Assessment in Team Training for UAV Operators

Daniel Luedemann, Guy Gouardères, Yon Dourisboure, Eric Gouardères
LIUPPA, University of Pau, Avenue de l'Université
BP 576 - 64012 Pau, France
{daniel.luedemann, guy.gouarderes, yon.dourisboure}@iutbayonne.univ-pau.fr,
eric.gouarderes@univ-pau.fr

Abstract. The main contribution of the paper focuses on social and collaborative learning. We propose a Team Training Architecture for UAV (Unmanned Air Vehicle) missions based on A.R.S. Agent, Roles, and Signature. After a brief discussion on motivations and background, the paper presents the concepts of the ECM model – Electronic Companionship Model – and the implementation of the A.R.S. agent based architecture with roles built on a P2P based interaction protocol to support the contract negotiation process. Finally we give an example of application.

Keywords: Team Training, Team Cognition, Peer-to-Peer Agents, Community of Practice, Roles.

1 Introduction

Today, in the field of aeronautic formation, expressions of individual training are well understood and paradigms for capturing and articulating skills and competencies at the level of the individual abound [1]. However, the same cannot be said for the capture of equivalent expressions at the collective or team level.

Team Training Environments are becoming prominent within Unmanned Aircraft System programs throughout the aviation community. In particular, the SOUL Project is developing a practical solution to this complex problem proposing a method for the joint training activities as a “Compagnonnage électronique” Model – Electronic Companionship Model or ECM (Fig. 1.).

The consequences for training design is that scenarios must be developed that reflects the real world and the importance of feedbacks from the battlefield where rapidly changing orders and information from multiple sources require skilled UAV crews. Three

1 System Oriented UAV Laboratory: Aerospace Valley Project, Thales Airborne - France
distinct roles for humans emerged: Pilot, Sensor Operator, and Flight Director, where one operator is assigned to execute one role exclusively during a team training session. Each of these roles relies on a different perspective on the operation: endocentric (or through the UAV’s “eyes”), exocentric (external view), or mixed (swapping between egocentric and exocentric).

Starting from the Combined Team Training model [2], two dimensions have been added to improve team performance and distinguish a high-performance team from other. The first dimension is Interpositional knowledge (IPK) [3], i.e. knowledge that individuals acquire through cross-training on each others’ jobs², and the second dimension is interdependent learning [4] because individual and team communication is essential for combined teams Training to accomplish their goals effectively.

Articulation of team training activities in the ECM framework requires that Team Members be trained within a Community of Practice (CoP) [5], under the supervision of training officers or instructors. Once all the team members are identified you can begin work on the third component -- Knowledge, Skill, Attitudes -- determining what role each team member plays. Each team member need to work in each position, actually performing the skills or procedures, to develop IPK as essential competence for the team.

The main contribution of the paper focuses on Implementing a Team training Model termed ECM with an innovative and open Contractual Agent Societies [6,13] built on the A.R.S. architecture (Agent, Role, Signature). We will first explain our motivation (section 2), in section 3 we will discuss about the A.R.S. architecture and we will finally see (section 4) how e-Qualification can be achieved based on the A.R.S. architecture giving a brief example of the first issues.

² To develop cross-training, the SOUL Lab offers simulation-based “On-the-Job” training sessions which include the procedures, skills, duties and responsibilities of all team members.
2. Motivation and Background

Interests in team training investigation, combined with successful tests of cross-training, gave rise to the concept of “Compagnonnage électronique” – or ECM in the field of aeronautic training. The underlying ideas of transactional relationships between learning and immediate and broader social context, dialog, collaboration and group learning have been privileged. According to [7], a team is “a distinguishable set of two or more people who interact dynamically, interdependently, and adaptively toward a common and valued goal/object/mission, who have each been assigned specific roles or functions to perform, and who have a limited life span of membership”.

Thus teams are special types of groups that are interdependent and that have specific roles for different team members. Command-and-control tasks and mission planning tasks involve teams with fluid-boundaries depending from episodic roles which appear and disappear.

In this context, team cognition is a cognitive activity that occurs at a team level. For example, team competency: awareness, knowledge, and skills. It is not the respective competency awareness, knowledge, and skills of each individual team member, but necessarily something based on interaction and probably something emergent [8].

2.1. Qualify both individual and team cognition

New training strategies (as IPK and Cross-Training) were readily adapted to the ECM framework and include interdependent team training and team reflexive helps and self-correction which allows team members to identify and correct problems in the team without help from an outside instructor. Interdependent Learning approach suggested that interaction processes could be studied as mediators of the effects of individual, group, and environmental factors on team output and cohesiveness. This approach to understanding team performance was called the input-process-output (I-P-O) framework and has been implemented with shared e-Portfolios in CoPs.

Intra-team (interpositional or common) knowledge and real competencies are seen as social constructs from both formal and informal learning. This part is supported by individual Agents proposing CoP members to enter contracts for a reflexive (P2P) assessment of their competencies.

An e-Portfolio is a tool for developing one's own knowledge and competencies, a kind of virtual representation of oneself. Each learner is trained alone on a networked simulation station but can share his experience with other CoP members. They are individually engaged in social contracts by agents that monitor their e-Portfolio to help trainees in fulfilling their goals. At the end of the learning process, learners are clustered in Communities of Competencies (CoCs) [9], thanks to the signatures extracted from contracts between e-Portfolios.
2.2. Actors, role and role models

According to the ECM Model an agent is viewed as an abstraction of an entity (human, software agents, devices) playing roles in CoPs. They may exhibit a flexible behavior in the same way that organizational roles refer to a position and a set of associated responsibilities in an organization [10, 13].

In sum, the major outcome of role facilities is the separation they enable between different phases of the development process. The most important limitation concerns the use of roles at run-time. Few approaches consider roles as first-class entities so that they are easily modified or reallocated. This is needed for some applications and environments such as Open Agent Systems that lack the ability of a dynamic assumption of roles in highly dynamic environments while not requiring the assignment of roles based on an agreement process between agents. This is a new contribution of the A.R.S. architecture.

3. The A.R.S. Architecture

The A.R.S. architecture is a conceptual framework based on three major concepts: Agents, Roles and Signature, designed from the ECM model to easily manage open groups with fluid boundaries to be used in dynamically and rapidly changing contexts.

Fig. 2. From CoP to CoC: enacting the ECM framework with A.R.S. agents
The A.R.S. architecture organizes various communication channels between humans, artificial agents and information storages in order to dynamically determine CoCs that are human communities sharing a common task. The members of a CoC have sufficient competencies and shared goals to assist each others in case of difficulties. The CoCs are often modified as a result of new tasks being assigned to the humans or as a result of some tasks being completed or abandoned. Several CoCs can be defined for a given CoP. (Fig. 2.) is an overall diagram that synthesizes the process of dynamically establishing CoCs. The detailed definitions and roles of the components and their functions are detailed afterwards in the paper.

In such environments there is no guaranty that a member is on-line at any moment. Moreover, members can move frequently from one group to another, the whole organization of agents is not predefined, neither are the rules that manage the groups. Decentralization seems to be, in such conditions, the only solution that avoids having “system-predefined” groups and efficiently manages groups. Open Agent Societies [11] are an ideal area in which to develop A.R.S. that could be considered as suitable solutions to the problems of flexibility and openness that many other organizational models in MAS don't solve. In this section we present the three main concepts of A.R.S

Team Training is seen as:
• Social learning, "located" (contextualized)
• Negotiating helps by communicative hints and shared resources
• Each team member is a "producer" (producer and consumer of resources)
• The learning is constantly co-constructed, negotiated and redefined by the team members

Interpositional knowledge emerges in open and interconnected situations and depends on social networking capabilities (Contracts).

3.1. Agents

An agent is an entity that is able to communicate with other agents. By design, this communication is done in a one-on-one, peer-to-peer (P2P) mode. An agent disposes of all necessary primitives needed to communicate properly in P2P: look for other peers, retrieve other peers' addresses, send and receive data. A.R.S. agents can be implemented on any platform providing access to low-level communication primitives.

We will assume that, in the other abstraction levels, an agent has a list containing other agent's addresses and that it can send and receive messages in a one-to-one way. Communication and interaction levels are defined using protocols as first class entities to provide a formal and reusable notation.
3.2. Communication level

As stated before, A.R.S. agents communicate in a P2P mode. They follow a straightforward protocol that works like a communication pipe. With this pipe, it doesn’t matters which agent sent the message. It's only important to know what a given couple of agents were talking about, not which of them initiated the interchange.

3.3. Interaction level

In the A.R.S. architecture an agent establishes contracts (commitments) with other agents. According to Dignum [12], a contract is a statement of intent that regulates agents' behavior among organizations designed with: a) specific norms: duration, conditions, sanctions, b) roles to be played (social contracts) and c) scenario issues (interaction contracts). In the following, the contract negotiation process is based on P2P exchanges between agents playing a particular role and is associated with the e-Portfolio changes. The signature describes roles and the normative rules of communities to which the agent belongs. When the negotiating process leads to a contract agreement, all the participants (via agents) share the same signature.

It is important to note that agents are triggered by context changes transmitted by e-Portfolio modifications. They don't care to know how the e-Portfolio was modified (possible sources of modification include the user, the simulator, an instructor, etc). The e-Portfolio in some way is the perception of the agent.

So, when something changes in the e-Portfolio, the agent that monitors that e-Portfolio will be activated and will try to find a suitable competence match for this modification in other e-Portfolios by broadcasting to the other agents that it knows and will wait for any positive answer.

4. Technical Features

4.1. Roles

Roles are a very important component of the A.R.S. architecture. We will try to carefully explain what “role” means in our model as it greatly differs from what we find in other models of agency. Usually, an agent is pre-assigned a role, giving the agents some “powers” or obligations, or even making it belong to a given team (more precisely to a CoP). In the A.R.S. architecture, the first basic underlying role is “to communicate in P2P mode”. Other roles emerge from the construction of opportunist contracts when they have been signed by two or more agents.
First, let us describe what contracts are. In the example we have been working on so far, thanks to their agents, users of the CoP make some agreements in order to share informal learning about a selected number of parameters (request or answer to a precise question exchanged between two members: i.e. “How to maintain flight heading”. This requires the altitude and speed values in addition to the heading value itself). An agent is allowed to make several agreements with several agents; in this case all agents that belong to one of these agreements make up what we call a contract based on social commitments that must be signed. Given the definition of contracts, we can now say that the role of an agent is the concatenation of the sub-roles that lead to agreements between that agent and other agents in the contract.

4.2. Signature

The signature of the CoC records the parameters of the successive roles (emerging competencies) played by the CoP members until they get the final contract. Each step is leading to a commitment in a contract. These steps make a strong use of the Mutual dependence of roles; this allows an indirect interaction between agents and e-Portfolios, and so between agents and learners.

Using the protocol shown previously for interaction for P2P exchanges implies that you define a mutual social influence (reflexive) between two agents and based on their respective competencies, extracted from the respective e-Portfolios.

As a consequence, a contract can be specified as a cluster of records of updated competencies and the signature is the vector that concatenates the competencies involved in successive commitments.
Moreover, being a member of a CoC (knowing that the membership is determined after the existing signatures) is an engagement towards all other members to comply with every rule of the signature.

4.3. Contract management

Let’s suppose that User A can’t get to accomplish a given task, because of some parameter P. This user will look into other users’ e-Portfolios in order to find some help. If he finds something useful, he will import that information into his own e-Portfolio. Upon detection of this modification, the Agent A that handles User A’s e-Portfolio will be triggered, and it will try to find other e-Portfolios in the CoC corresponding to the new value P’ for P.

Let’s suppose here that this agent does not belong to any contract. Looking for matches, Agent A broadcasts a request message and analyses every returned answer. If the answer is negative, it continues with the next answer. If the answer is positive, it will try to make a contract (existing or a new one) with the agent that answered positively.

From the point of view of the contract members, each member that accepts the newcomer must inform the other members, so that every member is aware of all the participants of the contract.

\[
\Delta(p_i, p_j) = \delta_{k(i)}
\]

\[
p_i - p_j = \delta_e
\]

5. E-Qualification Based on A.R.S.

The objective of the SOUL project was to develop radically new approaches for improving the future training and e-Qualification of the work force in aeronautics. The e-Qualification process is a two-step one. In the first step, agents will identify the major e-
Portfolio items that have changed during the training session. Simulation agents will follow learner's activities and detect errors and will try to find the best other learners who can help the requester. In the next step agents will try to map the learner signature in the suitable CoC.

5.1. First step: during the training session

This section details how agents involve their learners in contracts and these contracts during the training session. Before going further, let us explain how learners and agents are organized in CoPs and CoCs and how the e-Portfolio is updated according to the situation (see e-Portfolio items in Fig. 4.). Each user has, as we have mentioned before, one representative agent and his behavior (past, present and future), knowledge, skills and goals are stored in his e-Portfolio. We consider that there is a global ontology for all the training area.

Let's take an example in which we will follow the evolution of User A in Fig. 4., corresponding to “Izuko” who has the e-Portfolio containing the parameters depicted in Fig. 3.

From the initial state of the learner's e-Portfolio, gained TIPs\(^3\) are all null. The simulation starts and the values of parameters (Obtained VA, red arrow in Fig. 4. marked with ) will be changed by learners' agents. Agents will react consequently in order to find the best partners to help their learner improving his knowledge. The acquired knowledge will be seen as gained TIPs. When a learner is helped for a specific parameter he will assess the degree of benefits he obtained and will have new objectives consequently.

In the proposed example, User A has reached his objectives for Heading and BankAngle, but not for Airspeed. Thus, after the first exercise (blue arrow Fig. 4., marked with ), his agent (Agent A) looks for other agents in order to share some knowledge.

Difference on values between competencies \(\Delta(p_i, p_j)\) and goals raises the necessity to find a new contract. Since User A has reached his objectives in Heading and BankAngle, his agent will try to validate them by playing the role of advisor. For that, the agent needs to leave the current contract and above all it needs to find another contract within the CoP that fits better with its new goals.

Agent A proposes a signature containing Heading=\(-180\) and BankAngle=\(1.3\). Upon reception of this proposal, user’s U2 agent checks among its scores and objectives if any are compatible with the values proposed in the request.

Agent B accepts the agreement on both parameters (Heading and BankAngle), and Agent C accepts it only on parameter Heading. Agent B and C will then send each one a message to Agent A telling what they accept.

---

\(^3\) TIP : Training Information Point, it’s a kind of hint
From Agent A’s point of view, Agent B seems to be the most motivated to validate the agreement on the parameters Heading and BankAngle. But the acceptance of the agreement depends also on the signatures compatibility.

Thus, Agent A and Agent B negotiate in order to compute a new temporary signature, that represents what the contract’s signature would be if Agent A was included. It is computed by making the average of each item of Agent A’s signature and the new contract’s (Contract 2) signature.

The signatures of Agent A and Contract 2 have almost the same level of competence thus we assume that the compatibility is satisfied. Assuming that Agent B is also compatible with the new requested signature, they decide to try to introduce Agent A into the contract: Agent B broadcasts to all agents participating in Contract 2 the new signature in order to check if all agents of the contract are compatible.

Upon reception of this request, every agent of Contract 2 checks if it's compatible with the requested signature. If at least one of them is not compatible, then Agent A can't reach the contract and so the agreement is refused.

Let’s assume that Agent A is accepted by all agents of Contract 2. Then, once User 1 enters Contract 2 thanks to his agent, he plays the role of advisor on Heading and BankAngle, but he also plays the role of learner on the other parameters. This allows him to improve his knowledge. The acquired knowledge will be seen as gained TIPs. By the way, revises his objectives for the next exercise (black arrow on Fig. 4.,marked with ☀). Moreover, the new signature of Contract 2 will be accepted by all agents of the contract.

5.2. Signature construction

When the training session ends, each agent is member of at least one peer contract. The last contract is the most representative of the user’s profile. The last signature will contain all the parameters (related to previous competencies) of the initial signature (used to enter the CoP), and the new updates (acquired competencies). For instance, in Fig. 3., the CoP member “Izuko” has improved his personal planned score for BankAngle (< 0.5 degree) by getting a 0.0 which is far better than the requested 0.5 (less is better). If a contract acknowledges this enhancement for other members, this atomic competence becomes a common knowledge and the signature is updated.

5.3. Second step: debriefing and assessment

Once the signature has been extracted, the agent will search for a CoC that corresponds to the learner's profile and where the learner will be able to satisfy his goals. As specified in the A.R.S. architecture, all the agents of a CoC can play the role of manager according to the CoC life cycle: creation, operations, evolution and dissolution.
Creation of a new CoC: when an agent doesn't receive any positive answer to the request.

Two operations: a) Joining a CoC: the newcomer informs the other agents via a broadcast message that he is a new member; b) leaving: the agent has to inform the whole members that he leaves the community.

Evolution: When a new agent joins the CoC others agents choose whether they update the signature (adding for example new objectives) or not. All the members will 1) use the same rules, 2) apply the same changes and 3) get the same signature updated in a decentralized way.

Dissolution: A CoC disappears when it contains no members.

To sum up, two kinds of communities are required to qualify learner-acquired knowledge in team training mode: The CoP and the CoC. In CoP, agents engage the learners in contracts following their evolutions. Contracts are a social aspect that helps to group learners in the best way for belonging to a qualified CoC.

This first example of the A.R.S. architecture with parameter values to be attained collectively by heterogeneous member teams (with an average of ten to twelve members in a CoP, which is the norm for UAV missions) has been implemented, in the context of the SOUL project [13], highlighting the effectiveness of a MultiAgent System (implemented in Java and MadKit) enforcing a dynamic contractual agent framework Fig. 4.

6. Conclusion

Work on the A.R.S. architecture is continuing, with the aim of achieving more complex types of dynamic contracts between members of heterogeneous teams. The quality of human training achieved by this approach is being evaluated and compared to other approaches.

Current experiments: work on the “e-Portfolio server”, where one can manipulate e-Portfolios, realized by Master students; e-Portfolio alignment to group CoP members in CoCs, realized by “IUT” (undergraduate) Students (the chart seen in Fig. 3. is the result of the e-Portfolio alignment obtained by these students).

Future experiments on e-Portfolios are planned, with the contribution of the French artillery regiment specialized in UAVs.

Acknowledgment

Work partially supported by the SOUL program.

4 Aerospace Valley competitiveness cluster: Aeronautics, Space and Embedded Systems
References

7. Salas, E., et al., Toward an understanding of team performance and training. Teams: Their Training and Performance, pp. 3-29 (1992)