proposed in this paper is to provide team members (i.e., the location-sensitive devices they carry on) with simple yet contextual information supporting and facilitating the required coordination activities related to their movements. To realize the idea, we take inspiration from the physical world, and in particular from the way masses and particles in our universe move and globally self-organize accordingly to that contextual information which is represented by gravitational and elector-magnetic fields.

The Co-Fields model proposes expressing contextual information in terms of abstract "computational fields" (*Co-Fields*). Each device in an environment (mobile devices carried by team members as well as embedded computing devices) can generate and propagate, according to specific laws, component-specific fields conveying some application-specific information about the local environment and/or about itself. Moreover, mobile devices can perceive these fields and can react accordingly or, better, can suggest their owner on how to react). Such reactions are intended to solicit movement accordingly to the shape of the perceived fields, i.e., following the gradient downhill, uphill, or by following its equipotential lines. Therefore, team member activities and movements are simply driven by these abstract force fields, without any central controller. Engineering a coordination policy within this model is a bottom-up approach and consists in specify local interactions: how devices generate fields, how these fields are propagated, and how devices subscribe and react to the fields. The global coordination and teamwork simply emerges in a self-organized way from these local interactions patterns.

2. Co-Fields: Motivations and Model

In this section, a brief survey on current coordination models and middleware is presented to outline their inadequacy in coordinating activities in a mobile scenario. Then, the Co-Fields model, conceived to overcome such limitation, is presented. To give generality to the discussion, we will speak of "agents" as the active entities whose activities and movements are to be coordinated. Depending on the specific context, such agents may represent team members, the mobile devices they carry on, the software running on mobile devices, as well as mobile robots and sensors.

The concept of context-awareness, of primary importance in the areas of mobile and pervasive computing, will be central to our discussion too. In fact, Coordination and teamwork, by their own nature, require some sort of context awareness. An agent can teamwork and coordinate with other entities only if it is somehow aware of "what is around". Enabling adaptive and cost-effective teamwork between agents' requires minimizing both the agents' efforts in acquired contextual information and the agents' efforts in coordinating with each other accordingly to such information.

2.1. Inadequacy of Current Approaches

In the last few years, several middleware and coordination models, addressing – among the others – the problem of coordination and teamwork in a multi-agent system, have been proposed. We briefly discuss them by grouping into three categories: *(i)* models based on direct communication *(ii)* models based on shared data-spaces *(iii)* and models based on event publish/subscribe.

In *direct communication models*, a distributed application is designed by means of a group of agents that are in charge to communicate with each other in a direct and explicit way. Systems like Jini [9], FIPA-based agent systems [2], as well as P2P systems like JXTA [10] are examples of middleware infrastructures rooted on a direct communication model. The problem of this approach is that agents are placed in a "void" space: the model, per se, does not provide any contextual information, they can only perceive and interact with other agents, and the middleware support is mostly reduced to helping in finding communication partners. Thus, each agent has to "manually" become context aware by discovering the other entities in the environment. Therefore, the approach does not generally suits the coordination needs of mobile computing teamwork scenarios, in that it requires agents' notable efforts (both computational and communication) to acquire context-awareness and end up with ad-hoc solutions for a contingent coordination problem (decisions which are, consequently, brittle, not flexible, and not adaptable).

Shared data-space models exploit shared localized data structures in order to let agents interoperate and coordinate with each other. These data structures can be hosted in some data-space (e.g., tuple space), as in JavaSpaces [6] and MARS [4], or they can be carried on by agents themselves, as in Lime [13] and XMiddle [11]. In these cases, agents are no longer placed in a void space but they live in an environment that can be modeled and described in terms of the information stored in the data spaces that, being accessible only from a locality, can provide some sort of contextual information to agents without forcing them to directly communicate with each other. Still, the problem of the approach is that contextual information usually expresses raw local data that can be difficult for agents to "understand" and exploit to achieve their coordination tasks. In other words, coordination decisions have still to be taken directly by agents on the basis of the available data (thus requiring computational efforts), accordingly to some global policy that is either previously established (and thus is not flexible and adaptive) or it has to be acquired (thus requiring further communication efforts).

In *event-based publish/subscribe models*, a distributed application is modeled by a set of agents interacting with each other by generating events and by reacting to events of interest. Typical infrastructures rooted on this model are Jedi [5] and Jini Distributed Events [9]. Without doubt, event-based model promotes stronger context-awareness, in that components can be considered as dived in an active environment able of notifying them about what's happening around. This frees agents from the need of explicitly querying other agents or the environment (as in direct and data-space models), and thus leads to software systems that can be both computationally and communication efficient. The problem of this approach is that it is still too complicated: even if they are provided with all the information they need, agents have to apply a complex decisional algorithm to infer the right decision, about where to go, from their internal knowledge.

2.2. The Co-Fields Approach

The Co-Fields proposal is mainly driven by the above considerations and, to overcome the limitations of current approaches, aims at providing agents with abstract – simple yet effective – representations of the context. Such contextual information enables specific coordination activities to be implicitly and with minimal effort realized by agents, and to be automatically adapted to the dynamics of the execution scenarios.

The core idea in Co-Fields is to delegate to the infrastructure all the activities needed to set-up the proper conditions required to let the agents' coordinate in an almost automatic way. In particular we would like the infrastructure to build a global view of the environment tailored ad-hoc for specific agents' coordination tasks. Agents perceiving this coordination-specific view would be able to achieve their goal effortlessly, because the view represents the agents' context in the exact way needed for the agents' coordination task. Thus, while the infrastructure is in charge of tailoring this artificial, coordination-specific view, the agents simply have to blindly follow the prepared coordination policy. Following this approach, agents achieve their goal not because of their capabilities as single individuals, but because they are part of an (auto)organized system that leads them to the goal achievement.

The Co-Fields model can be schematized in the following four points:

1. The environment is represented and abstracted by "computational fields", spread by agents and by the infrastructure. These fields convey some useful information for the agents' coordination tasks and provide agents with strong coordination-task-tailored context awareness.
2. The coordination policy is realized by letting the agents to move following fields' "waveforms".
3. Environment dynamics (through the infrastructure) and agents' movements induce changes in the fields' surface, composing a feedback cycle that influences agents' movement.
4. This feedback cycle let the system (agents, environment and infrastructure) to auto-organize, so that the coordination task is finally achieved.

More in detail, a field can be defined as a distributed data structure composed by a unique identifier, a value (representing the field magnitude in that particular point), and a propagation rule. Fields can be generated by the agents or by the environment, and are propagated through the *space* as specified by their propagation rule. To support fields' propagation a proper infrastructure or middleware is required. This middleware can be based on an external server in charge of storing fields' values, but it can also be embedded in agents themselves and rely on an ad-hoc (epidemic) communication schema between agents. Fields can be static or dynamic: basically a field is static if once propagated its magnitude does not change over time; it is dynamic if its magnitude does. A field can be dynamic because for example its source moves. In a given environment, several different types of fields can exist and be propagated, accordingly to field-specific laws. Application-specific fields can also be defined and spread in an environment by application agents, to support application-specific problems. The achievement of an application-specific coordination task relies on the evaluation of an application-specific *coordination field*, as a combination (e.g., linear) of some of the perceived fields. The coordination field is a new field in itself, and it is built with the goal of encoding in its shape the agent's coordination task. Once a proper coordination field is available, agents can achieve their coordination task by simply following (deterministically or with some probability) the shape of their *coordination field*, like if they were walking upon the *coordination field* associated surface. Basically their actions will be based on following downhill or uphill the *coordination field*, (see Figure 1) or following one of its equipotential lines.

Our view is to consider a Co-Fields based system as a simple dynamical system. Agents are simply seen as balls rolling upon a surface whose shape is described by the coordination field. Complex movements are achieved not because of the agent will, but because dynamic re-shaping of this surface.

Coming to implementation issues, Co-Fields can potentially be implemented, as an overlay network, on any middleware providing basic support for data storing, communication and event-notification. In our simulations (see Section 3) we have assumed the presence of embedded servers capable of storing values and of notifying to other servers and to local agents events related to changes in the field values. In a preliminary set of implemented experiments, such servers have been implemented upon MARS tuple spaces [4]. MARS spaces have been allocated by IEEE 802.11 access points, have been programmed so as to store fields and to notify agents about local field changes, and have been complemented with a set of support agents in charge of propagating fields to neighbor access points.

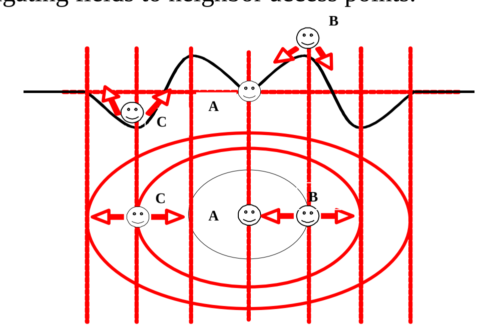

Fig 1. **Agent B follows downhill its *coordination field* evaluated as the field generated by agent A; Agent C follows uphill a *coordination field*; evaluated in the same way.**

3. An Application Scenario

To fix ideas on application scenarios and to clarify our model, we introduce a simple case study application consisting in a system to enable the teamwork and the coordination of respective movements of a group of employees driving forklift trucks in a warehouse company. For this application scenario, we can suppose that the warehouse is provided with an adequate computer infrastructure embedded in its rooms, and that employees drive computer-aided forklift trucks. In particular: one the one hand, there will be a network of computer hosts, each capable of communicating each other and with the mobile devices located in its proximity. In the following, we assume the presence of a host in each of the warehouse rooms and corridors, connected with each others accordingly to the warehouse plan (i.e., a server is connected to and only to servers in neighbors rooms and corridors), and providing the ability to determine which driving employee is in which room /corridor. On the other hand, a computer aided forklift truck provides users with a digital assistant capable to offer different services to its user based on the location-dependent information it is able to retrieve by connecting to the warehouse infrastructure. Given the above scenario, in particular we focus on how warehouse employees can be supported in coordinating and teamwork with the other employees. More in detail the coordination problems taken in consideration are: helping an employee to avoid traffic or queues while driving forklifts across the warehouse (thus realizing a sort of load balancing between employees and warehouse's rooms) and letting a group of employees to meet together at a suitable location.

3.1. Coordinating Agents Through Fields

Implementing the Co-Fields' model to the above described application scenario is straightforward: the networked infrastructure of the warehouse is used to store and propagate different types of fields, representing different aspects of the environment. Agents access the infrastructure by connecting to their closest host. Once connected, an agent can access only to the host's stored fields and to their gradients. In this way a strong locality scope for agent perception and interaction is enforced.

To solve the coordination activities studied in this paper (the load balancing and the meeting problems) three fields are required.

The *room field* (see Figure 2 left) is generated by every warehouse room. It simply has value 1 in the room that generates it and its value increases monotonically as the distance (measured in terms of hops number) from the source increases. In particular, we can simply imagine the field value is increased by 1 at every hop. Because the propagation rule follows a breadth first algorithm, problems related to multiple paths are avoided. The above fields are static and they do not change over time.

The *forklift field* is generated by every forklift within the warehouse. It simply has value 1 where the driving employee is located and its value increases monotonically as the distance from the source increases. The implementation description of this field is perfectly analogous to the room field's case, but these fields are dynamic and adjust their values over time, depending on agents' (i.e. forklifts') movements.

The *traffic-field* (see Figure 2 right) measures the amount of traffic in a room, and it is evaluated by considering the forklift trucks present in that room (i.e. connected to a certain server). The infrastructure evaluates the traffic field by simply considering the number of trucks connected to the infrastructure and normalizing that number to the dimensions of the room. The traffic field is dynamic and adjusts its values over time, depending on agents' movements.

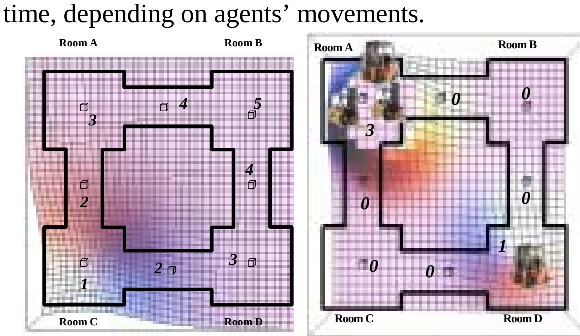

Fig. 2. (left) **room field of room C;** (right) **traffic field**

3.2. Load Balancing

The aim of this service is to help an employee to avoid traffic while driving its forklift truck through the warehouse. We assume that each room in the warehouse propagates the corresponding *room field*, each forklift in the warehouse propagates its own *forklift field* and that the infrastructure combine the forklifts' fields to obtain the *traffic field* as described above. In this section we are going to present the results obtained by a multi agent simulation of the problem. The model implementation is quiet straightforward: basically each forklift agents evaluates its *coordination field (CF)* as the sum between a minimum combination of the *room fields (RF)* in its working schedule (fields are combined by taking the in each point the minimum one) and the traffic field *(TRF)*.

$$CF = \min\,(RF_1, RF_2, \ldots, RF_n) + \lambda \cdot TRF$$

The first term of the coordination field, expresses a field surface having its minimum points in correspondence of the street/corners the agent has to visit. So, because each agent follows downhill the coordination field, this term guides the agent to visit the street/corners in its schedule. In order not to get stuck in a minimum, when the user completed the visit of a place, the corresponding field is removed from the combination. The place is thus removed and so it does not represent a minimum anymore. The second term of the coordination field takes into consideration the traffic management. In fact the term $\lambda \cdot TRF$ with $\lambda \geq 0$ is a field that has its maximum points where the traffic is particularly intense. When this term is added to the minimum combination it changes the steepness of the coordination field in the crowded zones. In particular a crowded zone tends to be a peak in the coordination field and thus it tends to repulse the income of other agents, thus enforcing the load balancing policy. To evaluate the performance of the Co-Fields model w.r.t. this problem, we tried a set of simulations in which a group of agents roam the warehouse independently visiting some rooms cyclically. We compared the case in which agents are not interested in the traffic field and no load balancing applies (see Figure 3) to the case in which the agents are interested to the traffic field, and thus the load balancing policies applies (see Figure 4).

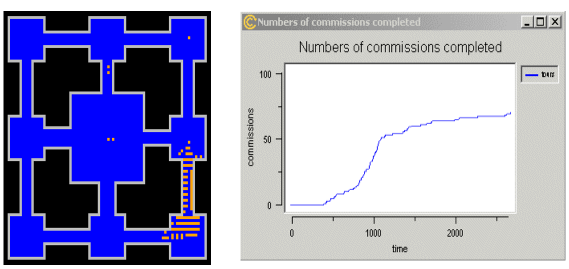

Fig 3. **Without load balancing (**left**) large traffic jams appears. The plot (**right**) represents the number of commissions completed by the agents.**

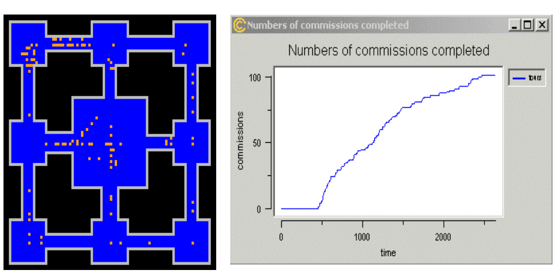

Fig 4. **With fields based load balancing (**left**) traffic jams are avoided. The plot (**right**) represents the number of commissions completed by the agents.**

3.3. Meeting

The aim of this service is to help a group of employees to dynamically find the most suitable room for a meeting. In particular, we can imagine that a group of employees want to meet in the room that is in the middle between them. To this purpose, each driving employee *i* belonging to the meeting-group can compose its coordination field by combining the fields of all the other employees in the group as well as the traffic field:

$$CF_i = \sum_{x \neq i} FF_x + \lambda \cdot TRF$$

Where FF_x is the field generated by forklift *x*.

In this way all the driving employees “fall” towards each other, and they meet in the room that is in the middle of them. The strength of this approach is that it is fully integrated with the field concept and that the meeting room is chosen dynamically to encounter the difficulties found by employees in real time (see Figure 5).

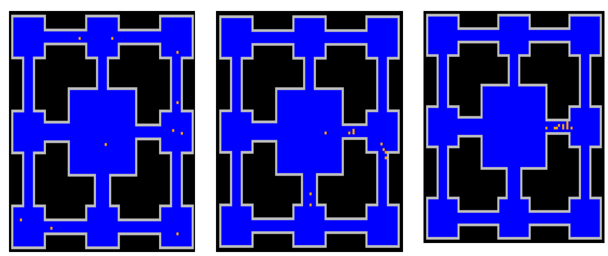

Fig 5. **From left to right different stages in the meeting process: forklifts converge towards each other.**

4. Related Works

An approach to coordinate movements that is closely related to Co-Fields is the one exploited to control non-player characters in the videogame “The Sims” [15]. The Sims are characters, living in a virtual world, whose behavior is directed by a "happiness landscape": the Sims traverse a spatial landscape of happiness values trying to increase their happiness, e.g., if they are hungry they perceive a happiness landscape where things providing food will have higher peaks and will start climbing that slope until getting to the fridge. After eating, all of a sudden the peak will collapse and a new landscape will appear to represent character happiness new requirements [16]. We think that the main difference between this and our approach is that “Sims’ happiness fields” tend to be static and generated only by the environment. On the contrary in our approach agents themselves are able to generate fields and thus a stronger (auto)organizational perspective is enforced.

Recent approaches to content-based distributed and pervasive computing exploit concepts that are somehow related to the ones exploited in Co-Fields. For instance, recent researches in the area of P2P computing recognize that retrieving information in dynamic networks, may require approaches in which information is searched following a down-hill approach, provided that the infrastructure makes available to peers some sort of overlay helping clients to navigate in the network progressively approaching the needed data [14]. In the area of sensor networks, the impossibility to have perfect knowledge about the position and status of the sensors requires queries to "diffuse" across sensors following some sort of gradient field, until the needed sensor is reached [8]. Although not explicitly oriented to movements' coordination, all these approaches shares with Co-Fields the physical inspiration and let us argues that they can be modeled by using Co-Fields (although we have not yet dealt with this issue).

The MMASS formal model for multi-agent coordination, described in [1], represent the environment as a multi-layered graph in which agents can spread abstract fields representing, different kinds of stimuli, so that agents' behavior can be influenced by the stimuli perceived in their location. The main difference between MMASS and Co-Fields is that, in Co-Fields, agents combine perceived fields and are constantly guided by the field produced. In their approach fields tend to be considered separately and they trigger one-shot reactions instead of guiding agents behaviors. Moreover also the application domain is quite different, while we are using this approach for the coordination of movements in a pervasive computing scenario, they are mainly focused on an agent approach to simulation (using a MAS to simulate artificial societies and social phenomena).

5. Conclusions and Future Works

In this paper we presented Co-Fields, a new model to coordinate the movements of a large number of autonomous agents in a mobile computing scenario. The model is based on the concept of computational force fields: distributed data structures providing to the agents an abstraction of the environment in terms of force fields driving agents towards the achievement of specific coordination task. A concrete case study has been presented to show the feasibility and the effectiveness of the approach.

Our future work will proceed towards two main directions. On the one hand we are currently completing the definition of a light, micro-kernel-based, event-based infrastructure [3], suitable as a supporting middleware for pervasive applications and resource limited devices. On the other hand, we are trying to extend the Co-Fields model and to formalize it. Our perception is that the model can be applied well beyond the case study application described in this paper, e.g. traffic management, manufacturing control, and robotics [7].

Acknowledgments. Work partially supported by Nokia Research Center Boston and by MIUR Project "MUSIQUE".

References

[1] S. Bandini, S. Manzoni, C. Simone, "Heterogeneous Agents Situated in Heterogeneous Spaces", 3rd International Symposium From Agent Theories to Agent Implementations, Wien (A), April 2002.

[2] F. Bellifemine, A. Poggi, G. Rimassa, "JADE - A FIPA2000 Compliant Agent Development Environment", 5th International Conference on Autonomous Agents, pp. 216-217, Montreal (CA), May 2001.

[3] G. Cabri, L. Leonardi, M. Mamei, F. Zambonelli, "Engineering Infrastructures for Mobile Organizations", 2nd International Workshop on Engineering Societies in the Agents' World, LNAI No. 2203, Dec. 2001.

[4] G. Cabri, L. Leonardi, F. Zambonelli, "Engineering Mobile Agent Applications via Context-Dependent Coordination", IEEE Transactions on Software Engineering, 2002, to appear.

[5] G. Cugola, A. Fuggetta, E. De Nitto, "The JEDI Event-based Infrastructure and its Application to the Development of the OPSS WFMS", IEEE Transactions on Software Engineering, 27(9): 827-850, Sept. 2001.

[6] E. Freeman, S. Hupfer, K. Arnold, "JavaSpaces Principles, Patterns, and Practice", Addison-Wesley, 1999.

[7] A. Howard, M. J. Mataric, "Cover Me! A Self-Deployment Algorithm for Mobile Sensor Networks", 2002 International Conference on Robotics and Automations, Washington DC, May 2002, to appear.

[8] C. Intanagonwiway, R. Govindam, D. Estrin, "Directed Diffusion: a Scalable and Robust Communication Paradigm for Sensor Networks", 5th ACM/IEEE Conference on Mobile Computing and Networking, Boston (MA), Aug. 2000.

[9] Jini, http://www.jini.org, Sun Microsystems

[10] JXTA, http://www.jxta.org, Sun Microsystems

[11] C. Mascolo, L. Capra, W. Emmerich, "An XML based Middleware for Peer-to-Peer Computing", 1st IEEE International Conference of Peer-to-Peer Computing, Linkoping (S), Aug. 2001.

[12] MOTION, IST Project, IST-1999-11400, http://www.motion.softeco.it

[13] G. P. Picco, A. L. Murphy, G. C. Roman, "LIME: Linda Meets Mobility", In Proceedings of the 21st International Conference of Software Engineering (ICSE'99), Los

Angeles, USA, May 1999.

[14] A. Rowstron, P. Druschel, "Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems", 18th Conference on Distributed Systems Platforms, Heidelberg (D), 2001.

[15] The Sims, http://thesims.ea.com

[16] S. Johnson, “Wild Things”, Wired, pp. 78-83, March 2002.

[17] F. Zambonelli, V. Parunak, “From Design to Intentions: Sign of a Revolution”, 1st International Joint Conference on Autonomous Agents and Multi-agent Systems, Bologna (I), July 2002.

[18] F. Zambonelli, A. Roli, ”On the Emergence of Macro Spatial Structures in Dissipative Cellular Automata, and its Implications for Agent-based Distributed Computing”, University of Modena and Reggio Emilia, Tech. Rep. No. DISMI-14, Oct. 2001.