Towards a Model of Evolving Agents for Ambient Intelligence

Stefania Costantini[1] and **Pierangelo Dell'Acqua**[2] and **Luís Moniz Pereira**[3] and **Francesca Toni**[4]

Abstract. We propose a general vision for agents in Ambient Intelligent applications, whereby agents monitor and train unintrusively human users, and learn their patterns of behavior by observing and generalizing their observations, but also by "imitating" them. Agents can also learn by "imitating" other agents, after being told by them. Within this vision, agents need to evolve to take into account what they learn from or about users, and as a result of monitoring the users. In this paper we focus on modelling, by means of dynamic-logic-like rules, the monitoring behavior of agents, and by modelling the corresponding evolution of the agents.

1 Motivation

We envisage a setting (see Figure 1) where agents interact with users (i) with the objective of training them in some particular task, and (ii) with the aim of monitoring them for ensuring some degree of consistence and coherence in user behavior. We assume that agents are able (iii) to elicit (e.g. by inductive learning) the behavioral patterns that the user is adopting, and (iv) to learn rules and plans from other agents by imitation (or being told). In fact, learning may allow agents to survive and reach their goals in environments where a static knowledge is insufficient. Here, for some aspects related to learning we take inspiration from recent evolutionary cultural studies of human societal organization to collectively cope with their environment. We believe in fact that some principles emerging from these studies can equally apply to societies of agents. This especially when agents cooperate to help humans adapt to environments that are new to them and/or their ability to cope with the environment is too costly, non-existent or impaired.

The envisaged agents will try to either modify or reinforce the rules/plans/patterns they hold, based on appropriate evaluation performed by an internal meta-control component. The evaluation might also convince the agent to modify its own behavior by means of advanced evolution capabilities.

This overall agent model is in accordance with the vision of Ambient Intelligence as that of a digitally augmented environment centered around the needs of humans, where appliances and services proactively and unintrusively provide support and assistance.

We consider it necessary for an agent to acquire knowledge from other agents, i.e. learning "by being told" instead of learning only by experience. Indeed, this is a fairly practical and economical way of increasing abilities, widely used by human beings, as widely studied in evolutionary biology [12].

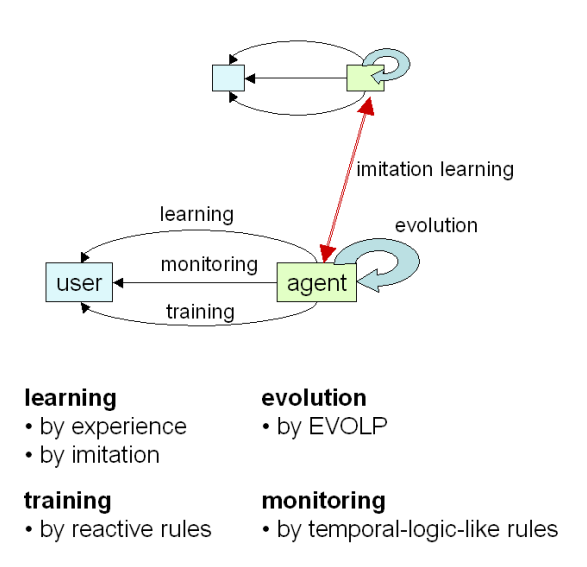

Figure 1. Agent interaction model

Note that avoiding the costs of learning is an important benefit of imitation, but nevertheless learning involves many issues and some potential risks. The issues are at least the following: (a) how to ask for what an agent needs; (b) how to evaluate the actual usefulness of the new knowledge; and, (c) how this kind of acquisition can be semantically justified in a logical agent. We will discuss issues (b) and (c) in Sections 3 and 4 while we shortly discuss (a) in Section 5. We make the simplifying assumption that agents speak the same language, and thus we overlook the problem of ontologies. We also assume that agents involved in the society are benevolent and trusted. Otherwise, incorporating and using learned knowledge would involve the management of related risks.

Note also that an agent that learns and re-elaborates the learned knowledge becomes in turn an information producer, from which others can in turn learn. Instead, an agent that just imitates blindly can be a burden for the society to which it belongs. Then, one my wonder about the effects and risks for the society to allow imitation. Evolutionary biology shows that the long-run of evolution of human

[1] Università degli Studi di L'Aquila, Dipartimento di Informatica, L'Aquila, Italy
[2] Department of Science and Technology - ITN, Linköping University, Norrköping, Sweden
[3] Centro de Inteligência Artificial (CENTRIA), Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Caparica, Portugal
[4] Department of Computing, Imperial College London, South-Kensington Campus, London, UK

societies is a mixture of learners and copiers, in which both types have the same fitness as purely individual learners in a population without copiers. To understand this result, think of imitators as information scroungers and of learners as information producers. Information producers bear a cost to learn. When scroungers are rare and producers common, almost all scroungers will imitate a producer. If the environment changes, any scroungers that imitate scroungers will get caught out with bad information, whereas producers will adapt.

Then, an agent will be able to increase its fitness in such a society in two ways: if it is capable of usefully exploiting learned knowledge thus deriving new knowledge and becoming an information producer; if it is capable to learn selectively, learning when learning is cheap and accurate, and imitating otherwise. A future direction of this work is that of equipping agents with a higher level responsible for coping with this kind of information exchange.

2 Application to Agent Societies for Ambient Intelligence

In the sequel we shall outline a model for the construction of logical agents that are able to learn and adapt agents in interaction with humans.

Let us emphasize that, to engage with humans, agents should have a description of how humans normally function. Clearly, the description will in general be initially limited to the "normal" user behavior in that ambient setting. We assume that the agents are deliberately designed and originally primed with some ambient setting in mind, and the humans are new to the setting and/or experience difficulties or impairments in coping with it. As deep learning is time consuming and costly, and thus needs not be repeated by one and all, an agent may apply a hybrid combination of both deep and imitation. The view is that all agents and the society as a whole will eventually take profit from the learning/imitation process, that can here be seen as a form of cooperation.

Each agent will thus initially contain abilities related to its supervision task. These may be enhanced by interaction with both the user and the environment, and with other similar agents. However, when some piece of knowledge is missing and a task cannot be properly carried out by an agent, that piece may eventually be acquired from the society, if extant there, for the agent may be unable or unwilling to deep learn it. Then it will exercise it in the context at hand, subsequently evaluate on the basis of such experience, and report back to the society. This evaluation of imparted knowledge builds up a network of agents' credibility and trustworthiness.

3 Agent model: sketch

In order to meet the vision outlined in Section 1, we consider an agent model composed of two layers:

- A *base layer* PA (for Personal Assistant) in charge of giving immediate answers to a user. We will assume that PA is a logic program, but will not commit to a particular semantics for it (for a survey of semantics for logic programs, see e.g. [1]). We will assume however a semantics possibly ascribing multiple models to PA, in order to deal with "uncertainty" (as we will see later). One such a semantics might be the stable model semantics [7].
- A *meta-layer* MPA in charge of updating PA when no model exists according to the chosen semantics for PA. This meta-layer relies on meta-knowledge, e.g. reporting long-term objectives about the user (e.g., safety and good health) and some domain-dependent meta-knowledge related to the PA. This domain-dependent knowledge may be updated by learning (by being told) from other agents.

To describe the dynamic changes of the user behavioral patterns as well as the environment, we assume that both PA and MPA are formalized via some kind of evolutionary programming paradigm. One possibility would be to exploit EVOLP, an extension of logic programming [8] that allows to model the dynamics of knowledge bases expressed by programs, as well as specifications that dynamically change.[5] EVOLP augments a given logic programming language by adding the new distinguished atom $assert(R)$, where R is a rule. The intended meaning is that whenever $assert(R)$ is a consequence of the given program, then the rule R is added to the program itself. Symmetrically, a rule can be removed by asserting its negation. The semantics of EVOLP is given in terms of *program sequences*, i.e., addition or removal of a rule transforms the given program into a new one which is its successor in the sequence. EVOLP is originally based on the "stable model" or (equivalently) "answer set" semantics [7]. However, EVOLP is a general framework that does not strictly depend upon the underlying semantics.

Note that the agent model we envisage here is an instance of a more general model, outlined in [6], whereby an agent results from activating some form of *control* in the context of an environment where the sensing, acting and communication capabilities can be put to work. An *initial agent* A_0 will in general *evolve* into A_1, A_2, ... through a sequence of stages, that will be affected by the interaction with the environment, that will lead it to respond, to set and pursue goals, to either record or prune items of information, etc. This more general model admits also KGP [2, 10, 13] and DALI [4, 5, 14] as instances.

In [3] we have introduced the possibility for the agent to learn reactive rules and plans. Once acquired, the new knowledge is stored in two forms: as plain knowledge added to the set of beliefs, so that the agent is able to use it and as meta-information, that permits the agent to "trace" the new knowledge, in the sense of recording what has been acquired, when and with what expectations. The meta-information allows the meta-control to reason about these aspects. If the agent should conclude that the new rules must be removed because the expectations have not been met, the meta-information will be used to locate the rules in the set of beliefs and remove them.

In this paper, we focus on the monitoring aspects of our agent vision (see Figure 1). For this purpose, we will assume that the meta-control includes rules inspired by temporal logic, but adapted to the agent context: the basic difference is that in the case of agents there is no way of verifying these rules along the full time-line (like it is e.g. done by model-checking). In fact, the notion of truth of a temporal formula in agents that evolve is necessarily bound to be checked at certain times and, as the agents evolve, the truth value may change, thus introducing an element of non-monotonicity. These temporal logic-like rules are described in the next section.

3.1 Temporal logic-like rules

Assume given a logical formalism L in which we can express sentences (including the sentence $true$). L will possibly include quantification. Then, temporal logic-like rules are defined as follows.

Definition 3.1 *Let P and C be sentences in L. Let T be a time-stamp or a time-interval. A* safety formula *$\mathcal{F}$ is a formula of the*

[5] An implementation of EVOLP is available from http://centria.fct.unl.pt/~jja/updates.

form $K\ F$ WHEN C *where either* $F = P$ *or* $F = P : T$, *and* $K \in \{\text{ALWAYS}, \text{SOMETIMES}, \text{NEVER}, \text{EVENTUALLY}\}$. *If* $K =$ EVENTUALLY *then the time-stamp is mandatory. If* C *is* $true$ *then the safety formula is abbreviated to* $K\ F$.

At a certain time t a safety formula can be either true or false (for simplicity we consider solely safety formulas $K\ P : T$ below). First, the inner sentence P will be either true or false according to the concrete logical formalism we are adopting. Then, $P : T$ is true at t if P is true at t and either T is a time-stamp and $t \leq T$ or T is a time-interval and $t \in T$.

Definition 3.2 *Let* T *be a time-stamp and* $\mathcal{F} = K\ P : T$ *a safety formula. Then:*

- $\mathcal{F}$ *is true at time* t *iff* $P : T$ *is true at* t *whenever* $K \in \{\text{ALWAYS}, \text{SOMETIMES}, \text{EVENTUALLY}\}$;
- $\mathcal{F}$ *is true at time* t *iff* $P : T$ *is false at* t *whenever* $K =$ NEVER.

Defining the truth of safety formulas in time-intervals requires the interval to be specified as a totally ordered set of discrete points.

Definition 3.3 *Let* T *be a time-interval and* $\mathcal{F} = K\ P : T$ *a safety formula. Then,* $\mathcal{F}$ *is true iff:*

- $K =$ ALWAYS *and* $\forall\ t \in T$ *it holds that* $K\ P$ *is true at* t;
- $K =$ NEVER *and* $\forall\ t \in T$ *it holds that* $K\ P$ *is false at* t;
- $K =$ SOMETIMES *and* $\exists\ t \in T$ *such that* $K\ P$ *is true at* t;
- $K =$ EVENTUALLY *and* $\exists\ t \in T$ *such that* $K\ P$ *is true at* t *and* $\forall\ t_2 \in T, t_2 > t$ *implies that* $K\ P$ *is true at* t_2.

Notice that the notion of truth/falsity is necessarily bound to be checked at certain times and the outcome in general will change as the agent evolves: what was *ALWAYS* (or *NEVER*, etc.) true at some point may not be so in a previous or later point.

4 User monitoring by learning-by-imitation and evolution: case study

The following scenario illustrates the dynamic aspects of the knowledge base of a PA/MPA whose knowledge evolves to reflect changes in the user behavior as well as in the environment.

Suppose we have a user who must undergo treatment for some illness and therefore needs to take medicine. He/she asks his/her personal assistant about what to do during treatment, e.g., "Can I drink a glass of wine if I have to take this medicine?" Or, more generally, the user may just ask "Can I drink a glass of wine now?" where the personal assistant should give advice based on whether there is medicine to be taken (or other related matters). Referring to the first question, PA may initially contain:

$\bot \leftarrow drink, take\_medicine$

plus default usage rules:

$drink \leftarrow not\ abnormal(drink)$

$take\_medicine \leftarrow not\ abnormal(take\_medicine)$

When asserting (triggered by the user's question):

$drink$

$take\_medicine$

an integrity violation is detected because the symbol $\bot$ is in some models (in fact all, in the stable model semantics). The PA can ask the MPA for help, and it might provide in the first place general rules such as:

$abnormal(drink) \leftarrow not\ abnormal(take\_medicine)$

$abnormal(take\_medicine) \leftarrow not\ abnormal(drink)$

together with rules stating that facts about abnormality should be rejected. The MPA can however have meta-axioms stating that a user action which is necessary to reach a basic objective should be undertaken, e.g.

ALWAYS $do(user, A)$
WHEN $goal(G), necessary(G, A)$

$goal(healthy)$

$necessary(healthy, take\_medicine)$

then, the provided rules might be, accordingly:

$abnormal(drink) \leftarrow not\ abnormal(take\_medicine)$

$abnormal(take\_medicine) \leftarrow not\ abnormal(drink)$

$\bot \leftarrow not\ take\_medicine, mandatory(take\_medicine)$

$mandatory(take\_medicine)$

where the latter fact signifies taking the medicine cannot be avoided.

Let us assume something more, to complicate the matter a little so as to show how the MPA can evaluate rules acquired from its siblings. Assume that the MPA knows:

$illness(user, cold)$

$goal(healthy) \leftarrow illness(user, X), recover(X)$

and has learnt:

$recover(cold) \leftarrow do(user, take\_aspirin)$

Now, the MPA will check the usefulness of the learnt rule, e.g. by means of the meta-axiom:

EVENTUALLY $goal(G) \leftarrow$
$known\_conds(C),\ learnt(Cond)\ :\ t$

The intended meaning is that goal G is expected to be reached by time t, by means of: (i) what the agent knew before, here indicated as $known\_conds(C)$ and corresponding to $illness(user, cold)$ in the example; (ii) the learnt condition $Cond$, i.e., $recover(cold)$ in the example. If this does not happen, in that the PA, by virtue of its interaction with the user, does not confirm recovered health, the learnt rule can be either de-activated or removed.

In accordance with the vision of Ambient Intelligence as a digitally augmented environment which is omnipresent and can observe and supervise the situation, our assistant agent will be able to perceive and record data about user behavior. In fact, the description of the user begins with an initial form and will then be subject to evolution according to what the agent observes along the interaction. These data can be exploited by means of either induction, abduction, or some other classification method so as to predict plausible future user behavior (for induction and abduction in logic programming, see e.g. [11] and [9] respectively) . For instance, assume that our agent is able to learn that the user normally takes a drink when coming back home. This can be represented by a rule such as:

$drink \leftarrow arrive\_home$

This learnt rule can be associated with a certainty factor. When the rule becomes later confronted with subsequent experience, its certainty factor will be updated, accordingly. Whenever this factor exceeds a threshold, this may lead to assert new meta-knowledge, such as:

USUALLY $drink$ *WHEN* $arrive\_home$

Formulas including *USUALLY* are an extension to the language for meta-rules given earlier. They express simply a constraint that should be checked periodically during evolution. A specification of the frequency of the check and of the conditions under which the check should be performed may be in principle added.

Th meta-knowledge expressed by the *USUALLY* formula should be managed by the meta-control MPA. In particular, MPA should consider all constraints that involve one of the elements. In this case, the outcome should be that, whenever the user arrives home, if she/he is undergoing some treatment and should then take medicine, he/she

is preemptively warned not to drink.

The initial description of the user can be either hard-wired in the agent program, or more generally be acquired from the agent society. The society will in general provide, initially and later:

- A few mandatory rules.
- Behavioral rules that each agent has the freedom to accept, reject or modify in accordance to its experience and type of user it is supervising.

5 Towards a society of agents

Throughout this paper, for the sake of simplicity we have assumed that learning is achieved via information exchange between sibling agents. However, in our envisaged system architecture, the role of the society is crucial. In fact, we plan to specify a meta-meta-level which is present in every agent which participates in the society. This higher level should be responsible for such information exchange. This could be achieved, for example, by exploiting and developing techniques based on social evaluation and consensus, involving credibility measures and overall preferences.

Thus, in this perspective, a set of rules should not be told directly by an agent to another agent but, instead, it should be acquired by the global agent society which, in turn, will have suitable self-evaluation mechanisms. According to this vision, the society will have the role of proposing behavioral rules (that are socially accepted) to its agents, which have the freedom to accept them in accordance to their experience and to the type of user they are monitoring. It is also plausible that the agent society should have the possibility to enforce mandatory rules.

In this architecture, any time an agent provides its evaluation to the agent society, that agent is responsible for the information it provides. This agent will then be rewarded in case the rule it proposes will be positively evaluated by other agents. Doing so will increase the reputation/trust of the proposing agent with respect to society, and the future rules proposed by it will be accepted with greater strength. On the contrary, agents proposing bad rules will be penalized, and eventually will be socially eliminated or outcast, and eventually replaced by new agents. The resulting agent society is thus not static, but self-evolves by trying to adapt to new situations. For example, it may revise its policy to reward/punish agents, etc.

The function of the society is particularly important in contexts where agents can be dynamically allocated to new "roles". Assume for instance that an agent is required to act as a baby-sitter. The kind of knowledge it will be equipped with can consist for instance of the following.

Mandatory rules (some examples):

- Children cannot drink alcohol. This is to be strictly observed.
- Children have to go to bed "early". Each agent can however interpret what "early" means, according to children's age and family habits.
- Children should not watch too much television. Here, each agent can define what "too much" may mean, also according to circumstances and type of program.

Optional" rules to be interpreted, adapted and possibly ignored or modified (some examples):

- Children should eat healthy food (if available, with the exception of e.g. birthday parties).
- Children should benefit from fresh air and exercise: the agent should find ways of fulfilling this requirement.

6 Concluding Remarks

There are several future directions for the ideas that we discussed and sketched in this initial work.

First, we intend to develop a full realization of these ideas, staring from EVOLP, DALI and KGP agents that provide the main elements and can be exploited in combination in an implementation. We have discussed a semantic framework for such an integration in [3].

Next, we aim at designing the meta-meta level for controlling knowledge exchange. Particular attention should be dedicated to strategies involving reputation and trust for the evaluation of learnt knowledge. The social acceptance of rules can be partly based on existing techniques and algorithms. However, we believe that an extension is necessary because, where learning is concerned, techniques that just measure reputation/trust on the basis of agents' feedback are not sufficient: some kind of economical and efficient evaluation of both the degree of compatibility of the new knowledge with an agent's previous knowledge base and of the performance of the acquired rules with respect to the expected objectives is also required.

REFERENCES

[1] K. R. Apt and R. Bol, 'Logic programming and negation: A survey', *The Journal of Logic Programming*, **19-20**, 9–71, (1994).

[2] A. Bracciali, N. Demetriou, U. Endriss, A. Kakas, W. Lu, P. Mancarella, F. Sadri, K. Stathis, G. Terreni, and F. Toni, 'The KGP model of agency: Computational model and prototype implementation', in *Global Computing: IST/FET International Workshop, Revised Selected Papers*, volume 3267 of *LNAI*, 340–367, Springer-Verlag, Berlin, (2005).

[3] S. Costantini, P. Dell'Acqua, L. M. Pereira, and A. Tocchio, 'Conditional learning of rules and plans in logical agents'. Submitted.

[4] S. Costantini and A. Tocchio, 'A logic programming language for multi-agent systems', in *Logics in Artificial Intelligence, Proc. of the 8th Europ. Conf.,JELIA 2002*, LNAI 2424. Springer-Verlag, Berlin, (2002).

[5] S. Costantini and A. Tocchio, 'The dali logic programming agent-oriented language', in *Logics in Artificial Intelligence, Proc. of the 9th European Conference, Jelia 2004*, LNAI 3229. Springer-Verlag, Berlin, (2004).

[6] S. Costantini, A. Tocchio, and F. Toni, 'A multi-layered general agent model'. Technical Report, 2006.

[7] M. Gelfond and V. Lifschitz, 'The stable model semantics for logic programming', in *Logic Programming, Proc. of the Fifth Joint Int. Conf. and Symposium*, pp. 1070–1080. MIT Press, (1988).

[8] J. J.Alferes, A. Brogi, J. A. Leite, and L. M. Pereira, 'Evolving logic programs', in *Logics in Artificial Intelligence, Proc. of the 8th Europ. Conf., JELIA 2002*, LNAI 2424, pp. 50–61. Springer-Verlag, Berlin, (2002).

[9] A. Kakas, R. Kowalski, and F. Toni. The role of abduction in logic programming, 1998.

[10] A. C. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni, 'The KGP model of agency', in *Proc. ECAI-2004*, (2004).

[11] Stephen Muggleton and Luc De Raedt, 'Inductive logic programming: Theory and methods', *Journal of Logic Programming*, **19/20**, 629–679, (1994).

[12] P. J. Richardson and R. Boyd, *Not by Genes Alone - How Culture Transformed Human Evolution*, The University of Chicago Press, 2005.

[13] K. Stathis and F. Toni, 'Ambient Intelligence using KGP Agents', in *Proceedings of the 2nd European Symposium for Ambient Intelligence (EUSAI 2004)*, eds., P. Markopoulos, B. Eggen, E. H. L. Aarts, and J. L. Crowley, number 3295 in Lecture Notes in Computer Science (LNCS), pp. 351–362, Eindhoven, The Netherlands, (8–10 November 2004). Springer Verlag.

[14] A. Tocchio, *Multi-Agent systems in computational logic*, Ph.D. dissertation, Dipartimento di Informatica, Universitá degli Studi di L'Aquila, 2005.