DESIGNING SELF-MODIFYING AGENTS[*]

FRANCES M.T. BRAZIER AND NIEK J.E. WIJNGAARDS
Intelligent Interactive Distributed Systems Group,
Faculty of Sciences, Vrije Universiteit Amsterdam,
de Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Email: {frances,niek}@cs.vu.nl
URL: http://www.iids.org

Abstract. Agents need to be able to adapt to changes in their environment. One way to achieve this, is to provide agents with the ability of self-modification. Self-modification requires reflection and strategies with which new knowledge can be acquired, a necessary condition for creativity. This paper describes a knowledge-level model for the design of self-modifying agents and explores the feasibility of automatically designing self-modifying agents.

1. Introduction

Intelligent agents typically operate in unpredictable domains. The problems to be solved are non-trivial, and may require non-standard solutions. Whenever the context of an agent changes to the extent that an agent is unable to cope with (parts of) the context, an agent needs to adapt.

There are choices in the extent to which an agent may adapt itself. For example, an agent may switch to a different plan or goal or learn new facts. A more extreme form of adaptation is for an agent to modify its internal processes: a self-modifying agent. This enables an agent to change the way it reasons and solves problems. It reflects on the manner in which it solves a specific problem, and adjusts its approach; an ability subscribed to reflective practitioners, as discussed by Schön (1983).

A self-modifying agent, in fact, re-designs its internal data and processes. This re-design process may be part of a self-modifying agent, or may be a service provided by another agent which represents an agent factory. A self-

[*] To appear in Proceedings of the Creative Design Workshop, December 2001.

modifying agent may, for example, employ an agent factory to modify its self-modification capabilities.

A design agent, capable of modifying other agents and itself, was presented in (Brazier, Jonker, Treur and Wijngaards, 2001). A self-modifying agent is a next step in the evolution of design agents. A *self-modifying agent* is defined in this paper to be an agent which can adapt to changes in its environment. This process may include modifying its data, task, and agent specific processes.

A knowledge-level analysis of a self-modifying agent is described in this paper. This analysis abstracts from specific tasks of a self-modifying agent, and whether self-modifying agents co-operate or pursue their goals individually. It shows that the design process of a self-modifiying agent entails not only the design of a conceptual specification of a self-modifying agent, but also a detailed description plus an operational description. The design process ranges from conceptual plans to realisation of a self-modifying agent. This makes the design process comparable to the design of many (dynamic) artefacts. Dynamic artefacts are artefacts that exhibit changes in their behaviour, based on changes in their environment. Examples of dynamic artefacts include houses which adjust lighting and temperature on the basis of occupation of rooms (Mozer, 1999), elevators which try to second-guess the behaviour of their clientèle, autopilots of aeroplanes, which take, and relinquish, control to the human pilots, or self-configuration of autonomous (spacecraft) systems (Williams and Nayak, 1996).

In literature on creativity (e.g., Schön, 1983; Finke, Ward and Smith, 1992; Edmonds and Candy, 1997; Gero, 1996; Lawson, 1997) it is generally accepted that a designer, or any expert, who is able to relax constraints, and modify implicit assumptions and requirements may devise creative solutions. Reflection on their problem-solving capabilities combined with reflection on the problem at hand, plays an important role in this behaviour. A model for a self-modifying agent needs to include such types of knowledge, a means for self-modification plus insight in designing such agents. The model for self-modification is independent of the agent's specific task. The design of self-modifying agents forms a step towards the design of creative agents.

In Section 2, research on self-modifying agents is discussed. A knowledge-level model of a self-modifying agent is described in Section 3. Issues involved in constructing self-modifying agents are discussed in Section 4. An information retrieval agent is used to illustrate concepts identified in sections 3 and 4. The results presented in this paper are discussed in Section 5.

2. Research on self-modifying agents

Research on self-modifying agents involves understanding agents, reflection, and design. A number of design methodologies for (intelligent) agents and multi-agent systems that are being developed are discussed in Section 2.1. Research on adaptive agents is discussed in Section 2.2 and research on self-modifying agents is discussed in Section 2.3.

2.1. DESIGN METHODOLOGIES FOR (INTELLIGENT) AGENTS

Agents are manifold in the real world. The (multi-) agent paradigm provides a means to characterise interactions between autonomous agents and their environment. Agents (either human or automated) are responsible for these processes, where each agent has its own environment, consisting of other agents and a material world. Agents are able to communicate with each other, can co-operate to jointly perform tasks, interact with the world (observe and/or act), and perform specific tasks. Some agents interact directly with humans, other agents interact with automated agents only (Kautz, Selman and Coen, 1994). In the near future the co-operation among agents and humans is expected to have impact on social conventions in society (Norman, 1994).

During the past years extensive research has been conducted in the field of multi-agent systems. Different notions of agency have been proposed (e.g., Nwana, 1996; Wooldridge and Jennings, 1995; Shoham, 1993). One notion of agents in which weak agency is distinguished from strong agency has been proposed by Wooldridge and Jennings (1995): weak agency is characterised by autonomy, social ability, reactiveness, and pro-activeness. In contrast the notion of strong agency is based on the characteristics of mentalistic and intentional notions (related to the notion of intentional stance by Dennett, 1987).

The characteristics of weak agency defined by Wooldridge and Jennings (1995) provide a means to reflect on the tasks an agent needs to be able to perform. Pro-activeness and autonomy are related to an agent's ability to reason about its own processes, goals and plans. Reactivity and social ability are related to the ability to interact with the material world and to communicate with other agents. The ability to communicate and co-operate with other agents and to interact with the material world often relies on an agent's ability to acquire and maintain its own knowledge of the world and other agents.

Agents, and multi-agent systems, are currently widely studied. Recent publications on agents include literature on software agents, e.g., see (Bradshaw, 1997), and literature on agent technology, e.g., see (Jennings and Wooldridge, 1998). Information brokering and information gathering agents (Levy, Sagiv and Srivastava, 1994; Sycara and Zeng, 1996; Knoblock and

Ambite, 1997; Jonker and Treur, 1998), a special kind of agent, play an important role in exploiting agent technology in the context of the Internet. Information gathering agents are sometimes developed ‘ad hoc’, or can be developed in a structured manner.

Methodologies and tools for the development of multi-agent systems are currently mainstream, e.g., AgentBuilder (Reticular Systems, 1999), D'agents/AgentTCL (Gray, Kotz, Cybenko, and Rus, 1997), ZEUS (Nwana, Ndumu, Lyndon, and Collis, 1999) and Tryllian's Agent Development Kit (Tryllian, 2001). All of these approaches commit to a specific operational description of agents, and usually also commit to a specific conceptual description of their agents.

The agent metaphor offers a means to model situations with distributive activity on a conceptual level (e.g., Jennings, 2000). Multi-agent systems have been proposed to model collaborative tasks such as design (Edmonds, Candy, Jones and Soufi, 1994; Vanwelkenhuysen and Mizoguchi, 1995; Dunskus, Grecu, Brown and Berker, 1995; Berker and Brown, 1996).

2.2 RESEARCH ON ADAPTIVE AGENTS

Agents that adapt to their environment are one of the areas of research in multi-agent systems. One application of adaptive agents entails personification, for example an information gathering agent may maintain a profile of another agent, and adapt this profile on the basis of interaction with that agent (e.g., as also encountered in negotiation settings (Bui, Kieronska and Venkatesh, 1996)). Note that in this example personification may be aimed at personalising an agent’s representation of a human user (e.g., see Wells and Wolfers, 2000; Soltysiak and Crabtree, 1998), as well as the profile of an agent.

Sometimes reactive behaviour of an agent is dubbed 'adaptive behaviour', e.g. by (Rus, Gray and Kotz, 1996) where an agent is, e.g., capable of abandoning a previous goal or plan, and adopting a new goal or plan which better fits the current situation of the agent. In addition, learning techniques are often used for adaptive agents, e.g. as described by (Reffat and Gero, 2000; Grefenstette, 1992). Yet another perspective on adaptive agents is that the population of agents may change in time, this is more of an adaptive agent architecture (Maturana, Shen and Norrie, 1999).

2.3 RESEARCH ON SELF-MODIFYING AGENTS

The agent metaphor can also be used to develop agents that are able to dynamically design and create new agents, or to dynamically modify existing agents. For example, Internet agents that are capable of dynamically creating new agents to assist them in information gathering, or agents that are capable of creating interface agents tuned to specific users, are agents of this type. Also agents (including users) may be given the ability to influence the agent which re-designs itself or part of the multi-agent system: requirements, partial design object descriptions and process objectives can be communicated and negotiated.

Literature which partially addresses the topics ‘re-design of compositional systems’ and ‘self-modification’ includes approaches based on genetic programming and parametric design, approaches based on meta-level architectures, and approaches based on mind-matter interactions. These approaches are described below.

Approaches based on *genetic programming & parametric design*. Most of the research in the area of dynamic agent creation is based on a genetic programming approach; e.g., (Cetnarowicz, Kisiel-Dorohinicki, and Nawarecki, 1996; Numaoka, 1996): design descriptions of agents are combined to evolve to a most suitable design description of an agent, according to some criteria. Modifying problem solving methods by means of parametric design is an approach taken by (Teije, Harmelen, Schreiber and Wielinga, 1998) in which parameters of an otherwise fixed problem solving method are given appropriate values. In the genetic programming & parametric design approach a modified system is acquired by changing parameters of the system according to the modifications in the design description.

Approaches based on *meta-level architectures*. A reflective approach, in which an agent reasons about its own representation and re-designs this representation, is taken by e.g., (Schubert, 1997; Stroulia and Goal, 1994*a*; 1994*b*). A model-based approach to self-configuration of autonomous (spacecraft) systems is taken by (Williams and Nayak, 1996). Adapting a fixed task structure for different situations has been described by (Stroulia and Goel, 1994*a*). Reflecting on a problem solving method has been described by (Harmelen, Wielinga, Bredeweg, Schreiber, Karbach, Reinders, Voß, Akkermans, Bartsch-Spörl, and Vinkhuyzen, 1992; Teije and Harmelen, 1996). Modification of control knowledge in a problem solving method on the basis of inspection of the performance of the control knowledge is described by (Straatman, 1997).

Research on (distributed) design (Grecu and Brown, 1996; Cross, Christiaan, and Dorst, 1996; Campbell, Cagan and Kotovsky, 1998; McAlinden, Florida-James, Chao, Norman, Hills and Smith, 1998) does not include explicit representations for *reflective reasoning*. Being able to reason about, or even from, the viewpoint of another agent is a means with which, e.g., conflicts can be prevented. In the literature on reflection such as (Weyhrauch, 1980; Davis, 1980; Maes and Nardi, 1998; Attardi and Simi, 1994; Clancey and Bock, 1988) a restricted number of types of reflective reasoning are modelled. Non-trivial combinations of different types of reflective reasoning, however, have not been studied extensively. In literature (Fisher and Wooldridge, 1993; Wooldridge and Jennings, 1995; Cimattie and Serafini, 1995; Wagner, 1996) on multi-agent systems, most often the types of reflective reasoning agents are capable of performing is limited. For example, in the literature mentioned reflective reasoning about communication is not explicitly modelled, but is of importance (Brazier and Treur, 1999; Brazier, Moshkina and Wijngaards, 2001).

Approaches based on *mind-matter interactions*. Self-modification entails the re-design of an agent's own description on the basis of a relationship between the actual 'physical' description of oneself and the dynamic flow of information within one's thought processes (Jonker and Treur, 1997). The emphasis is that to create new agents, an existing agent must be capable of re-designing itself on the basis of a model for design and then be capable of bringing its new description to life by performing actions modifying the material world. The integration of re-design on a conceptual and logical level (the mind aspect), and run-time modification of the system at the implementation level by performing material actions (the matter aspect) is of importance.

3. A Knowledge-Level model of a Self-modifying Agent

In this section a knowledge-level model of a self-modifying agent is presented by first making explicit assumptions and requirements in Section 3.1, and then presenting a generic (process) model of a self-modifying agent in Section 3.2.

3.1 ASSUMPTIONS AND REQUIREMENTS

The feasibility of designing self-modifying agents depends on the assumptions and requirements imposed on self-modifying agents. The most important underlying assumptions and requirements for self-modifying agents are described in this section. The following *assumptions* underlie a self-modifying agent:

1. An agent has a compositional structure
2. Any agent may be given 'self-modification abilities'
3. Re-usable parts of an agent can be identified.
4. Re-usable parts of an agent can be combined.
5. A self-modification process is a re-design process.
6. Knowledge can be found enabling an agent to determine that it is not doing well in performing a task.
7. Knowledge can be found enabling an agent to re-design itself.
8. A language can be found enabling an agent to express self-modification requirements.

9. A language can be found enabling an agent to express self-modification process objectives.

The first assumption states that a compositional structure is used in describing all agents, including self-modifying agents. A compositional structure of an artefact factilitates the re-design of that artefact, as described in (Brazier, Jonker, Treur and Wijngaards, 2000b). The second assumption is that each agent may acquire an ability for self-modification, irrespective of its agent's specific tasks. This assumption enforces a generalisation of the model for self-modification. The third and fourth assumption make explicit that building blocks can be distinguished and used to compose agents. Building blocks may be general (cf. design patterns by Gamma, Helm, Johnson and Vlissides (1994)) or detailed / domain specific (cf. problem-solving methods by (Gomez Perez and Benjamins, 1999)).

The fifth assumption states that a process of self-modification is the same as a process of 'agent re-design'. This enables the re-use of models, theories, knowledge, and concepts used in (re-)design processes. The generic model of design, extended for re-design of compositional structures (Brazier, Jonker, Treur and Wijngaards, 2001) is employed as a model of self-modification.

Assumptions six to nine refer to specific knowledge and languages which are needed by an agent to re-design itself, and control the re-design process, which forms the model for a self-modification process.

The following *requirements* are posed on a self-modifying agent:

1. Self-modification is not limited to an agent's specific tasks, but also all other processes within an agent, excluding the self-modification process.
2. A self-modifying agent needs to monitor its own behaviour, and be able to decide when its behaviour is not appropriate.
3. A self-modifying agent needs to express the nature of a problem and the nature of required behaviour.
4. A self-modifying agent knows how to effectuate a modification.

The first requirement expresses that, e.g., an agent's ability to communicate with other agents using a specific ACL (agent communication language) may be modified, e.g., to include another ACL. To simplify the self-modification process, no recursion in self-modification of the self-modification process is allowed. The second requirement expresses that a self-modifying agent is able to reflect on its own behaviour, including its progress in solving problems. The third requirement expresses that a self-modifying agent is able to reflect on what it has done, what it would like to do, its ability to adapt, its limitations, and strategies to acquire new knowledge. The second and third requirements are based on the notion of reflection in action (Schön, 1983).

The fourth requirement states that the agent is able to effectuate a modification of itself. It doesn't matter whether the agent iself is capable of effectuating modification, or the agent platform and/or middleware provides facilities for effectuating a modification.

3.2 MODEL OF A SELF-MODIFYING AGENT

The basis for the model of a self-modifying agent is an existing knowledge-level DESIRE model of a design agent for single-agent design (Brazier, Jonker, Treur and Wijngaards, 2001), enhanced with a component which manages co-operation between agents for project co-ordination (Brazier, Jonker and Treur, 1996).

DESIRE is a formal knowledge-level modelling and specification framework for knowledge-intensive (multi-agent) systems (Brazier, Dunin-Keplicz, Jennings and Treur, 1995, 1997; Brazier, Jonker and Treur, 1998). Both conceptual models and detailed formal specifications are supported by the framework. The compositional nature of the models, and the separation between processes and knowledge makes it possible to build knowledge intensive systems from reusable components. Automated prototype generation on the basis of detailed formal specifications facilitates verification and validation of knowledge intensive systems.

This model distinguishes seven main processes within an agent, as depicted in Figure 1 below. This architecture models an agent that:

1. reasons about its own processes (component Own Process Control),
2. communicates with other agents (component Agent Interaction Management),
3. maintains information about other agents (component Maintenance of Agent Information),
4. interacts with the external world (component World Interaction Management),
5. maintains information about the external world (component Maintenance of World Information),
6. participates in project co-ordination (component Co-operation Management) and
7. designs an artefact (within component Agent Specific Tasks is a component Design).

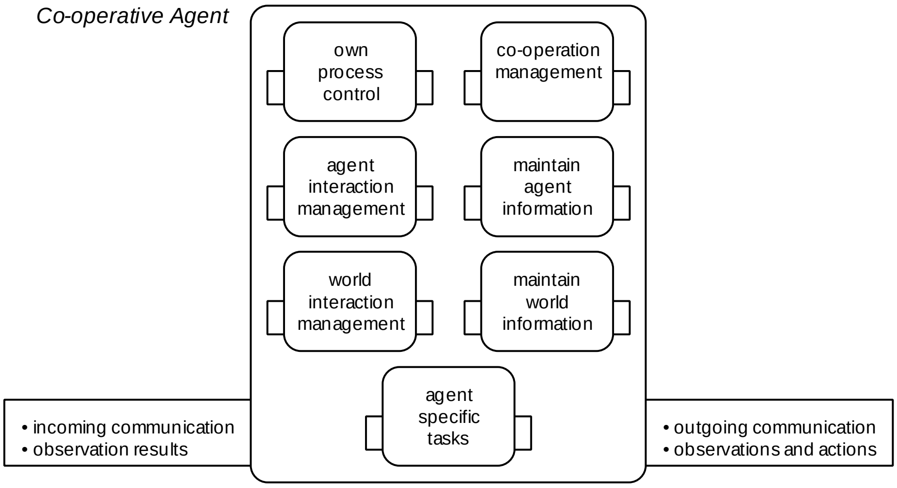

Figure 1. Process abstraction levels for a generic co-operative agent.

The self-modification process is modelled as a re-design process. The model of a self-modifying agent deviates from the model described in Figure 1 in the location of its self-modification process. The self-modification process is *not* part of the agent's specific tasks and thus not part of the agent's specific task in this conceptual description. The self-modification process (i.e., the design process in disguise) is located within the component Own Process Control.

The component Own process Control needs knowledge not only about the agent's own characteristics and strategies for performing its agent's specific tasks, but also about interacting with a self-modification process, and providing a model of the agent itself to the self-modification process. In particular, a language needs to be available to express self-modification requirements and self-modification process objectives, akin to design requirements and design process objectives (Brazier, Langen, Ruttkay and Treur, 1994).

Alternatively, the modification process by which an agent is modified may be external to the agent as in an agent factory (Brazier and Wijngaards, 2001). In this situation, the agent may initiate a modification and be modified. The results of an external modification process should be the same as the results of an internal self-modification process.

The model of a self-modifying agent is generic in both the domain and the task of the self-modifying agent. The agent's specific tasks are not pre-defined, and may be any combination of tasks including design tasks, and diagnosis tasks. If required, some of the components within the model can be left out, e.g. if the agent never directly interacts with the external world the component World Interaction Management may be omitted, but the component

Maintain World information is usually retained as the agent may receive information about the world via communication.

The self-modification process is shown in Figure 2. In this model an initial self-modification problem statement is expressed as a set of initial self-modification requirements and requirement qualifications. *Requirements* impose conditions and restrictions on the structure, functionality and behaviour of the *agent itself* for which a structural description is to be modified during self-modification. *Qualifications* of requirements are qualitative expressions of the extent to which (individual or groups of) self-modification requirements are considered to be hard or preferred, either in isolation or in relation to other (individual or groups of) self-modification requirements. At any one point in time during design, the self-modification process focuses on a specific subset of the set of self-modification requirements. This subset of requirements plays a central role; the design process is (temporarily) committed to the current self-modification requirement qualification set: the aim of generating a design object description is to satisfy these requirements.

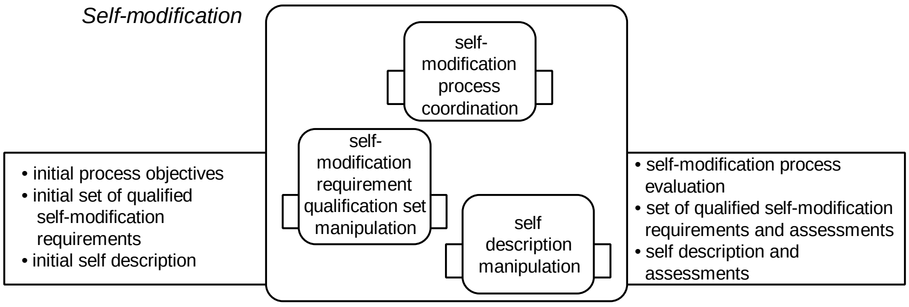
Self-modification
self-modification process coordination
self-modification requirement qualification set manipulation
self description manipulation
• initial process objectives
• initial set of qualified self-modification requirements
• initial self description
• self-modification process evaluation
• set of qualified self-modification requirements and assessments
• self description and assessments

Figure 2. Process abstraction levels for the process of self-modification.

Figure 2 describes one level of process abstraction for the process of self-modification. The left hand side describes the input information to the self-modification process; the right hand side describes the output information. The self-modification process is shown to be composed of three sub-processes: self-modification process co-ordination, self-modification requirement qualification set manipulation, and self description manipulation. The process Self-Modification Process Co-ordination co-ordinates the design process by issuing information related to overall self-modification strategies on the basis of progress reports of the manipulation components and given self-modification process objectives. The process Self-Modification Requirement Qualification Set Manipulation manipulates sets of requirements, on the basis of an overall self-modification strategy, information from Self Description Manipulation, and given sets of qualified requirements. The process Self

Description Manipulation manipulates descriptions of an agent, on the basis of an overall self-modification strategy, information from Self-Modification Requirement Qualification Set Manipulation, and given self descriptions.

3.3 EXAMPLE

To illustrate the use of the model of a self-modifying agent described in the previous sections, an example is given of an information retrieval agent. In this example self-modification entails adding additional functionality to the agent.

The information retrieval agent's task is to find information about a specific topic. Its knowledge for finding information on the web and relating information to queries is located in the component Agent Specific Task.

For a specific query, the information retrieval agent has arrived at a very informative website. One of the most promising links on that webpage is in a format that is unknown to the information retrieval agent: it doesn't know how to handle ftp-sites, only html-sites.

The information retrieval agent starts a self-modification process with the intent of obtaining the ability to interact with ftp-sites. Its self-modification process consults, e.g., an external library containing code and knowledge fragments, and re-designs the information retrieval agent to the extent that it can interact with an ftp-site, and resolve references to ftp-sites.

Alternatively, the information retrieval agent may have used an external service for its modification.

3.4 SELF-MODIFYING AGENTS AND CREATIVITY

A self-modifying agent may be creative in two ways: theself-modification process itself may be creative (self modification is namely a re-design task), or self-modification may result in a more creative agent with respect to an agent’s specific task. Creativity within the process of self-modification is directly related to creativity in design processes (e.g., Schön, 1983; Finke, Ward and Smith, 1992; Edmonds and Candy, 1997; Gero, 1996; Lawson, 1997). The application of the self-modification process to influence creativity within other processes within an agent, e.g. an agent's specific task, may result in more creative results.

The self-modification process described in this paper includes four of the five stages distinguished in the five-stage model introduced by Kneller and described by Lawson (1997), shown in Figure 3: stage 4 is the exception. Unconscious effort is not easily defined for an automated agent.

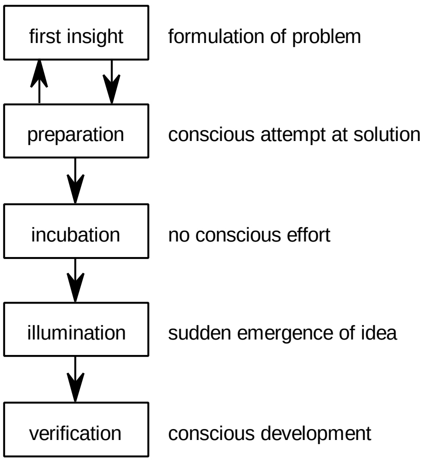

Figure 3. Five-stage model of the creative process (taken from Lawson, 1997).

Gero's model for creative design (1996) is based on the view that creativity results from a discrepancy between expectations and unexpected results. If the unexpected results can be understood, then they are considered to be a creative solution. When, however, the unexpected results cannot be understood, these results are rejected as faulty. A self-modification process may have side effects that had not been anticipated. Influences on the creativity of an agent with respect to its own specific task may be purposefully sought by re-design, without knowing the results. In both cases, how well the results are understood will depend on the situation, e.g. who is responsible for monitoring an agent's behaviour – a human being or another automated agent.

4. Designing a self-modifying agent

The process of designing a dynamic entity is described in this section. A number of issues related to the feasibility of designing self-modiying agents are described in Section 4.1. The design of the information retrieval agent, introduced in Section 3.3, is described in Section 4.2. The current status of a prototype agent factory is described in Section 4.3.

4.1 FEASIBILITY

The feasibility of designing self-modifying agents hinges on a number of issues. These issues can be categorised into (1) issues related to design processes, (2) issues related to the design of agents, (3) issues related to self-modification of agents, and (4) issues related to the design of self-modifying agents.

First of all, issues related to a design process in general play a role. As described in Section 3.2, a generic model of a design process (Brazier, Langen, Ruttkay and Treur, 1994) forms the basis for the process of self-modification. This generic model of design has associated a logical theory of design (Brazier, Langen and Treur, 1996). Within this model and theory of design, design strategies (Brazier, Langen and Treur, 1998) and design rationale can be modelled (Brazier, Langen and Treur, 1997), and conflict management can be explicitly described (Brazier, Langen and Treur, 1995).

Second, a number of issues are related to the design of agents. The characteristics that play a role are described in (Brazier, Jonker and Treur, 1998). A generic model of an agent (Brazier, Jonker and Treur, 2000), based on a notion of weak agency proposed by Wooldridge and Jenings (1995): weak agency is characterised by autonomy, social ability, reactiveness, and pro-activeness. In contrast the notion of strong agency is based on the characteristics of mentalistic and intentional notions (related to the notion of intentional stance by Dennet (1987)). Models of co-operation and co-ordination between agents have been proposed (Brazier, Jonker and Treur, 1996).

Third, issues related to self-modification of agents are discussed in Section 3.1.

Fourth, a number of issues are related to the design of a self-modifying agent. Although agents have a compositional structure, it is not trivial to define larger building blocks, from which agents can be configured. Building blocks do not only need to be identified, but also described (in terms of structure, function, and behaviour) such that a design process can use building blocks in the design of an agent.

Related to these issues is the feasibility of the agent factory. The agent factory is basically a modification process external to agents, capable of achieving similar modifications as an agent's self-modification process. The feasibility of the agent factory hinges on the following five issues (Brazier and Wijngaards, 2001): (1) agents have a compositional structure, (2) re-usable parts of agents can be identified, (3) two levels of descriptions are used: conceptual and operational, (4) properties and knowledge of properties are available, and (5) no commitments are made to specific languages and/or ontologies.

In short, when designing self-modifying agents,
ontologies are needed to express

- properties of agents (structural, functional, and behavioural)
- properties of tasks (structural, functional, and behavioural)
- descriptions of agents
- design strategies

knowledge is needed to

- relate description of agent to properties of agent
- relate required properties to required properties
- relate required properties to structure modifications of an agent
- understand tasks, the ontologies, knowledge, and control involved
- make strategic decisions within the agent's own process control
- determine which strategies to deploy within an agent
- resolve conflicts within an agent

and strategies are needed to

- guide the overall design process
- alternate between viewpoints on design requirements and agent descriptions
- prioritize possible strategies for self-modification.

4.2 EXAMPLE

The process of designing a self-modifying information retrieval agent, introduced in Section 3.3, is used as an illustration. The self-modifying information retrieval agent was designed on the basis of the following requirements:

1. The agent is able to find information on web-pages.
2. The agent is able to adapt itself.
3. The agent is able to communicate with other agents on queries and query results.

These requirements state that the information retrieval agent is able to find information on the web. To design such an agent, the design process needs specific knowledge of

- designing an information retrieval agent
- self-modification capabilities
- reflective capabilities
- modification of information retrieval agents.

4.3 AGENT FACTORY: CURRENT STATUS

Current research of the IIDS group focusses on the design of an agent operating system, its services, and applications. One of the intended services is the automated creation and modification of agents via an agent factory. The agent factory is used to illustrate the modification process that take place inside a self-modifying agent. As stated earlier in this paper, the location of the self-modification process may be internal, or external, to an agent which wishes to be modified.

The agent factory is a continuation of almost a decade of research in AI and Design, applied to multi-agent systems. The generic model of design (Brazier, Langen, Ruttkay and Treur, 1994) has been specialised for the re-design of compositional systems: agents (Brazier, Jonker, Treur and Wijngaards, 2000b). This work included the specification and implementation of a design agent, capable of re-designing agents (Brazier, Jonker, Treur and Wijngaards, 2001). The aforementioned approached used a start-from-first-principles design process, while the agent factory is based on building-blocks and aims at simplifying the re-design process by moving towards a configuration-based design process.

A number of prototypes of an agent factory for information retrieval agents, with limited functionality, have been designed and implemented. One of the current problems is in finding the right heuristics to be used by a self-modifying agent. This problem is clearly related to enhancing creativity.

5. Discussion

The main purpose of this study was to understand the process of designing self-modifying agents and the role of creativity. The knowledge-level model of a self-modifying agent presented in this paper is based on an existing model of re-design. The self modification process is a re-design process. The task of the agent may be anything: diagnosis, information retrieval, scheduler. An interesting application would be automated re-design of *design agents* (Brazier, Jonker, Treur and Wijngaards, 2001): self-modifying design agents. This would entail creativity both at the level of agent configuration and at the level of an agent's specific task, namely design.

One of the key problems is identifying strategies. Strategies for a self-modifying agent regarding decisions concerning when to adapt itself, for what reason, etc. And strategies for designing a self-modifying agent: what knowledge is needed inside such an agent to be able to function well? These strategies determine how successful or creative a self-modifying agent can be.

Note that the location of the modification process may be internal to the agent (i.e., a self-modification process) or external to the agent (e.g., via an agent factory). In both cases the agent itself may initiate a modification to itself, and requires some reflective capabilities.

Acknowledgements

This work was supported by NLnet Foundation, http://www.nlnet.nl. The authors are grateful to Hidde Boonstra, David Mobach, Oscar van Scholten, and Sander van Splunter for their contribution to the work on the agent factory.

References

Attardi, G. and Simi, M.: 1994, Proofs in Context, *in* L. Fribourg and F. Turini (eds), *Logic Program Synthesis and Transformation-Meta-Programming in Logic, Proceedings of the Fourth International Workshop on Meta-Programming in Logic, META'94*, Springer Verlag, Lecture Notes in Computer Science, **883**, pp. 410-424.

Berker, I. and Brown, D. C.: 1996, Conflicts and Negotiation in Single Function Agent Based Design Systems, *in* D. C. Brown, S. E. Landes and C. J. PETRIE (eds), *Concurrent Engineering: Research and Applications*, *Journal*, Special Issue: Multi Agent Systems in Concurrent Engineering, Technomic Publishing Inc., **4**(1), 17-33.

Bradshaw, J. M. (ed): 1997, *Software Agents*, AAAI Press / MIT Press.

Brazier, F. M. T. and Treur, J.: 1999, Compositional Modelling of Reflective Agents, *International Journal of Human-Computer Studies*, **50**, 407-431.

Brazier, F. M. T. and Wijngaards, N. J. E.: 2001, Automated servicing of agents. D. Kudenko & E. Alonso (eds), *Proceedings of the AISB-01 Symposium on Adaptive Agents and Multi-agent systems, at the Agents & Cognition AISB-01 conference*, the society for the study of artificial intelligence and the simulation of behaviour, ISBN 1.902956.17.0, pp. 54 - 64.

Brazier, F. M. T., Dunin-Keplicz B. M., Jennings, N.R. and Treur, J.:1995,1997, Formal specification of Multi-Agent Systems: a real-world case, in: V. Lesser (ed), *Proceedings of the First International Conference on Multi-Agent Systems, ICMAS'95*. Cambridge MA: MIT Press, pp. 25-32. Extended version: 1997, *in*: M. Huhns and M. Singh (eds), *International Journal of Co-operative Information Systems*, special issue on Formal Methods in Co-operative Information Systems: Multi-Agent Systems, **6**, 67-94.

Brazier, F. M. T., Jonker, C. M. and Treur J.: 1996, Modelling project coordination in a multi-agent framework, in *Proceedings Fifth Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE'96*, Los Alamitos: IEEE Computer Society Press, pp. 148-155.

Brazier, F. M. T., Jonker, C. M. and Treur J.: 1998, Principles of Compositional Multi-agent System Development, *in* J. Cuena (ed), *Proceedings of the 15th IFIP World Computer Congress, WCC'98, Conference on Information Technology and Knowledge Systems, IT&KNOWS'98*, pp. 347-360.

Brazier, F. M. T., Jonker, C. M. and Treur, J.: 2000, Compositional Design and Reuse of a Generic Agent Model, *Applied Artificial Intelligence Journal*, **14**, 491-538.

Brazier, F. M. T., Jonker, C. M., Treur, J. and Wijngaards, N. J. E.: 2000b, Deliberate Evolution in Multi-Agent Systems, *in* J. Gero (ed.), *Proceedings of the Sixth International Conference on AI in Design, AID'2000*. Kluwer Academic Publishers, 2000, pp 633-650.

Brazier, F. M. T., Jonker, C. M., Treur, J. and Wijngaards, N. J. E.: 2001, Compositional Design of a Generic Design Agent, *Design Studies journal*, **22**, 439-471..

Brazier, F. M. T., Langen, P. H. G. van, Ruttkay, Zs. and Treur J: 1994, On formal specification of design tasks, *in* J. S. Gero and F. Sudweeks (eds), *Proceedings Artificial Intelligence in Design (AID'94)*, Dordrecht: Kluwer Academic Publishers, pp. 535-552.

Brazier, F. M. T., Langen, P. H. G. van and Treur, J.: 1995, Modelling conflict management in design: an explicit approach, *Artificial Intelligence for Engineering Design, Analysis and Manifacturing, (AIEDAM), in* I. F. C. Smith (ed.), Special Issue on Conflict Management in Design, **9**(4), 353-366.

Brazier, F. M. T., Langen, P. H. G. van and Treur J.: 1996, A logical theory of design, *in* J. S. Gero (ed.), *Advances in Formal Design Methods for CAD, Proc. of the Second*

International Workshop on Formal Methods in Design, Chapman & Hall, New York, pp. 243-266.

Brazier, F. M. T., Langen, P. H. G. van and Treur, J.: 1997, A compositional approach to modelling design rationale, *Artificial Intelligence for Engineering Design, Analysis and Manifacturing, (AIEDAM), in* P. W. H. Chung and R. Banares-Alcantara (eds), Special Issue on Representing and Using Design Rationale, **11**(2), 125-139.

Brazier, F. M. T., Langen, P. H. G. van and Treur, J.: 1998, Strategic Knowledge in Compositional Design Models, *in* J. S. Gero and F. Sudweeks (eds), *Proceedings of the Fifth International Conference on Artificial Intelligence in Design, AID'98,* Kluwer Academic Publishers, Dordrecht, pp. 129-147.

Brazier, F. M. T., Moshkina, L. V. and Wijngaards, N. J. E.: 2001, Knowledge level model of an individual designer agent in collaborative distributed design, *Journal of Artificial Intelligence in Engineering*, To appear.

Bui, H. H., Kieronska, D. and Venkatesh, S.: 1996, Learning other agents' preferences in multiagent negotiation, *in Proceedings of the National Conference on Artificial Intelligence (AAAI-96)*, pp. 114-119.

Campbell, M. I, Cagan, J. and Kotovsky, K.: 1998, A-Design: theory and implementation of an adaptive, agent-based method of conceptual design, *in* J.S. Gero and F. Sudweeks (eds), *Artificial Intelligence in Design '98 (AID '98)*, Dordrecht: Kluwer Academic Publishers, pp. 579-598.

Cetnarowicz, K., Kisiel-Dorohinicki, M. and Nawarecki, E.: 1996, The Application of Evolution Process in Multi-Agent World to the Prediction System, *in* M. Tokoro (ed), *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS'96)*, MIT/AAAI Press, Menlo Park CA, pp. 26-32.

Cimatti, A. and Serafini L.: 1995, Multi-agent Reasoning with Belief Contexts II: Elaboration Tolerance, *in* V. Lesser (ed), *Proceedings of the First International Conference on Multi-Agent Systems, ICMAS-95,* MIT Press, pp. 57-64.

Clancey, W. J. and Bock, C.: 1988, Representing control knowledge as abstract tasks and metarules, *in* L. Bolc and M. J. Coombs (eds), *Computer Expert Systems*, Heidelberg: Springer-Verlag, pp. 1-77.

Cross, N., Christiaans, H. and Dorst, K. (eds): 1996, Analysing Design Activity, John Wiley & Sons Ltd, Chichester, England.

Davis, R.: 1980, Metarules: reasoning about control, *Artificial Intelligence*, **15**, 179-222.

Dennett, D. C.: 1987, *The Intentional Stance*, MIT Press, Cambridge.

Dunskus, B. V., Grecu, D. L., Brown, D. C. and Berker, I.: 1995, Using Single Function Agents to Investigate Conflict, *Artificial Intelligence in Engineering Design and Manufacturing (AIEDAM)*, Special Issue: Conflict Management in Design, **9**(4), 299-312.

Edmonds, E. A. and Candy, L.: 1997, Supporting the Creative User: A Criteria-based Approach to Interactive Design, *Design Studies*, **18**(2), 185-194.

Edmonds, E. A., Candy, L., Jones, R. and Soufi, B.: 1994, Support for Collaborative Design: Agents and Emergence, *Communications of the ACM*, **37**(7), 41-47.

Finke, R. A., Ward, T. B. and Smith, S. M.: 1992, Creative Cognition, theory, research, and applications, MIT press, Cambridge, MA.

Fisher, M. and Wooldridge, M.: 1993, Specifying and Verifying Distributed Intelligent Systems, *in* M. Filqueiras and L. Damas (eds), *Progress in AI. Proc. EPAI'93*, Springer Verlag, Lecture Notes in AI, **727**, pp. 13-28.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J.: 1994, *Design Patterns: Elements of reusable object-oriented software*, Addison Wesley Longman, Reading, Massachusetts.

Gero, J. S.: 1996. Creativity, emergence and evolution in design: concepts and framework, *Knowledge-Based Systems*, **9**(7), 435-448.

Gomez Perez, A. and Benjamins, V. R.: 1999, Applications of Ontologies and Problem-Solving Methods, *AI-Magazine*, **20**(1), 119-122.

Gray, R., Kotz, D., Cybenko, G. E. and Rus, D.: 1997, Agent Tcl, *in* W. Cockayne and M. Zyda (eds), *Mobile Agents: Explanations and Examples*, Manning Publishing, 1997.

Grecu, D. L. and Brown, D. C.: 1996, Learning by single function agents during spring design, *in* J. S. Gero and F. Sudweeks (eds), *Artificial Intelligence in Design '96 (AID '96),* Dordrecht: Kluwer Academic Publishers, pp. 409-428.

Grefenstette, J.: 1992, The evolution of strategies for multiagent environments, *Adaptive Behavior*, **1**(1). 65-90.

Harmelen, F. van, Wielinga, B., Bredeweg, B., Schreiber, G., Karbach, W., Reinders, M., Voß, A., Akkermans, H., Bartsch-Spörl, B. and Vinkhuyzen, E.: 1992, Knowledge-level reflection, *in* B. Le Pape and L. Steels (eds), *Enhancing the knowledge engineering process - contributions from ESPRIT*, Elsevier Science, Amsterdam, pp. 175-204.

Jennings, N. R. and Wooldridge, M. J. (eds): 1998, *Agent Technology; Foundations, Application, and Markets*, Springer Verlag.

Jennings, N. R.: 2000, On agent-based software engineering, *Artificial Intelligence*, **117**, 277-296.

Jonker, C. M. and Treur, J.: 1997, Modelling an Agent's Mind and Matter, *in* M. Boman and W. van de Velde (eds), *Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'97*, Lecture Notes in AI, **1237**, Springer Verlag, pp. 210-233.

Jonker, C. M., and Treur, J.: 1998, A Generic Architecture for Broker Agents, *in* Nwanam H.S., and Ndumu, D.T. (eds), *Proceedings of the Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'98),* The Practical Application Company Ltd, pp. 623-624.

Kautz, H. A., Selman, B. and Coen, M.: 1994, Bottom-Up Design of Software Agents, *Communications of the ACM*, **37**(7), 143-146.

Knoblock, C. A. and Ambite, J. L.: 1997, Agents for Information Gathering, *in* (Bradshaw, 1997), pp. 347-373.

Lawson, B.: 1997, *How designers think: the design process demystified*, 3rd edition, Architectural press.

Levy, A. Y., Sagiv, Y. and Srivastava, D.: 1994, Towards efficient information gathering agents, *in Software Agents, proceedings of the AAAI 1994 spring symposium*, AAAI Press, pp. 64-70.

Maes, P. and Nardi, D. (editors): 1998, *Meta-level architectures and reflection*, Elsevier Science Publishers.

Maturana F., Shen, W. and Norrie, D. H.: 1999, MetaMorph: An Adaptive Agent-Based Architecture for Intelligent Manufacturing, *International Journal of Production Research*, **37**(10), 2159-2174.

McAlinden, L. P., Florida-James, B. O., Chao, K-M., Norman, P. W., Hills, W. and Smith, P.: 1998, Information and knowledge sharing for distributed design agents, *in* J. S. Gero and F. Sudweeks (eds), *Artificial Intelligence in Design '98 (AID '98),* Dordrecht: Kluwer Academic Publishers, pp. 537-556.

Mozer, M. C.: 1999, An intelligent environment must be adaptive, *IEEE Intelligent Systems and their Applications*, **14**(2), 11-13.

Norman, D. A.: 1994, How Might People Interact with Agents, *Communications of the ACM*, **37**(7), 68-71.

Numaoka, C.: 1996, Bacterial Evolution Algorithm for Rapid Adaptation, *in* W. Van de Velde and J. W. Perram (eds), *Proceedings of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96)*, Lecture Notes in Artificial Intelligence, **1038**, Springer Verlag, pp. 139-148.

Nwana, H. S.: 1996, Software agents: an overview, *The Knowledge Engineering Review*, Cambridge University Press, **11**(3), 205-244.

Nwana, H., Ndumu, D., Lyndon, L., and Collis, J.: 1999, ZEUS: A Tookit and Approach for Building Distributed Multi-agent System, *in Proceedings of the Third International Conference on Autonomous Agents (Autonomous Agents'99)*, pp. 360-361.

Reffat, R. M. and Gero, J. S.: 2000, Computational Situated Learning in Design, in: J. S. Gero (ed), *Artificial Intelligence in Design '00*, Kluwer Academic Publishers, Dordrecht, pp. 589-610.

Reticular Systems Inc.: 1999, AgentBuilder: An integrated toolkit for constructing intelligent software agents, *White Paper*, http://www.agentbuilder.com.

Rus, D., Gray, R. and Kotz, D.: 1996, Autonomous and Adaptive Agents that Gather Information, *in AAAI'96 International Workshop on Intelligent Adaptive Agents*, pp. 107--116.

Schön, D. A.: 1983, *The Reflective Practitioner: how professionals think in action*, Basic Books Inc.

Schubert, F.: 1997, A reflective architecture for an adaptable object-oriented operating system based on C++, *in Proceedings of the ECOOP'97 workshop on object-orientation and operating systems*, Springer Verlag, Lecture notes in Computer Science, **1357**.

Shoham, Y.: 1993, Agent-oriented programming, *Artificial Intelligence*, **60**, 51-92.

Soltysiak, S. and Crabtree, B.: 1998, Knowing Me, Knowing You: Practical Issues in the Personalisation of Agent Technology, *in Proceedings of the third international conference on the practical applications of intelligent agents and multi-agent technology* (PAAM98), London.

Straatman, R.: 1997, Kids for Kads, *in* E. Plaza and R. Benjamins (eds), *Proceedings of the 10th European Workshop on Knowledge Acquisition, Modelling, and Management (EKAW'97)*, Sant Feliu de Guixols, Catalania, Lecture Notes in Artificial Intelligence, **1319**, Springer Verlag, pp. 371-376.

Stroulia, E. and Goel, A. K.: 1994a. Reflective, Self-Adaptive Problem Solvers, *in* L. Steels, G. Schreiber and W. van de Velde (eds), *A future for knowledge acquisition, proceedings of the 1994 European Conference on Knowledge Acquisition (EKAW'94)*, Lecture Notes in Artificial Intelligence, **867**, Springer-Verlag, pp. 394-413.

Stroulia, E. and Goel, A. K.: 1994b, Learning Problem-Solving Concepts by Reflecting on Problem Solving, *in* Bergadano and L. De Raedt (eds), *Proceedings of ECML-94*, Springer-Verlag, pp. 287-306.

Sycara, K. and Zeng, D.: 1996, Multi-agent integration of information gathering and decision support, *in* Wahlster, W. (ed), *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI'96)*, Wiley and Sons, pp. 549-553.

Teije, A. ten and Harmelen, F. van: 1996, Using reflection techniques for flexible problem solving, *Future Generation Computer Systems*, **12**, special issue Reflection and Meta-level AI Architectures, 217-234.

Teije, A. ten, Harmelen, F. van, Schreiber, A. Th. and Wielinga, B. J.: 1998, Construction of problem-solving methods as parametric design, *International Journal of Human-Computer Studies*, Special issue on problem-solving methods, **49**(4).

Tryllian: 2001, Agent Development Kit, *Technical White Paper*, Version 1.0, June 2001, http://www.tryllian.nl/sub_downl/Technical%20white%20paper%20ADK%20v1.0.pdf

Vanwelkenhuysen, J. and Mizoguchi, R.: 1995, Workplace-Adapted Behaviors: Lessons Learned for Knowledge Reuse, *in Proceedings of Second International Conference on Building and Sharing Very Large-Scale Knowledge Bases*, Enschede, Netherlands.

Wagner, G.: 1996, A Logical and Operational Model of Scalable Knowledge- and Perception-based Agents, *in* W. van der Velde and J. W. Perram (eds), *Agents breaking away, Proc. MAAMAW'96,* Springer Verlag, Lecture Notes in AI, **1038**, pp. 26-41.

Wells, N. and Wolfers, J.: 2000, Finance with a Personalized Touch, *Communications of the ACM*, Special Issue on Personalization, **43**(8), 31-34.

Weyhrauch, R. W.: 1980, Prolegomena to a theory of mechanized formal reasoning, *Artificial Intelligence*, **13**, 133-170.

Williams, B. C. and Nayak, P. P.: 1996, A Model-Based Approach to Reactive Self-Configuring Systems, *in Proceedings of the AAAI-96*, **2**, pp. 971-978.

Wooldridge, M. J. and Jennings, N. R.: 1995, Intelligent Agents: Theory and Practice, *The Knowledge Engineering Review*, **10**(2), 115-152.

Wooldridge, M. J. and Jennings, N. R.:1995, Intelligent agents: theory and practice, *The Knowledge Engineering Review*, **10**(2), 115-152.