2015

Modeling the Evolution of Artifact Capabilities in Multi-Agent Based Simulations

Felicitas Mokom
University of Windsor

MODELING THE EVOLUTION OF ARTIFACT CAPABILITIES IN MULTI-AGENT BASED SIMULATIONS

by

FELICITAS ANYICHA MOKOM

A Dissertation
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada

2015

Modeling the Evolution of Artifact Capabilities in Multi-Agent Based Simulations

by

Felicitas Anyicha Mokom

APPROVED BY

B. Ombuki-Berman, External Examiner

Brock University

C. Thrasher

School of Nursing

S. Goodwin

School of Computer Science

R. Gras

School of Computer Science

Z. Kobti, Advisor

School of Computer Science

April 28, 2015

Declaration of Co-Authorship / Previous Publication

I. Co-Authorship Declaration

I hereby declare that this dissertation incorporates the outcome of joint research undertaken under the supervision of Dr. Ziad Kobti. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by Felicitas Mokom (the candidate) and Dr. Ziad Kobti (the supervisor) as primary authors and contributors. For the work presented in chapter 6, I collaborated with Kyle Bocinsky, Stefani Crabtree and Dr. Tim Kohler from Washington State University for many valuable discussions and analysis.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this dissertation, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This dissertation includes four original papers that have been previously published or accepted for publication in peer-reviewed conference proceedings, as follows:

Thesis Chapter	Publication Title/Full Citation	Publication Status
3,4	Evolution of artifact capabilities. *IEEE Congress on Evolutionary Computation (CEC)* 2011, 476–483, 2011.	Published
3,5	A cultural evolutionary model for artifact capabilities. *Advances in Artificial Life, European Conference on the Synthesis and Simulation of Living Systems* (*ECAL*) 2011, 542-549.	Published
3,7	Improving artifact selection via agent migration in multi-population cultural algorithms. *IEEE Symposium on Swarm Intelligence* (*SIS*) 2014, 1-8.	Published
3,6	Exploiting objects as artifacts in multi-agent based social simulations. *The International Conference on Autonomous Agents and Multiagent Systems* (*AAMAS*) 2015.	Accepted for Publication

I certify that I have obtained written permission from the copyright owner(s) to include the above published material(s) in my dissertation. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my dissertation does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my dissertation, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my dissertation.

I declare that this is a true copy of my dissertation, including any final revisions, as approved by my dissertation committee and the Graduate Studies office, and that this dissertation has not been submitted for a higher degree to any other University or Institution.

Abstract

Cognitive scientists agree that the exploitation of objects as tools or artifacts has played a significant role in the evolution of human societies. In the realm of autonomous agents and multi-agent systems, a recent artifact theory proposes the artifact concept as an abstraction for representing functional system components that proactive agents may exploit towards realizing their goals. As a complement, the cognition of rational agents has been extended to accommodate the notion of artifact capabilities denoting the reasoning and planning capacities of agents with respect to artifacts. Multi-Agent Based Simulation (MABS) a well established discipline for modeling complex social systems, has been identified as an area that should benefit from these theories. In MABS the evolution of artifact exploitation can play an important role in the overall performance of the system.

The primary contribution of this dissertation is a computational model for integrating artifacts into MABS. The emphasis of the model is on an evolutionary approach that facilitates understanding the effects of artifacts and their exploitation in artificial social systems over time. The artifact theories are extended to support agents designed to evolve artifact exploitation through a variety of learning and adaptation strategies. The model accents strategies that benefit from the social dimensions of MABS. Realized with evolutionary computation methods specifically genetic algorithms, cultural algorithms and multi-population cultural algorithms, artifact capability evolution is supported at individual, population and multi-population levels. A generic MABS and case studies are provided to demonstrate the use of the model in new and existing MABS systems.

The accommodation of artifact capability evolution in artificial social systems is applicable in many domains, particularly when the modeled system is one where artifact exploitation is relevant to the evolution of the society and its overall behavior. With artifacts acknowledged as major contributors to societal evolution the impact

of our model is significant, providing advanced tools that enable social scientists to analyze their findings. The model can inform archaeologists, economists, evolution theorists, sociologists and anthropologists among others.

Dedication

I dedicate this thesis to J.N. and our amazing children Mayang, Atabong, Didi and Kojo. You make it all worthwhile.

Acknowledgements

I would like to thank my supervisor, Dr. Ziad Kobti who has provided me with so much guidance and counseling throughout the years of my PhD research. He has been an exemplar of a supervisor for me, constantly motivating me with challenging and innovative research ideas and providing me with more than enough encouragement to see them through. I would like to thank him for being a wonderful mentor and let him know that he will always have my deepest appreciation and utmost respect.

I would like to thank the members of my committee Dr. Gras, Dr. Goodwin and Dr. Thrasher. Their valuable time and support is very much appreciated.

Many thanks to my friends and numerous family members who have encouraged and motivated me through the ups and downs of the past years. Thanks to my parents ... Mami - I can picture you dancing when I tell you this is all over.

Finally, I would like to thank my incredible husband who has "held down the fort" so effortlessly all this time. I must say you have indeed humbled me once more with your dedication and loyalty to our family, and the depth of your faith in me. Mayang, Atabong and my twins Didi and Kojo, you are the most wonderful kids any mother could ever hope for. I have been blessed with your patience, understanding and unconditional love which have all made this possible. I look forward to seeing the look on your faces when I get to tell you "Yes - my school has finally finished!"

Contents

List of Tables

List of Figures

List of Algorithms

List of Symbols and Acronyms

Symbol/Acronym	Explanation
$\triangleq$	is defined as
MAS	Multi-agent systems
MABS	Multi-agent based simulation
GA	Genetic algorithm
CA	Cultural algorithm
MPCA	Multi-population cultural algorithm

Chapter 1

Introduction

Providing a technology for naturally simulating the evolution of complex social systems, Multi-Agent Based Simulation (MABS) has distinguished itself in the past two decades as one of the most prominent areas of agent-oriented computing [18, 34, 45]. Recently in the related field of autonomous agents and multi-agent systems (MAS) an artifact theory has been introduced, proposing artifacts as an abstraction for representing reactive system components exploitable by proactive agents towards achieving their goals [77, 79]. Artifact capabilities refer to an agent's goal-directed subset of plans that constitute the exploitation of objects as artifacts [4]. MABS has been identified as one of the primary application areas where the artifact and artifact capability abstractions should provide significant benefits [80]. In complex socio-cultural systems the use of artifacts over time can play an essential role in the evolution of human capabilities and more generally the overall system performance. Integrating the artifact and artifact capability theories into their corresponding MABS systems warrants an extension that supports learning, adaptation and evolution.

The primary research question in this thesis is: "How can artifacts be integrated into MABS such that social agents can evolve artifact capabilities towards achieving their goals?" The primary question is addressed by providing a generic computational model for artifact capability evolution in MABS. The model extends the artifact and artifact capability theories and uses evolutionary computation methods to develop learning and adaptation strategies for evolving capabilities. Case studies are utilized to conduct experiments in both new and existing MABS systems.

Scientists are constantly searching for new ways to explain and provide insight into the complexities of human societal evolution [34]. Cognitive scientists are generally in agreement about the vital role played by tools or artifacts in the evolution of societies [79, 86, 89, 91, 117]. It is my hope that a domain independent computational evolutionary model for artifact capabilities should prove beneficial to both new and existing MABS systems towards understanding the effects of artifacts and their exploitation in the evolution of complex societies.

1.1 Background

1.1.1 Multi-Agent Based Simulation

In Artificial Intelligence (AI) an intelligent agent is any entity that can observe and act upon its environment. An intelligent agent is considered rational if it always takes the action that maximizes its performance given what it has observed so far and what it knows about its environment [118]. Russell and Norvig [103] describe a variety of agent types. For instance, reflex agents respond instantly to what they have observed using built-in condition-action rules. Goal-based agents act to simply realize their goal while utility-based agents act to optimally realize their goal. Learning agents improve their knowledge for achieving their goals over time. The environments that agents operate in differ in the difficulties they pose. Environments span from fully observable, deterministic, static, single agent environments to partially observable, stochastic, dynamic multi-agent environments which are most challenging.

A subfield of Distributed AI specifically used for problem solving, Multi-Agent Systems (MAS) or Agent-Based Models (ABM) are computational models characterized by multiple interacting autonomous agents operating in an artificial world. ABM agents can represent humans, robots, animals, households, organizations, countries or any other entity that can act upon its environment. Several characteristics of ABMs are responsible for their promotion as a prominent technology for modeling societies and other complex systems. In addition to their inherent characteristic of simulating the collective interactions and actions among autonomous and often heterogeneous individuals these models have the ability to naturally describe a system and generate observable emergent behavior at the population level [34, 45]. They are considered flexible as agents can be added or removed at will, created with varying degrees of rationality and given the ability to learn and evolve in the presence of others [18]. ABM applications for analyzing complex systems span a variety of fields including economics, engineering, anthropology, archaeology, social and biological sciences [45].

Social simulation is a scientific discipline concerned with simulating social interactions in order to study various issues in the social sciences. Multi-Agent Based Simulations (MABS) also called Multi-Agent Based Social Simulations or Agent-Based Social Simulations are social simulations built using ABMs. In addition to modeling communication between social agents sometimes realized with social networks [51, 102], MABS systems can include a cultural evolutionary component resulting in a powerful tool for modeling social and cultural effects on the overall performance of the system [52, 53, 50, 76]. Assisting in the formulation and validation of theories in the broad field of social science, MABS applications have transitioned from the modeling of simpler societies such as ant colonies [28] to complex human societies such as the Village Eco-Dynamics Project (VEP) [57, 58].

1.1.2 Exploiting Objects as Artifacts

1.1.2.1 Artifacts

Across the cognitive sciences researchers contend that tools have been instrumental in the evolution of human societies [23, 33, 79, 86, 89]. Over time the abilities to exploit objects in their environment commonly studied as tool use, have assisted humans in dealing with environmental changes ultimately leading to a modification of the environment in order to suit their needs. The most widely used definition of tool use in the literature is the one offered by Beck [11] where tool use involves the use of an environmental object detached from its user towards an objective with the user responsible for its proper use. It is well acknowledged that the use of tools is not limited to human societies. Some of the earliest studies in tool use behavior involve the experiments of Köhler conducted over several years to document tool use in chimpanzees [59]. Some other animal tool use studies involve the use of a pebble by wasps to pound earth into a nest [81], bottlenose dolphins using marine sponges for foraging [6] and the collection of dry manure by burrowing owls in order to attract insect prey [107]. Humans however exceed other species at their abilities to construct and exploit objects in their environment towards meeting their objectives. Among other fields studies in tool use include psychologists examining how children deal with tool use complexity [32, 90], archaeologists investigating early recordings of tool use [89], roboticists creating industrial robots preprogrammed to use specific tools [16] and a philosophical perspective defending the ontological status of artifacts as objects with practical functions that are made up of parts [7]. In relating tool use to intelligence philosopher Preston [91] argues that tool use should be considered a rival to language in the illustration of the high level cognition attributed to humans and philosopher Ronald Endicott proposes the "tooling test", an equivalent of the Turing test that assesses intelligence in terms of one's ability to exploit tools [117].

The argument that tool use is a sign of intelligent behavior has given rise to an interest in the AI community resulting in theories for tools and reasoning about their exploitation [4, 79], tool representation and recognition mechanisms [3, 15, 80, 119] and models for learning tool selection and use [21, 44, 50, 110]. In the MAS domain, the Agents and Artifacts (A&A) model [77, 78, 79, 80] proposes artifacts as an abstraction for tools in MAS, representing the function-oriented components of a MAS exploitable by its embedded agents towards realizing their goals. With MABS identified as one of the primary application areas that can benefit from the abstraction, the model categorizes the relationship between agents and artifacts in terms of three aspects: artifact selection, artifact use and artifact construction stipulating the ways in which artifact exploitation can occur. Omicini *et al.* [80] declare that one way to facilitate agents reasoning about exploiting artifacts in open and dynamic MAS environments (where artifacts are introduced into at any time and agents enter, leave or move around at will) is to render artifacts *cognitional*. Cognitional artifacts expose all that is needed by agents to properly select and use them for their goals. This includes the artifact's structure, behavior and effects of its use.

1.1.2.2 Planning

In AI, a planning agent is defined as a goal-driven agent whose objective is to construct a sequence of actions or a plan towards achieving a goal [103]. Planning is generally studied under two broad categories namely classical and non-classical planning distinguished by the features of the environment in which the planning agent or agents operate. Classical planning focuses on single planning agents operating in fully observable, deterministic, static and non temporal environments where fluents, that is, conditions that change over time, are propositional and only altered by the agent. Non classical planning extends classical planning to address a variety of practical planning problems including those encountered by multiple planning agents. Fluents may be metric or continuous and alterable by factors other than the planning agent's actions. The environment may be partially observable, stochastic and/or dynamic and the effects of the agent's actions may be temporal.

1.1.2.3 Learning

In AI, machine learning entails the use of percepts and observations of past experiences by learning agents to improve future actions [103]. Characterized by the feedback provided to the learning agent the three main approaches to learning are supervised learning, unsupervised learning and reward-based learning. In supervised learning such as decision tree learning, the feedback provides the correct output to the agent. In unsupervised learning for example clustering, no feedback is provided at all. In reward-based learning the agent receives a fitness evaluation of its actions usually in some form of penalty or reward. An agent applying reward-based learning methods learns to achieve its goals through a trial and error process of visiting its environmental states online. Reward-based learning is usually considered more suitable for addressing learning problems in MAS given the complexities that arise from acting by multiple interacting agents [83]. Reward-based learning is often characterized in two facets: reinforcement learning and stochastic search. In reinforcement learning such as Q-learning [115, 116], agents are concerned with estimating action-value functions. Stochastic search which includes methods such as simulated annealing and the family of evolutionary computation techniques, involves direct learning without learning value functions.

Analogous to their counterparts in real societies, social agents embedded in MABS can develop learning strategies that benefit from its social dimensions, that is, enhance their learning capabilities through feedbacks received from other agents. These social learning mechanisms include observational learning, learning by instruction and collaborative learning [8, 112, 113].

1.1.2.4 Artifact Capabilities

In order to incorporate reasoning about artifacts into rational agency, Acay *et al.* [4] integrated Omicini *et al.*'s [79, 80] artifact theory into the Belief-Desire-Intention software model (BDI) [82, 92]. BDI is based on the philosophical theory of Bratman [20] which characterizes the cognition of a rational agent in terms of beliefs, desires and intentions. The agent's beliefs describe its informative state about the world while its desires used to formulate goals, represent what the agent would like to accomplish. The agent's adopted goals are considered its intentions and act as a trigger to plans that constitute actions the agent will perform. BDI addresses the problem of rationally selecting and executing existing plans. Padgham and Lambrix [82] introduced capabilities into BDI. Capabilities which according to the authors promote modularity and reusability and support meta-level reasoning, resided within an agent's intentions and abstracted the set of plans relevant to a goal. Acay *et al.* [4] further extended capabilities to include internal and external capabilities. In their logical representation, internal capabilities denote plans that an agent can accomplish on its own while external capabilities refer to plans that an agent can carry out with the help of artifacts or other agents. An artifact capability relates a set of artifact plans to a goal where artifact plans specify artifact functionalities describing ways to exploit artifacts in order to achieve the goal. Evolving artifact capabilities in MABS involves the construction of these plans. Within the context of planning it can be viewed as a non classical planning problem as it concerns at a minimum, multiple agents constructing plans. It can also be approached as a learning problem, since artifact plans can be evolved through learning. Acay *et al.*'s extended formalization of the mental attitudes of rational agents to include artifact capabilities provides a theoretical foundation for studying artifacts and their role in the evolution of societies.

1.2 Research Motivation

MABS systems are built with the objective of understanding the intricacies involved in the evolution of complex societies. With artifacts established as significant contributors to societal evolution, the evolution of their exploitation is as relevant as the exploitation itself. How social agents evolve artifact capabilities may have measurable effects on the overall performance of the society.

Existing models that integrate artifacts based on Omicini *et al.*'s artifact theory into MABS [30, 31, 35, 71, 85, 105, 106] have done so with the A&A model's cognitional artifacts, which expose to agents all that is needed to successfully exploit them. Cognitional artifacts distribute autonomy over artifacts rather than agents, giving artifacts control over the knowledge agents can possess for exploiting them. Agents in these models require a complete information model for artifacts as they do not learn. Furthermore adaptation and evolution is only supported in terms of agent encounters with artifacts and automatic replication of the knowledge embedded in them. As a result, agents loose heterogeneity with respect to artifact capabilities. The authors of the A&A model state that although cognitional artifacts facilitate cognitive artifact exploitation they are not a requirement for applying the underlying concepts of their artifact theory. Artifacts can expose minimal information, as long as agents are built with the ability to learn unexposed properties [80](pg. 447, Section 4.3). This approach which we argue is necessary for MABS systems that are interested in the evolution of cognitive artifact selection and use, is yet to be adopted.

Some other studies that are not based on the artifact theory but integrate artifacts into MABS in a way that supports agents learning some aspects of artifact capabilities do so in a domain dependent manner [37, 38, 50, 54], do not accommodate an evolutionary dimension [74, 75], or are driven by the requirements in robotics that warrant a detailed focus on robotic sensors and body schemas[61].

The primary research question is: "How can artifacts be integrated into MABS such that social agents can evolve artifact capabilities towards achieving their goals?". That is, given a MABS system consisting of social agents AG and objects O:

1. How should $o \in O$ be represented such that it can be exploited as an artifact by $ag \in AG$.

2. How should $ag \in AG$ be represented so that it can cognitively reason about artifacts and evolve the knowledge for their exploitation.

3. What learning and adaptation strategies can $ag \in AG$ develop for exploiting artifacts from O especially taking advantage of the social dimensions in MABS.

1.3 Assumptions

The presented generic computational model for artifact capability evolution in MABS makes the following assumptions:

- Agents are autonomous.

- Agents can be heterogeneous with respect to their learning abilities.

- Agents have common adopted goals.

- Agents are driven to best achieve their goals and may cooperate with respect to sharing knowledge in order to do so.

- Agents can communicate via static or dynamic social networks.

- Artifacts can be static or dynamic with respect to feedback from their exploitation.

- The MABS system can include a cultural evolutionary component that facilitates learning from a common knowledge base.

- The MABS system can consist of multiple populations that evolve independently of each other.

1.4 Research Objectives

The aim of this research is to enhance the ability of artificial social agents in MABS to achieve their goals through learning and adaptation strategies for exploiting objects in their environments as artifacts. This will be realized by providing a generic computational model for integrating artifacts and artifact capability evolution in MABS systems. Two of the three aspects of artifact exploitation given by Omicini *et al.*'s artifact theory will be addressed namely artifact selection and artifact use.

The goals of the research therefore are:

- To provide a representation for artifacts based on the A&A model's artifact theory that facilitates learning and adapting artifact selection and use.

- To provide a representation for an agent's cognition that extends the BDI-based artifact capability theory to support learning and adaptation. This is realized by combining the cognition in the existing theory with the cognition of general learning agents in AI.

- To enable agents to evolve artifact capabilities by means of learning and adaptation strategies with an emphasis on strategies where agents take advantage of the social dimensions of MABS. All strategies are developed with computational intelligence techniques specifically evolutionary computation methods. Individual experience learning is realized via genetic algorithms (GA). Social experience learning is realized through GAs, social networks, cultural algorithms (CA) and multi-population cultural algorithms (MPCA). As a result learning and adaptation are accommodated at individual, population and multi-population levels.

- To demonstrate the use of the model for building new MABS systems with artifact capable agents. A generic MABS system and a MABS system for child auto safety restraints are provided.
- To demonstrate the integration of the model into an existing MABS system. The model is integrated into the MABS system which constitutes a significant part of the Village EcoDynamics Project [52, 56, 57, 58] developed over the past two decades to study the lives of the ancient Pueblo Indian settlers in the American Southwest during a period spanning 700 years.

1.5 Research Contributions

This thesis provides a generic computational model for incorporating objects into MABS as artifacts (based on the A&A model) and enabling social agents to properly exploit them over time towards realizing their goals, by means of learning and adaptation strategies.

The model uses an evolutionary approach to integrate artifacts and the capabilities for exploiting them into MABS. Although this approach is not entirely new, existing models are domain dependent and limited. Our model is generic, grounded in established theories and provides a more extensive set of learning and adaptation strategies. The versatility of our model is also apparent in its support for heterogeneous agents, static and dynamic social networks, dynamic artifacts and dynamic environments. The model is scalable, accommodating learning and adaptation at individual, population and multi-population levels. We also acknowledge that the adoption of the A&A paradigm in MABS is not new. What is novel in our work is the extension of the existing theories to include learning, adaptation and evolution mechanisms. To the best of my knowledge the work presented in this thesis constitutes the first A&A-based domain independent model that specifically addresses artifact capability evolution in MABS.

The model can be used to model artifacts in new MABS systems or integrate the artifact concept into existing ones. Integration into existing systems permits the elimination of presumptions that artifact capabilities are inherent to agents. With artifact capabilities evolving over time, different aspects of system performance can be measured.

Artifact capability evolution in MABS is applicable in a wide variety of domains. Any artificial social system where artifacts and their exploitation impacts the evolution of the system or its behavior can benefit from our model. The model can inform fields such as sociology, anthropology, economics, evolution theory and archaeology. Among other areas the model can be used in Health Care for instance, to study relevant problems related to patient self-management and equipment use in hospitals. In this thesis the model's applicability is demonstrated in two distinct domains: Transportation / Injury Prevention with the child auto safety restraint case study and Anthropology/Archaeology with the Village case study.

1.6 Dissertation Outline

The rest of the thesis is structured as follows. In Chapter 2 we provide a background on studies related to artifacts in multi-agent environments. We review work on the representation of artifacts in MAS environments, the representation of MAS agents that can reason about the exploitation of objects as artifacts and the exploitation of artifacts in MAS. In addition some studies in other fields that address artifact exploitation in social and cultural contexts are presented.

In Chapter 3 we provide our representations for artifacts along with MABS agents that reason about them and can evolve knowledge for their exploitation. The artifact and artifact capability theories are extended to facilitate learning and adaptation of artifact selection and use. Additionally the methods that will be utilized to implement the learning and adaptation strategies for MABS agents are presented along with the technology used for implementing our MABS, its agents and artifacts.

In Chapter 4 we present a model for individual and social learning of artifact use in MABS. Agents are expected to learn one way to use a given artifact for an adopted goal. Learning strategies are developed for individual and observational learning using GAs. Artifacts are assumed to be predictable with respect to their outcome and behavior. A generic MABS is developed to conduct experiments comparing agent performance with respect to the learning strategies.

In Chapter 5 we extend the model to include a cultural component. Agents can now learn through individual experience and two forms of social experience, namely observational learning and collaborative learning. Collaborative learning is realized through the use of a CA where two categories of knowledge sources are maintained in a shared belief space accessible by the agent population. As in the previous chapter, the agents are given an artifact and a goal to realize with it. The new learning strategy is added to the previously implemented generic MABS and experiments are conducted comparing it to the others.

In Chapter 6 the model is further extended to address adaptation strategies for artifact use in unpredictable environments. Social structures are introduced into the agent population and agents maintain static or dynamic social networks through which they communicate with other members of the population. Agents are expected to adapt artifact use for unpredictable artifacts in dynamic environments. Additional learning strategies are provided including learning through social networks, combining learning strategies and using a meta-learning strategy for strategy evolution. A case study that integrates the model into the existing MABS of the Village EcoDynamics Project is presented.

In Chapter 7 we address the other aspect of artifact exploitation, namely artifact selection. Agents are presented with a set of artifacts and given the objective of selecting the proper one to realize their goal. Artifact selection is modeled in a multi-population setting using MPCAs and agent migration is used to facilitate knowledge transfer between independently evolving social populations. Agents within each population learn both through social networks and their respective cultural belief spaces. A MABS for child auto safety restraints is provided as a case study and used to measure the effects of migration on artifact selection knowledge.

In Chapter 7 conclusions are presented summarizing the overall contributions of the model. We identify some of the model's limitations and discuss some potential future directions for the work.

Chapter 2

Literature Review

When compared to other hallmarks of intelligence such as language, the exploitation of objects as tools or artifacts has been much less studied in AI [21, 79]. Most studies that have addressed the problem focus on artifact capabilities for agents operating in isolation [21, 44, 110, 119]. As a result research that explicitly deals with artifact capabilities in multi-agent environments especially their evolution are quite limited. In this chapter we review studies related to the three primary objectives of our research. We begin with a short report on studies that address representing artifacts in MAS followed by representations for MAS agents that reason about artifacts. Next, previous work that deal with artifact exploitation in multi-agent environments are provided. Finally a a few studies on artifact exploitation in other fields are reviewed.

2.1 Representing Artifacts in MAS

Omicini *et al.* [79, 80] introduced the first and to the best of our knowledge, the only established artifact theory for MAS in their Agents and Artifacts (A&A) model. The authors argued for the *Agens Faber* approach to modeling intelligent agents in MAS, analogous to the philosophical concept of *Homo Faber* which characterizes humans affecting their environment through tools. Acknowledging the interdisciplinary nature of the subject the authors used inspiration from many fields including Activity Theory (AT), Distributed Cognition, Sociology, Anthropology and computer-supported cooperative work (CSCW) to develop the theory. In the theory artifacts are proposed as an abstraction for representing the reactive system components in MAS made available to agents. While agents characterize the proactive MAS components responsible for acting upon the environment, artifacts depict the functional components exploitable by agents towards realizing their goals. To facilitate agent's reasoning about exploiting artifacts, the theory defines an artifact in terms of three essential properties that it may expose:

- usage interface (*UI*)
- operating instructions (*OI*)
- function or service descriptions (*FD*)

UI describes the external structure of the artifact that is observable in the form of a set of permissible operations. An agent performs an action on an artifact by executing these operations. Analogous to a user manual that guides the use of an object, the *OI* set describes procedures for using the artifact for a given purpose. An element of *OI* is a sequence of *UI* operations. Finally, *FD* abstracts the functionality provided by the artifact according to the intentions of its creator. Thus a function description is related to one or more operating instructions, each of which are composed of usage interface operations. While *FD* specifies *what* an artifact is used for hence facilitating artifact selection, *OI* indicates *how* it is to be used therefore aiding artifact use.

Omicini *et al.* [80] characterize artifacts that expose these three properties as *cognitional artifacts* arguing that in open and dynamic MAS environments they permit cognitive selection and use of artifacts by agents. *FDs* can be used by agents to select the proper artifacts and artifact use can be realized by executing the correct *UI* operations with the help of *OI*. With this knowledge embedded in the artifacts themselves agents can rely on the environment to supply and maintain all the necessary information for proper artifact exploitation. Artifacts can be added, removed or modified and agents would automatically adapt to the current environment and the current state of the available artifacts. The artifact theory suggests other properties for artifacts. For instance *predictability* can be offered by *FD* for predicting the outcome of an artifact. Another property *linkability* allows artifacts to interact with each other. Linkability for example could be used to capture a remote control turning on a TV.

Implementations of A&A's properties have been in the form of identifiers and function calls that the agent can invoke such as in Ricci *et al.*'s *CArtAgO* (Common ARtifact infrastructure for AGent Open environments) [100, 101]. For instance, an artifact *Camera* may have a function description *basic-photo-shoot* which is associated with an operating instruction consisting of operations *power-on* and *shutter-release*. The operations are specified in terms of a name, arguments and outcome. An agent with the objective of taking a photo can exploit *Camera* with the details of its operations hidden within the artifact.

Acay *et al.* [2] argued for the use of semantic technologies to construct the information model that describes artifacts and their properties, in order to facilitate semantic interoperability or shared meanings with respect to artifact exploitation. They presented a tool ontology (OWL-T), a sort of tool manual for describing artifacts and their exploitation in MAS environments. OWL-T resided in the MAS environment and could be queried by agents in order to dynamically select and use artifacts. The authors used description logic (DL) to build OWL-T which included formal descriptions of all properties for artifacts along with agents that reason about them. In OWL-T two concept categories primarily describe environmental objects: *ObjectModel* and *AbstractConcept*. According to *ObjectModel*, an *Object* can have a *PhysicalProperty* and can be an *Artifact* or a *Tool*. An *Artifact* realizes an agent's goal while a *Tool* can have an *IdealProperty*. The distinction between *Artifact* and *Tool* separates the object's functionality captured in *Tool* from its relation to an agent's objective specified in *Artifact*. An *Artifact* is the object involved in the agent's activities according to the agent's role. *PhysicalProperty* captures the shape and spatial features of an object. *IdealProperty* describes an object's functionality in terms of which activities it can be used for and the actions that are supported. Revisiting the *Camera* object, its shape and location in the environment may constitute *PhysicalProperty*. As an *Artifact* it can be used to realize the objective *basic-photo-shoot* by an agent with the role *photographer* involved in the activity *photo-taking*. As a *Tool*, *power-on* and *shutter-release* are the supported actions of *IdealProperty* for the activity *photo-taking*.

Representing artifacts as cognitional artifacts empowers the environment and can be quite suitable for MAS systems that model interactive objects or objects that will be exploited by agents in a uniform fashion. With the complete information model for artifacts stored in the environment, agents can dynamically select and use them. The A&A model has been applied in different MAS sub fields [77, 80] including MAS programming [87, 101, 111], ambient intelligent applications such as HomeManager [70], self-organizing systems and MABS [30, 85, 35]. The representation is however ill-suited for MABS systems concerned with examining the evolution of artifact selection and use. Representing operations as high-level function calls do not facilitate agents that wish to manipulate the artifact themselves and learn how to exploit them to realize their goals. In order to do so, a complete information model should not be assumed and an artifact should expose minimal information such as only its *UI* from which agent's can evolve knowledge for its *FD* and *OI* properties. Furthermore an artifact's *UI* should expose its structure at a lower level. In other words, if an agent is going to learn to *power-on* a camera, then it needs a representation that describes exactly what *power-on* characterizes in terms of the artifact's structure.

2.2 Representing Agents Reasoning about Artifacts in MAS

Omicini *et al.*'s artifact theory described how MAS agents can reason about exploiting artifacts for their goals. Artifact exploitation by agents is depicted in three aspects:

- artifact selection
- artifact use
- artifact construction and manipulation

Artifact selection involves the agent's ability to select the proper artifact that it can use to realize its goal. Artifact use involves the use of a selected artifact by carrying out a sequence of instructions to realize a goal. When either artifact selection or use results in the failure to achieve its goal, an agent may then construct new artifacts or manipulate existing ones. The theory further defined five cognitive levels at which agents could reason about artifacts: unaware use, programmed use, cognitive use, cognitive selection/use and construction/manipulation. At the first level, unaware use involves the implicit use of artifacts by agents, that is agents never explicitly act on artifacts. At the second level, agents are preprogrammed to select and use artifacts. Agents at this level are designed with embedded plans for artifact exploitation. Agents at the level of cognitive use are designed to know the proper artifact to select but need to discover at run-time how to use those artifacts. At the fourth level, agents discover both artifact selection and use at run-time. Finally, in the fifth level of artifact construction/manipulation agents themselves become designers of artifacts. Cognitional artifacts offered by A&A correspond to level four, where agents can dynamically select and use artifacts in MAS.

While providing the general descriptions for artifact-capable agents, A&A did not provide an architecture for the agents. That contribution was offered by Acay *et al.*

[2, 4]. In Acay *et al.* [2] the authors coined the term *extrospection* to characterize the reasoning process carried out by agents in order to select and use artifacts to realize their goals. The cognition of an agent that can reason about exploiting artifacts was represented in terms of goals, beliefs and plans. The agent queries the OWL-T ontology about which artifact to select for a given goal and the plan for using it. In addition to the concepts that describe artifacts, OWL-T provides concepts describing the agent. According to *AgentModel* an agent has goals and beliefs. *ActionModel* is used to describe how the agent carries out its activities by itself or with other agents by reasoning using its beliefs and goals and involved artifacts. Acay *et al.* [4] introduced the notion of artifact capabilities integrating the artifact theory with the Belief-Desire-Intention software model (BDI) model [92] of rational agency. BDI agents maintain a pre-existing plan library with plan selection and execution driven by their beliefs (informational states), desires (motivational states) and intentions (deliberative states). While the agent's beliefs describe what it knows about the world, its desires are used to create goals that once adopted become intentions. Intentions are represented as plans which constitute action sequences that the agent can carry out. Acay *et al.* [4] extended Padgham and Lambrix's [82] concept of capabilities in BDI that encapsulates plans relevant to a goal, to distinguish between plans an agent can perform on its own (internal capabilities) and plans that it can carry out with the help of tools or other agents (external capabilities). Hence artifact capabilities refer to plans that are artifact functionalities specifying ways to exploit an artifact for an objective, that is, plans the agent can carry out with an artifact to achieve a goal.

Since an inherent aspect of BDI agents is that they are concerned with balancing the selection and execution of *existing* plans, the artifact capability theory works quite well with the A&A model's cognitional artifacts. Since artifacts themselves expose the properties needed to exploit them, BDI agents can simply duplicate this information upon encountering an artifact and represent it as an artifact capability.

On the other hand, if a complete information model for artifacts is not available or the objective of the system is to gain insight into the evolution of artifact capabilities by agents then the agent representation becomes insufficient. Agents being *told* how to select and use artifacts are not evolving as a result of their own abilities.

2.3 Artifact Exploitation in MAS

In general we have identified two main approaches to addressing artifact exploitation in systems consisting of multiple interacting agents:

- *Agent Design Perspective*: These models adopt an agent-centric perspective to the artifact exploitation problem. Agents are built to employ learning strategies towards discovering how to properly exploit artifacts for their goals.

- *Environment Design Perspective*: In these models artifacts are designed to expose to agents all that is needed for their proper exploitation. This automatically enables agent discovery as agents can dynamically exploit artifacts for their goals.

2.3.1 Agent Design Perspective

Artificial life researchers Noble and Franks [74, 75] utilized the exploitation of tools to demonstrate social learning in animals. In their study a variety of social learning methods namely *imitation*, *emulation*, *following* and *contagious behavior* were compared in a simulation composed of animal agents. The authors employed reinforcement learning, specifically Q-learning in which an agent learns current and delayed payoffs of taking an action in a state. Agents learned actions or sequences of actions that were necessary to best acquire resources with the selection and use of tools. The simulation accommodated specific tools and resources which were associated with different payoffs dependent on the manner in which they were obtained and what action was carried out with them. Hence the information model exposed by the tools was incomplete, that is, agents were not told the correct optimal actions to utilize and were expected to employ reinforcement learning methods towards evolving tool exploitation. *Following* was realized by situating an agent in the same location as its parent for part of its lifetime. *Contagious behavior* was defined as a probability that an agent would perform an action it just observed. *Emulation* involved an agent recognizing that another in its presence obtained a positive payoff and positively adjusting the payoffs of all actions related to its current state. With *imitation* the agent recognizes both the successful agent's state and action and adjusts the payoff of the specific action. The authors concluded that *emulation* is sometimes superior to *imitation* due to its promotion of exploration. Given the objective of the studies to compare simple social learning methods rather than address artifact exploitation in MABS, actions were specific and tools and agents were represented for the most part as identifiers. Moreover the authors themselves acknowledge the lack of an evolutionary dimension to their work.

Mohan and Morasso [61] explore how learning from previous experiences and social interactions can enhance knowledge for exploiting tools. The authors presented a learning architecture for cognitive robots which supported combining knowledge gained from practice with the new tool, past experiences and social interaction in the form of imitating an observed demonstration from a teacher. Robots could learn to use the tool in a new way towards realizing a goal. The model focused on the robot learning coordination of the movements of its upper body such as its spatial and temporal trajectories and the geometric relationships between the movements made and the resulting effects on the tool. A demonstration was conducted with a humanoid, iCub, learning to coordinate a toy crane to pick up unreachable objects in its environment. The authors argued that their skill learning architecture was novel in that imitating the teacher's demonstration and utilizing parameters obtained from past experience reduced the space that the robot needed to explore with the new toy. Given their objective of building useful machines robotic models pay significant attention to robotic sensors and aspects of the robot's body schema. This renders the learning architecture very complex focusing on details that may be abstracted when the objective is to explore artifact exploitation effects on the evolution of a system rather than the intricate exploitation at agent level. In addition social learning occurs only in the model from robots observing a teacher rather than interacting with others in the environment.

Kobti *et al.* [50] presented a MABS built for multiple interacting driver agents in a population learning to select child restraints. As an integral component of a health care decision support system [48] the model supported social and cultural influences such that an agent's restraint knowledge could be altered by others in its social networks and cultural beliefs respectively. The social networks consisted of a subnetwork of kinship connections and another characterized by neighbors. A cultural algorithm was used to accommodate cultural influence. Cultural algorithms (CA) [95, 96] defined as computational models of cultural evolution, consist of a population space and a belief space with a communication protocol between them. Agents in the population space contribute knowledge to the belief space, which maintains various categories of knowledge. The knowledge from the belief space in turn influences the evolution of the population. In Kobti *et al.*'s model, situational knowledge characterizing the best examples extracted from the population was used as the source of cultural influence. Driver agents could also learn from individual experience through interventions from a source of standard correct knowledge. Agent knowledge structures were defined to capture the agent's knowledge with respect to selecting a restraint and the appropriate location to place it in the vehicle according to the age, weight and height of the child. Agents were defined with learning and retention rates affecting how they were influenced. The learning rate defined the probability that an agent learned a bit of knowledge correctly from an influential source while the retention rate defined the probability that the agent resisted influence. These rates were utilized to mutate the agent's knowledge during influence. The model was extended to capture both positive and negative examples in the cultural space in Kobti *et al.* [54]. Experiments conducted with the models demonstrated that the overall performance of the population, measured using randomly simulated accidents and the injury outcome, increased with both standard interventions as well as social and cultural influences over time. They also showed that the cultural aspect rendered the population resilient to changes after standard interventions. The child safety restraint simulation was validated in Gupta *et al.* [37, 38]. Using data mining techniques including decision trees and regression analysis on an actual survey in child safety restraint to generate parameters for initializing agents in the simulation, the authors conducted experiments and validated the results against subsequently gathered survey data.

The child safety model demonstrates agents evolving artifact exploitation in MABS, however there are several limitations. First, the model is domain dependent and not based on an artifact or artifact exploitation theory. The provided agent knowledge structures are specific to auto restraints without any suggestion for generalization of the model to encompass other domains. Next, only restraint selection and placement in the vehicle are addressed neglecting the step by step operational use of the restraint. Restraints in the model are represented only by a label. While this suffices for learning the selection of a restraint according to characteristics of the child it is to be used with, a model that addresses its actual use would have to be concerned with actions performed with its relevant parts. Proper restraint use beyond the seat type and location is important as research has shown that sub-optimally restrained children are injured more often and are at a higher risk of more serious injuries [22, 108]. Another limitation arises from the fact that individual learning only occurs through a direct intervention. Agents do not evolve their knowledge on their own through exploration. Next, the CA framework proposes and supports up to five different types of influence from the cultural space [84] however only situational knowledge is utilized. Also, social networks are not dynamic. Finally while the model supports knowledge propagation through social and cultural influences within a single population it does not address knowledge transfer between multiple populations as has been suggested by researchers employing multi-population cultural algorithms (MPCA) [40, 73].

2.3.2 Environment Design Perspective

Existing studies in artifact exploitation in MAS that utilize an environment design perspective are based on the artifact theory and the A&A model.

2.3.2.1 Artifact Exploitation by BDI Agents

Artifact capabilities [4] and OWL-T [2] were combined in Acay *et al.* [3] to present a model that demonstrated artifact exploitation by BDI agents. The authors claimed that the manner in which their agents could reason about exploiting artifacts which they referred to as *extrospection* can be construed as a form of learning and planning. However, we have distinguished their work from agents that employ learning strategies since learning in their model was only accomplished by agents replicating information exposed by OWL-T in a local *tool base.* Complimentary to the agent's plan library, the tool base contained an agent's local copy of OWL-T's tool plans or artifact capabilities which basically *tell* the agent when to select an artifact and provides the step by step instructions for using it.

The exploitation of artifacts by BDI agents has been used in environment programming studies. Piunti *et al.* [88] demonstrate how service-oriented architectures (SOA) and Web Service (WS) systems can be programmed using BDI agents operating in artifact-based environments. The authors used the BDI-based agent oriented programming language *Jason* [19] for their agents and the *CArtAgo* framework [100] for artifact-based environments. The result was a platform for programming environments composed of artifacts exploitable by BDI agents. In another example of a system consisting of BDI agents and artifacts, a platform combining organization-aware BDI agents exploiting artifacts is offered [17, 111]. The proposed platform *JaCaMo* combined *Jason* agents, *CArtAgo* environments and *Moise* [43], a technology for incorporating organizations in MAS. In *JaCaMo* participant agents conformed to organization rules and could cooperate to achieve goals. The authors argued that the unified platform supported agents, environment and organizations which they consider the three primary levels of MAS abstractions.

Systems that support BDI agents exploiting artifacts do not accommodate learning and evolution by agents themselves. As a result they are not suitable for MABS systems where agents are expected to evolve artifact capabilities using learning abilities.

2.3.2.2 Artifact Exploitation in MABS

Adaptive complex systems capture the abilities of many simple agents collaborating to result in the emergence of complex behaviors observable at the global level. An adoption of the A&A paradigm in MABS is offered by Gardelli *et al.* [30, 31] for designing its self-organizing aspects. Employing the A&A model the authors proposed an architecture consisting of user agents, artifacts and *environmental agents.* Complimentary to user agents (standard agents) that represent the proactive entities that can use artifacts which wrap system resources, environmental agents do not interact with user agents but rather are responsible for managing artifacts towards facilitating self-organization. These agents handle tasks such as modifying artifact properties so that the system adapts to unpredictable aspects of agents exploiting artifacts. The authors propose a three step process in self-organization design. First an abstract model of the system is developed as a formal specification. Next a MABS simulation is used to examine the dynamics of the model with different parameters to produce correct system behavior. Finally the model is fine-tuned with any relevant revisions.

Although the application demonstrates an implementation of the artifact abstraction in MABS the objective of the study was to introduce system design techniques rather than to explore artifact exploitation. As such artifact exploitation by standard agents only evolved with respect to the behavior of environmental agents. Standard agents did not employ any learning methods and still interacted with artifacts that exposed all needed for their exploitation. The study only suggested a means for the artifacts in the environment to alter or improve their cognitional aspects.

Another application of the A&A model in MABS involves the modeling of biological systems as complex systems. Montagna *et al.* [71] argue alongside the A&A model authors for the applicability of A&A in MABS. They suggest using A&A's agents which they termed *bio-agents* to represent biological system components that display autonomous behavior for instance macro-molecular components such as proteins at the intra-cellular level. Abstracted artifacts (*bio-artifacts*) are proposed for the function-oriented aspects of the biochemical environment such as cell micro-environments at the inter-cellular level. Bio-artifacts can mediate actions and interactions among bio-agents contributing to coordination in the biological system. The authors provided an implemented case study modeling the glycolysis metabolic pathway. Along similar lines of providing a coordination-based model for modeling the interaction among system components in a biological system, Perez *et al.* [85] presented a biological system for capturing the complex interaction patterns of intracellular signaling pathways employing the TuCSoN tuple-based middleware for MAS coordination.

The biological system models are concerned with coordinating the interactions among biological system components. The bio-agents do not employ learning strategies towards augmenting their capabilities for exploiting bio-artifacts. Employing the A&A model once again involves cognitional artifacts that provide a complete information model usable for their exploitation.

Siebert *et al.* [105, 106] use the A&A model to build their proposed complex system simulation which models a society of heterogeneous interacting models. The model focused on addressing difficulties regarding coordination among models from different domains. The authors suggested *coupling-artifacts* for implementing the coordination model and handling compatibility issues among the interacting models and *model-artifacts* for controlling the simulation process. *Model-agents* could execute their tasks by exploiting the artifacts. The authors argue that their framework facilitates reusing existing models and promotes modeling interactions between scientific domains. The model was recently applied to co-simulate a smart space heating environment consisting of an electrical heating-based simulation and networking event-based simulations in order to understand the relationship between geometrically represented rooms in a house and the efficiency of heating and network connectivity [35].

With respect to artifact exploitation the focus of the multiple model simulation was on coordination aspects such as resolving conflicts that may arise with regard to execution times. The model therefore did not address learning or evolution of exploiting the artifacts for the model-agents. The artifacts exposed to the model-agents all that was needed for their proper selection and use.

2.4 Artifact Exploitation in Other Fields

Given the interdisciplinary nature of the subject there are many studies addressing artifact exploitation in other fields. A few recent studies that deal particularly with artifact exploitation in social and cultural contexts are reviewed.

Animal cognitivists study artifact exploitation under animal tool use behavior. Bacher *et al.*'s [6] attempt to distinguish between socially learned and and genetically transmitted tool use knowledge in dolphins. The authors conducted experiments involving the use of marine sponges for foraging by bottlenose dolphins, to address prior contentions that the tool use behavior is socially learned, particularly by female offsprings from their mothers. This contention had been attributed to dolphins possessing the capability for imitation, a form of social learning. The authors argued for the possibility that the behavior is a result of gene-culture co-evolution, providing evidence that mitochondrial genes though relevant were not sufficient to explain the multiple observed variations in the sponging activity. Yamazaki *et al.* [121] presented a case study involving the training of common marmosets not known to use tools in the wild, to use a rake-shaped tool to retrieve food. The training process involved exploratory tool use learning with a four stage process where the monkeys were rewarded if the right action was performed during each stage. The training was incremental as required actions increased in difficulty during each stage. With substantial training the five marmosets involved in the study successfully learned to use the tool. An agent-based model (ABM) that includes a component for tool use behavior in bearded capuchin monkeys was presented by Bernades *et al.* [12]. The model explored the use of stones by the monkeys for cracking nuts. With tool use considered central to their work the authors claimed that the model would allow for its simulation and added that they were in the process of incorporating learning into their model. Learning would be accommodated via reinforcement learning and would support different learning scenarios. The authors however did not provide any details on the representations for their tool or the learning strategies that would be employed. Yamamoto *et al.* [120] presented a study where chimpanzees augmented their tool using behaviors from observing more efficient techniques invented by others. The authors argued that their study was the first to demonstrate animals ameliorating their tool using efficiency over time through this kind of social learning. They also suggested that their study provided some insight into incremental cultural evolution in non-humans. In their attempt to capture the social spread of tool use behavior over time in chimpanzees Hobaiter *et al.* [41] compared two variants of tool use behavior in static and dynamic social networks models. Dynamic networks differ from their static counterparts in that they attribute time to observations and can therefore track not just observed behavior but its ability to only affect subsequent ones. The authors claimed that in their approach they were able to distinguish between behaviors obtained primarily through individual learning from those learned socially.

Cognitive scientist David Kirsh [49] argued that humans, artifacts, artifact exploitation behavior and tasks co-evolve in what they described as an artifact ecology. According to Kirsh aside from artifacts that are esteemed for reasons other than their utility, artifacts are usually created to carry out a task. They usually belong to collections for example a needle and thread, or a car seat and a car. Once the artifacts exist individuals realize the task in different ways. The ways in which the artifact is exploited also evolves as new uses unintended by its designer are discovered. For instance a pen may have been designed for writing but can be used as a bookmark. New skills developed by individuals increase the demand for better tools. The author argues for a link between the superior functionality of an artifact to its prevalence and persistence within a culture given that an artifact when used properly should yield a better performance than when utilized incorrectly with everything else being equal.

Gardiner *et al.* [32] compared observational and individual learning in studying the difficulty of tool use abilities in children. Conducted experiments with two and three year olds learning by exploration demonstrated that observational learners outperformed individual learners. They also observed that an increase in task complexity correlated with a reduction in performance. Children were also more likely to employ observational learning in lieu of individual learning as the difficulty of successfully using a tool increased.

Aside from the study by Bernades *et al.* [12] these studies are primarily conducted with surveys or experiments in the field and are limited by their use of descriptive approaches lacking formal representations for tools or those that exploit them. Technologies such as MABS provide a means for social scientists to computationally test their theories and gain some insight into the underlying reasons behind the emergent phenomena. Bernades *et al.* propose to contribute to that effort with their ABM, however to the best of our knowledge details of their model with regard to tool use learning are yet to be made available.

Chapter 3

Representing Artifacts and Artifact Capabilities for Artificial Social Agents

In this chapter, representations for artifacts in MABS along with agents that can reason about, learn, adapt and evolve knowledge for their selection and use are provided. The artifact representation is based on Omicini *et al.*'s [79, 80] artifact theory. The objective here is to construct a representation for artifacts based on the theory that facilitates learning and adaptation of their exploitation by MABS agents, as opposed to cognitional artifacts which expose all aspects of their functionality. The agent model is based on Acay *et al.*'s artifact capability theory [4]. Accordingly, it extends the theory to include learning and adaptation. Some aspects of the representations have been previously published in our included studies. We also present the underlying methodologies that will be utilized to implement the agent's learning and adaptation strategies and the technology used for implementing our MABS, its agents and artifacts.

3.1 Artifact Representation

In Omicini *et al.*'s artifact theory, an artifact is any object in the environment that can provide functionality and is defined in terms of three properties to facilitate its exploitation: a usage interface (UI), function descriptions (FD) and operating instructions (OI). *UI* defines the operations that are permissible on the artifact. *FD* facilitates artifact selection by identifying the services provided by the artifact, that is, what the artifact can be used for. *OI* specifies the instructions for successfully using the artifact to realize an element of *FD*. Since MABS agents are expected to learn and adapt artifact selection and use, artifacts should not expose *OI*. *FD* could be left out as well, in which case agents would have to determine an artifact's service without any information. However, we include *FD* with some information that should assist agents in the artifact selection learning process.

An artifact t is defined as:

$$t \triangleq \langle UI_t, FD_t \rangle \tag{3.1.1}$$

where UI_t represents its usage interface and FD_t constitutes its function descriptions.

3.1.1 Usage Interface

UI is essential for the agent to interact with the artifact and needs a representation that facilitates the learning and evolutionary process. To accomplish this, *UI* is reduced to a set of variables whose values can be learned. Motivated by the notion that an artifact is an object with one or more parts that provide functionality [7] *UI* defines an artifact's structure as consisting of parts, each of which has functional attributes with finite predefined domains.

UI_t for an artifact t is defined as: $UI_t = P_t$ where P_t constitutes the parts of the artifact with each part $p_t \in P_t$ defined as:

$$p_t \triangleq H_{p_t} \tag{3.1.2}$$

p_t is specified in terms of a functional attribute set H_{p_t} with each functional attribute $h_{p_t} \in H_{p_t}$ defined as:

$$h_{p_t} \triangleq \langle UD, b \rangle \tag{3.1.3}$$

where $UD = \{x \mid x \in [l, u], x \in \mathbb{R}, l \leq u\}$. l and u specify the lower and upper inclusive boundaries of the domain UD of a functional attribute's possible values. An additional element of the tuple $b \in \{0, 1\}$ indicates a visibility property which specifies whether the applied value of a functional attribute is visible to other agents. This aspect is utilized in one type of learning strategy that will be provided.

3.1.2 Function Description

FD specifies the services that the artifact provides. We are interested in agents learning to select the proper artifacts, therefore we define *FD* to expose only enough information to facilitate this process. We assume that an artifact exposes categorical information. Part of this knowledge is described in the A&A model as the artifact's intended use or the external goal that motivated its creation [80]. For instance each artifact in a set of *writing-tools* may expose its external goal as an artifact used for *writing*. When presented with the set the agent would need to learn which *writing-tool* is useful for writing under certain conditions, for instance *writing* on *stone* or on *paper*. The artifact *chalk* may provide good results on a *stone* and not do so well on *paper* while the opposite may be the case for a *pen*. We refer to these other objects (*stone* or *paper*) as criteria objects, since their characteristics provide the criteria for appropriate artifact selection. It should be noted that criteria objects could be artifacts or even agents. In fact, they need not even be used with the artifact. For example, selecting to drive a convertible car could depend on characteristics of the weather. What is important is that criteria objects provide the properties that define conditions under which an artifact will be selected.

The criteria object abstraction is an information model that provides physical or descriptive attributes that can be used to learn the conditions under which an artifact can be selected to realize a goal. Given a set of criteria objects CR, a criteria object $cr \in CR$ is defined as:

$$cr \triangleq \langle c_{cr}, Q_{cr} \rangle \tag{3.1.4}$$

where c_{cr} indicates a name or identifier for the object and Q_{cr} is a set of physical attributes with each physical attribute $q_{cr} \in Q_{cr}$ defined as:

$$q_{cr} \triangleq \langle q, SD \rangle \tag{3.1.5}$$

where q specifies the name of the physical attribute and $SD = [l, u] \wedge l, u \in \mathbb{R} \wedge l \leq u$ describes the domain of the physical attributes possible values.

The artifact's FD can now be defined in terms of both its external goals and criteria objects. FD_t defines t's set of function descriptions, with each function description $fd \in FD_t$ defined as:

$$fd \triangleq \langle xg_{fd}, d, CR_{fd} \rangle \tag{3.1.6}$$

where xg_{fd} denotes an external goal of the artifact and $d = \{0, 1\}$ specifies if the artifact's outcome or behavior is unpredictable with respect to the function description. The predictability of an artifact for a particular service is based on its dynamic nature concerning whether the artifact produces the same effect for a particular action over time. For instance, an agent learning to use a seat belt artifact for an adult can assume that throughout a simulation run an action such as attaching the seat belt securely will always result in the same positive outcome. A pen however may lose ink over time and no longer produce a positive outcome for writing. This distinction is important for adaptive agents as it lets the agent know whether to consider the best results so far for a particular action, or the latest results. Predictability may also be related to the heterogeneity of the artifact. This concerns whether agents that perform the same action at the same time with the same type of artifact can obtain different results.

CR_{fd} indicates the set of criteria objects. With the above example both artifacts *chalk* and *pen* have a function description $<$*writing-tool,{stone,paper}*$>$. Criteria objects *stone* and *paper* may be described in terms of a single physical attribute: $<$*coarseness,*[*0,200*]$>$ if we assume that its coarseness which can be measured from 0 to 200 determines how well a *writing-tool* will perform with it. The agent can now learn when to select *pen* versus *chalk* for writing on *stone* and *paper.* For instance, the agent could learn that if *coarseness=[10,50]* then it is better to use the artifact *pen,* or that if *coarseness=[190,200]* neither artifact produces good results. If the weather was a factor in the selection of the artifact, *weather* could be represented as a criteria object with a physical attribute *heat-index.* The agent can now use values of the *heat-index* domain in learning selection for the artifact.

3.2 Agent Representation

In Acay *et al.*'s artifact capability theory, an agent that can reason about artifacts is defined as a BDI agent, that is a rational agent that selects and executes plans according to its beliefs, goals and existing library of plans. The theory abstracts artifact capabilities to refer to those plans that specify artifact functionality for realizing a goal. Figure 3.2.1 shows a rational agent that interacts with artifacts according to their theory.

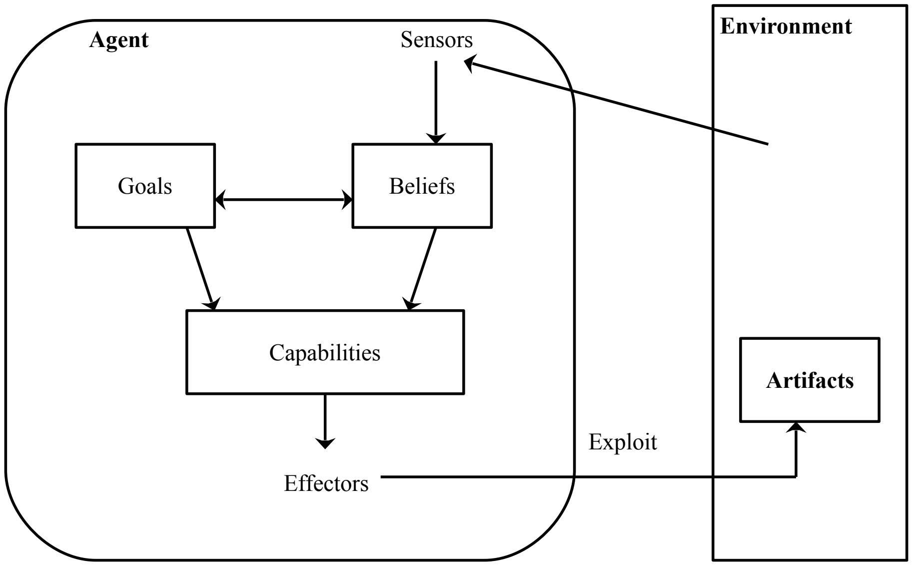

Figure 3.2.1: Agent with artifact capabilities © [2011] IEEE

The agent has goals, beliefs and capabilities which it can utilize to exploit artifacts. Resulting effects of applying actions belonging to capabilities are received by the agent through sensors and used to deliberate on what to do next. We extend the BDI-based artifact capability theory to support learning and adaptation by integrating the agent's cognition in the existing theory into Russell and Norvig's [103] general model for learning agents in AI.

According to Russell and Norvig the cognition of a general learning agent is comprised of a performance element (PE), a learning element (LE), a critic (CE) and a problem generator (PG). PE is responsible for deliberating and choosing the agent's actions which would represent the entire cognition of agents that act without learning from experience. CE evaluates the agent's actions with the help of resulting percepts received through sensors measured against an external predefined standard of performance (PS). Russell and Norvig argue that PS must be outside the agent's cognition to prevent the agent from adjusting the standard to match its behavior. CE provides feedback on the agent's performance to LE which is responsible for improving PE so that its actions yield better results in the future. LE suggests learning goals to the

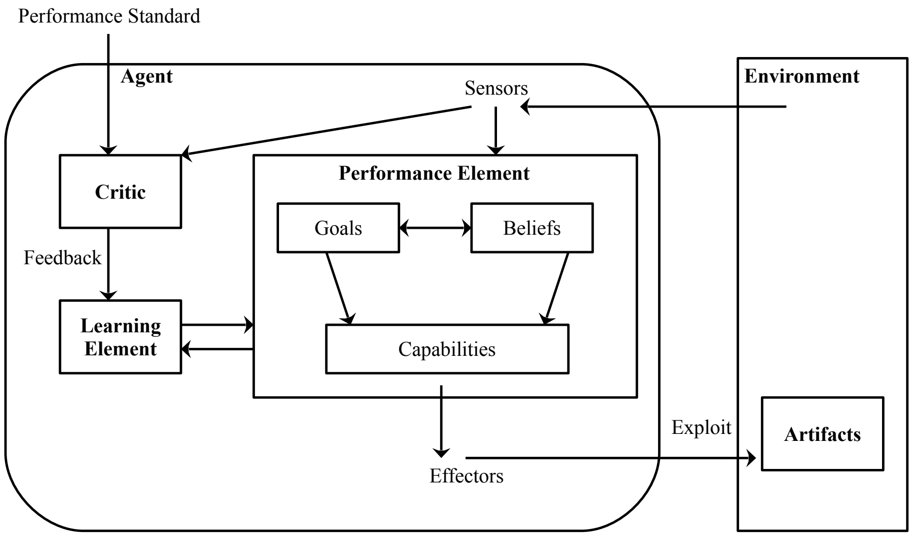

Figure 3.2.2: An artifact capability-learning agent © [2011] IEEE

final component *PG*. *PG* is an exploratory component which offers suggestions to *PE* on trying out new experiences.

Our model for an artifact capability-learning agent is shown in Figure 3.2.2. The agent ag is therefore a tuple:

$$ag \triangleq \langle PE_{ag}, CE_{ag}, LE_{ag} \rangle \tag{3.2.1}$$

PG has been deliberately omitted since our model will promote exploration on its own. The agent's beliefs, goals and capabilities make up *PE* representing the deliberation and decision making aspect of the agent. *PE* is therefore the equivalent of the artifact capable agent in Figure 3.2.1. Learning artifact capabilities primarily involves learning strategies developed by *LE* for the improvement of the capability component of *PE*. Since several different learning strategies will be provided, different representations of *LE* will be presented in subsequent chapters along with *PS* which is a domain dependent feature. Representations for *PE* and a *CE* are provided next.

3.2.1 Performance Element

PE of an artifact capability-learning agent ag is formerly defined as:

$$PE_{ag} \triangleq \langle G_{ag}, C_{ag}, B_{ag}, a_{ag} \rangle \tag{3.2.2}$$

Goals

G_{ag} is the agent's set of goals. Each goal $g_{ag} \in G_{ag} = \langle gid, st \rangle$ simply has a name gid and maintains its active status: $st = \{0, 1\}$.

Capabilities

C_{ag} denotes the agent's artifact capability set which according to Acay *et al.* [4] is the union of its inherent internal capabilities and its external capabilities: $C_{ag} = IC_{ag} \cup AC_{ag}$. Since the only external capabilities addressed in this thesis are artifact capabilities AC_{ag} refers to the agent's set of artifact capabilities. For simplicity we drop the agent subscript ag in the subsequent formulas. An artifact capability for the agent $ac \in AC$ is defined as:

$$ac \triangleq \langle gid, T_{ac}, CP_{ac} \rangle \tag{3.2.3}$$

where gid is a goal of the agent, T_{ac} represents a set of artifacts, each of which can be used for realizing the goal and CP_{ac} consists of the plans for exploiting those artifacts. The capability plan set is defined as:

$$CP_{ac} \triangleq \langle SP_{ac}, UP_{ac} \rangle \tag{3.2.4}$$

where SP_{ac} denotes plans for artifact selection and UP_{ac} specifies plans for artifact use.

Artifact Selection Plan A selection plan $sp \in SP_{ac}$ is defined as:

$$sp \triangleq \langle K_{sp}, s, y_{sp} \rangle \tag{3.2.5}$$

where K_{sp} specifies the knowledge for artifact selection, s is a function for choosing a specific artifact when the applied knowledge results in more than one artifact and $y_{sp} \in \mathbb{R}$ is an associated score attributing a utility to how good the selection is at realizing the goal. K_{sp} is a generalization inspired by the knowledge structures for restraint selection defined in Kobti *et al.* [50]. K_{sp} is defined as $K_{sp} = \{kn_1, \ldots kn_z\}$ where $kn_j \in K_{sp}$ is a unit of knowledge describing artifact selection for a criteria object, one of its physical attributes and one or more corresponding physical attribute value ranges. These criteria objects are the objects specified as part of the artifact's FD as per Formula (3.1.6). Artifacts in T_{ac} that have a common FD with the same criteria objects can be used to form a selection plan. Depending on the problem domain, a unit of knowledge may be defined for different ranges of all physical attributes of all these criteria objects or a relevant subset of them. A unit of knowledge $kn_j \in K_{sp}$ is defined as follows:

$$\begin{aligned} kn_j\left(c,q\right) \;\; &\triangleq \;\; \langle \left[l_1\left(c,q\right), u_1\left(c,q\right)\right], b_0, b_{t_1}, \ldots, b_{t_e} \rangle \\ & \;\;\;\;\;\; \langle \left[l_2\left(c,q\right), u_2\left(c,q\right)\right], b_0, b_{t_1}, \ldots, b_{t_e} \rangle \\ & \;\;\;\;\;\; \mid \\ & \;\;\;\;\;\; \langle \left[l_o\left(c,q\right), u_o\left(c,q\right)\right], b_0, b_{t_1}, \ldots, b_{t_e} \rangle \end{aligned} \tag{3.2.6}$$

where $l_i\left(c,q\right)$ and $u_i\left(c,q\right)$ define lower and upper all inclusive ranges for criteria object c and its physical attribute q. The ranges are assumed to be in ascending order and no ranges overlap: $l_1 \leq u_1 < \ldots < l_o \leq u_o$. The bit sequence $b_0, b_{t_1}, \ldots, b_{t_e}$ denotes bit values for a bit string where $b_{t_x} = \{0, 1\}$ represents the selection or non-selection of an artifact in the artifact subset of e artifacts chosen from T_{ac}. The bit string is prepended with an additional bit b_0 to accommodate knowledge for the selection of no artifact. The idea is that if an agent believes that artifact t_x should be selected when the physical attribute q for criteria object c has value v, then $b_{t_x} = 1$ in the tuple containing the range that v falls within, otherwise $b_{t_x} = 0$. If the agent believes no artifact should be selected given that criteria then $b_0 = 1$ otherwise $b_0 = 0$. Revisiting the example given in the artifact representation, *stone* and *paper* are the criteria objects with physical attribute *coarseness* while *chalk* and *pen* are the artifacts. With three bits representing artifact selection in the sequence no artifact, chalk, pen, the partial unit of knowledge $kn\,(paper, coarseness) = \langle [10, 20]\,, 001\rangle$ specifies the agent's knowledge to select a pen when the paper has a coarseness between 10 and 20.

In order to choose an artifact given a particular criteria object the agent applies its selection knowledge. For a specific criteria object C, the result is a set of bit strings: $BS_C = \{bs_{C,1}, \ldots, bs_{C,z}\}$ denoting z bit strings, extracted from each range of a unit of knowledge within which C falls based on its physical attributes values. In order to produce a final bit string that can be used by the agent to select a single artifact t_x, a domain dependent artifact assignment function may be necessary, specified as the second tuple element in sp_{ac} and defined as:

$$s : BS_C \rightarrow t_x \tag{3.2.7}$$

Artifact Use Plan An element of the use plan set $up \in UP_{ac}$ is defined as:

$$up \triangleq \langle t, UA_{up}, y_{up}\rangle \tag{3.2.8}$$

where $t \in T_{ac}$ denotes the artifact, $UA_{up} = \langle ua_1, \ldots, ua_k\rangle$ is a sequence of k use actions and $y_{up} \in \mathbb{R}$ is a score associated with the realization of the goal by the plan. A use action $ua_j \in UA_{up}$ is defined as:

$$ua_j \triangleq \langle V, r, y \rangle \tag{3.2.9}$$

where V is a combination of functional attribute values, $r \in \mathbb{N}$ denotes the social network radius for social learning agents that evolve the extent of their social network. The final tuple element $y \in \mathbb{R}$ indicates a score attributed to the specific use action once its applied and evaluated. V denotes a selected functional attribute value for each of the artifact t's functional attributes, defined as:

$$V \triangleq \{ \langle p, h, v \rangle \mid p \in P_t \wedge h \in H_{pt} \wedge l_{UD_h} \leq v \leq u_{UD_h} \} \tag{3.2.10}$$

With t defined using Formulae (3.1.1,3.1.2 and 3.1.3), V is specified in terms of its associated artifact part p and functional attribute h. The functional attribute value v is constrained by the functional attributes domain UD_h. Only one value is selected for a functional attribute therefore the functional attribute of an artifact part only appears once in the use action. The number of elements in V is the cumulative total of functional attributes belonging to the artifact. An agent ag's knowledge structure for an artifact use plan up that specifies the functionality for using artifact t in order to achieve goal g can be viewed as:

$$
\begin{array}{ll}
up\,(g,t) = & \{ \\
ua_1 = & <<p,h,v>_1 \;\ldots\; <p,h,v>_{\mathrm{n}}, r_{ua_1}, y_{ua_1}>, \\
ua_2 = & <<p,h,v>_1 \;\ldots\; <p,h,v>_{\mathrm{n}}, r_{ua_2}, y_{ua_2}>, \\
 & \qquad\qquad\quad | \\
ua_{\mathrm{k}} = & <<p,h,v>_1 \;\ldots\; <p,h,v>_{\mathrm{n}}, r_{ua_k}, y_{ua_k}> \\
 & \}
\end{array}
$$

for n total functional attribute values and k use actions, where $\langle p, h, v \rangle$ associates one of t's parts, one of the part's functional attributes and a single value chosen from its domain.

The score of use plan y_{up} is simply the average score over all the use actions, that the average over the use action y_{ua} values.

Beliefs

The agent's belief set B_{ag} can maintain failed use actions for an active goal. This can be used by the learning agent to avoid repeating failed actions. The maintenance of such beliefs will depend on the learning strategy being implemented by the agent. It is assumed that agents when learning artifact use the agent only learns one action at a time. A belief element $b_{ag} \in B_{ag}$ is defined as:

$$b_{ag} \triangleq \langle t, gid, ua_l \rangle \tag{3.2.11}$$

where t is an artifact used towards goal gid and ua_l is a use action that was unsuccessful.

Action Generation Function

The final element of PE specifies the agent's action generation function a. This involves the selection and use of an artifact for the agent's goal. It is defined as:

$$a : G_{ag} \times B_{ag} \times C_{ag} \rightarrow ua \tag{3.2.12}$$

indicating that the agent uses its goal, belief and capability set to generate the action to perform with an artifact it selects.

3.2.2 Critic Element

CE is responsible for evaluating the perceived results of the agent's use action against an external predefined PS in order to provide feedback to the LE on the agent's progress. PS is domain dependent and the type of learning strategy being employed by LE may play a role in CE's evaluation function. One possibility is that sensors only indicate that the action was performed, PS provides the proper values to measure the action's attribute values against and CE defines a fitness function that evaluates the action against PS assigning it a utility. The fitness function would be defined as:

$$f : PS \times ua \rightarrow y \tag{3.2.13}$$

where the performed action ua measured against PS yields a fitness score y for the action. Another possibility is that the sensors provide the fitness score y, PS indicates good and bad scores which CE uses to classify the score. For instance an agent attempts to write with a pen, perceives how much it has written and the standard indicates if that is good enough. Different examples of CE will be provided in subsequent chapters.

3.2.3 The Learning Problem

Artifact Selection Learning Problem The problem that an agent learning artifact selection is trying to solve can be defined using definitions (3.2.3, 3.2.4 and 3.2.5). Given a set of artifacts R consisting of artifacts that can be used for realizing an active goal gid and a criteria object for each criteria object category in each artifact's FD, determine an $sp \in SP$ with an acceptable y_{sp} score. A score is obtained by using the assignment function s in Formula (3.2.7) to choose an artifact $t \in T$ with the knowledge K_{sp}, applying a use plan for the capability and obtaining feedback.

Artifact Use Learning Problem The use learning problem an agent is trying to solve can be defined using definitions (3.2.3, 3.2.4 and 3.2.8): Given artifact r that can be used for an active goal gid, find $up \in UP$ composed of the use action sequence UA such that its score is acceptable. If 1 is considered a good enough score for an action, then definitions (3.2.9 and 3.2.10) requires that the agent find each use action $ua_j \in UA$ such that applying its values V and r if in use, results in $y_{ua_j} \geq 1$, that is, the selected combination of functional attribute values for the use action are successful. As a result the agent successfully learns or evolves one of the artifact capability plans useful for its goal.

3.2.4 Methodologies for the Learning Element

Evolutionary computation (EC) methods will be used to realize the reward based learning strategies that agents will employ through the *LE* component in order to learn artifact selection and use. EC constitutes a family of techniques for automated problem solving inspired by Darwin's principles of evolution and natural selection [29, 47], that include evolutionary algorithms and other population-based algorithms such as cultural algorithms. A basic evolutionary algorithm (EA) begins with a randomly generated population of individuals or candidate solutions to the problem. After each individual is evaluated and given a fitness or quality assessment the EA applies evolutionary operators such as selection, reproduction, recombination and mutation to produce subsequent generations. Each successive generation is expected to improve the population and the EA runs until a designated time limit or an adequate solution is found. Evolutionary algorithms include genetic algorithms (GA) [42] and evolution strategies (ES) [14] both of which are usually used for finding solutions in multidimensional parameter spaces. Genetic Programming (GP) another kind of EA evolves computer programs [60]. GAs are used by the artifact capability learning agents in some of the learning strategies along with cultural algorithms (CA) and multi-population cultural algorithms (MPCA).

Another way that agents can learn to exploit artifacts is through direct communication with other agents. A social network defines connectivity between individuals in a population. Agents may evolve their knowledge for artifacts through influence from other members of social networks that they belong to.

3.2.4.1 Genetic Algorithms

Considered the most popular EA, GAs were introduced by Holland [42]. Although they are used for problem solving Holland's original objective was to understand adaptation in nature and determine a means for integrating the concept into computer systems. In a GA candidate solutions or chromosomes are usually encoded as bit strings or integers, although representations using real values also exist [72]. GAs primarily include three operators: selection, crossover and mutation. The selection operator is used to select candidate solutions for reproduction which usually depends on the fitness of the chromosome. A popular method for selection is roulette wheel in which a candidate solution's chances for selection is proportional to its fitness. Fitter solutions have a greater chance of being selected. Another method is tournament selection where "tournaments" are conducted among a few randomly chosen solutions with the winners selected for reproduction. Analogous to biological recombination, crossover chooses one or more points and exchanges the bit sequence of the rest of the string or between those points. For instance, given two bit strings 11001 and 10010 a single point crossover at the 4th bit would yield two new offsprings: 11011 and 10000. A crossover rate is usually used to define a probability that crossover occurs. Finally, mutation flips bits in the chromosome. This can also happen according to a mutation rate which defines the probability that a bit (in the case of a binary representation) is mutated. For instance, flipping the third bit of the bit string 11001 would yield 11101.

The pseudo-code of a basic GA is depicted in Algorithm 1.

A basic GA begins with a randomly generated population or pool of candidate solutions to the problem. Each candidate solution is then evaluated and attributed a fitness. The selection operator is applied to select candidates for reproduction. Crossover and mutation are applied to selected candidates in order to breed a new generation of candidate solutions. The new generation is then evaluated and the pro-

Algorithm 1 Pseudo-code for a basic genetic algorithm (GA)

```
Begin

    Generate initial random population of candidate solutions
    Evaluate fitness of population
    repeat

        Select, crossover, mutate to breed new population
        Evaluate fitness of new population
    until termination criteria

End
```

cess continues until some termination criteria is reached. Often the GA terminates when a given number of generations is reached or a suitable solution is found. As is the case with other stochastic search methods, GAs do not guarantee optimal solutions. They are however well suited for finding good solutions to a wide variety of problems. With respect to artifact selection and use, GAs are used primarily for individual learning, learning through observation and learning through social networks.

3.2.4.2 Cultural Algorithms

Introduced by Reynolds [95, 96] cultural algorithms (CA) are computational models of cultural evolution. A CA is characterized by a population space and a belief space connected via a communication protocol. The population space may consist of social agents and is usually implemented with any EA such as a GA. Selected individuals from the evolving population contribute their experiences to the belief space through an acceptance function. The belief space maintains these experiences as categories of knowledge sources, which can be used to influence the evolution of the individuals in the population space by means of an influence function. The interaction and support that occurs between the population and belief space components, a sort of dual inheritance, is considered similar to the evolution of human culture [94, 99].

Five categories of knowledge sources have been identified to characterize the belief space component [84]: situational, normative, topographic, historical or temporal and domain. Situational knowledge constitutes the best performers in the population referred to as the exemplars. Normative knowledge maintains encouraging variable ranges and can help individuals leap into good ranges. Topographical knowledge refers to spatial characteristics of the search space. Historical or temporal knowledge constitutes important events or temporal patterns during the search process. Domain knowledge is knowledge specific to the domain of the problem being addressed by the CA. The knowledge sources can be used selectively or collectively to guide the search process of the CA. The CA framework facilitates extracting, storing and exploiting experiences in a population of individuals over time thus permitting self-adaptation and learning at various levels in an evolving model [52, 99]. CAs provide a way to model cultural evolution of artifact exploitation.

CAs have been applied to solve a variety of optimization problems including unconstrained optimization [24, 98] and constrained optimization [10, 25, 46]. They have also been used to build complex social systems [52, 53, 50, 76]. CAs have undergone some extensions such as multi-objective CAs [13, 26, 94] proposed for solving multi-objective optimization problems and multi-population CAs (MPCA).

3.2.4.3 Multi-Population Cultural Algorithms

MPCAs were introduced by Digalakis and Margaritis [27] to address the scheduling of electrical generators. Although the primary characteristic of an MPCA is that it involves multiple independently evolving populations rather than a single one as in a CA, MPCAs can take on different forms. For instance in Digalakis and Margaritis [27] a global "master" creates and manages the evolution of sub populations that are embedded in local CAs. The local cooperative CAs share knowledge about the best performers extracted from their respective sub populations. Alami *et al.*'s [5] proposed approach which involved information exchange between the belief spaces of local CAs was explicitly modeled in Guo *et al.* [73]. The authors provided details on implicit knowledge migration between CA belief spaces arguing for their effectiveness when compared to the knowledge exchanged through the selection of best individuals from population spaces. In Guo *et al.* [36] a global belief space extracts knowledge from subpopulations and shares it with individually evolving subpopulations. Hlynka and Kobti [40] offered an MPCA where the individuals evolved knowledge algorithms and migrated between subpopulations to transfer their knowledge of successfully applied algorithms. In their proposed algorithm the Transfer Agent Multi-Population Cultural Algorithm (TAMPCA), randomly selected agents in different populations swapped places taking their currently used knowledge algorithm with them. As a result the new knowledge influenced the evolution of their new population. For artifact exploitation, MPCAs provide the opportunity to examine artifact exploitation evolution at a multi-population level.

3.2.4.4 Social Networks

A social network defines relationships between social individuals in a population, for instance a network of friends, colleagues, neighbors and so on. When viewed as a graph the social individuals can be represented as nodes with dyadic ties between them. Nodes in social networks are characterized by their degree of connectivity and clustering coefficient [39]. While a node's degree of connectivity is the number of connections or links it has to other nodes, the clustering coefficient also referred to as density is the extent to which linked nodes are linked to others. The latter refers for example, to the extent to which one's neighbors are neighbors of each other. Four types of social network models are usually found in ABMs: regular lattice, random, small world and scale-free [39]. In regular lattice networks, each node has the same degree of connectivity. In random networks, a random variable is used to create connections between nodes. In small world networks the majority of nodes are connected to their nearest neighbors. Finally, the probability distribution of the degree of connectivity in scale-free networks follows a power law. In one well known example of scale-free networks, networks are created using preferential treatment where new nodes are connected to existing nodes with many links [9].

Agents can learn artifact selection and use with any type of social network model. The networks can be static or dynamic during the evolutionary process. Random and regular networks will be used to generate networks in the MABS in this thesis however when integrated into existing MABS it is possible to use any existing social networks for propagation of artifact exploitation knowledge.

3.3 Implementing Agents and Artifacts

Agents and artifacts in this thesis will be implemented in MABS systems built with the "Recursive Porous Agent Simulation Toolkit" (Repast) [1], a commonly used cross platform, open source and free agent-based modeling and simulation toolkit. Repast has many valuable features such as a fully object-oriented architecture, concurrent and discrete event scheduler, support for social networking tools, built-in libraries for various algorithms such as GAs and neural networks, graphing and output gathering tools. Although Repast provides algorithms such as GAs and social networking tools we have built our own algorithms and designed the social networks. Repast is available in several languages including C++, Python, .NET and Java. We have used Java based version of Repast. Repast Simphony is used for the MABS models that we construct while Repast J is used in the existing Village MABS model that is employed as a case study. The distinction between the two as it relates to our work lies in additional features provided by Repast Simphony for simplifying the creation and manipulation of the agents and the environment. In particular Repast Simphony provides a graphical interface equivalent to several method calls for setting up the model in Repast J.

Chapter 4

An Individual and Observational Learning Model for Artifact Use

In this chapter a model for individual and observational learning of artifact use is provided. The model uses the artifact and agent representations from Chapter 3 and describes two strategies implemented by the *LE* component of the agent's cognition. The work has been previously published in Mokom and Kobti [63]. The model addresses only artifact use therefore agents are assumed to know the proper artifact to select for an adopted goal but need to learn one way to use it. Hence artifact selection plans are assumed to exist while the agent needs to learn artifact use plans. The learning strategies are developed using genetic algorithms (GA) with observational learning chosen to represent a form of social learning. In this version of the learning model the domain of artifact functional attribute values is restricted to integers. A generic MABS is built to conduct experiments comparing the learning strategies and demonstrate agents learning an artifact capability from observations of their own behavior and from observing others in their environment.

4.1 Performance Standard

Although external to the agent, *PS* is relevant to the artifact capability-learning agent as shown in Figure 3.2.2. Playing the role of a friendly teacher *PS* in this version of the model relays to the agent the number of use actions needed to realize its goal as well as the correctness of the attempted use actions. This is used by *CE* in evaluating the results of the agent's actions. In the model sensors are implicitly used, that is they are assumed to simply inform the agent that the action has been performed. It is assumed that the agent will always be able to complete the action and focus on evaluation conducted by the critic as the primary feedback mechanism.

Two possible forms of *PS* are defined for the agent. A *fixed-value* standard provides a single value that the chosen functional attribute value is measured against and a *range-of-values* standard specifies a subset of a value's domain, a range within which the chosen value is expected to fall inclusively. It should be noted that these are just examples of *PS* and that different kinds can be defined depending on the problem domain. This will be further evident in case studies presented later on.

4.2 Critic Element

CE is responsible for comparing the chosen values in the agent's use action against the available *PS* and providing feedback to *LE* on the agent's progress. To determine the utility of the result *CE* applies a simple distance measure as a fitness function that averages over all attribute values to determine a fitness score for the use action. Once again, the fitness functions defined here are designed specifically to work with the given *PS* above. The distance measure function is provided for each type of *PS*. Given a use action reduced to its values and specified in a predefined functional attribute sequence $ua_j = \langle v_1, \ldots, v_n \rangle$:

The distance measure for a *fixed-value* standard with a standard value for functional attribute i given as ps_i is calculated using the following function:

$$g\left(v_i, ps_i\right) = \begin{cases} 1.01, & v_i = ps_i \\ \frac{1}{|ps_i - v_i|}, & otherwise \end{cases} \tag{4.2.1}$$

The distance measure for a *range-of-values* standard with a standard range for functional attribute i given as $ps_i = [l_{ps}, u_{ps}]$ where l_{ps} is the range's lower bound and u_{ps} is the range's upper bound, is calculated using the following function:

$$g\left(v_i, ps_i\right) = \begin{cases} 1.01, & l_{ps} \leq v_i \leq u_{ps} \\ \frac{1}{|l_{ps} - v_i|}, & v_i < l_{ps} \\ \frac{1}{|v_i - u_{ps}|}, & v_i > u_{ps} \end{cases} \tag{4.2.2}$$

The mean fitness score for the use action ua_j and standard PS is calculated as follows:

$$f(ua_j, PS) = avg\left(\sum_{i=1}^{n} g\left(v_i, ps_i\right)\right) \tag{4.2.3}$$

Functional attribute values that do not have a defined standard and therefore do not contribute to the success or failure of the action are ignored by CE in the evaluation process.

The feedback CE offers to LE includes whether the goal has been achieved (the current action succeeded and there are no more actions to learn), or the current action succeeded and the agent needs to learn the next action. In the event that the action failed, there are two possible feedbacks that CE provides to LE. CE either advises LE on the failure of the action or it tells LE the fitness of the failed action. According to the distance functions, a successful action will result in a score $f(ua_j, PS) > 1$ which would correspond to the use action score y_{ua_j} of the use action of an artifact use plan as defined in Formula (3.2.10).

4.3 Learning Strategies for Artifact Use

Agents learning and evolving artifact use in the model utilize individual or social experiences in the process under the guidance of feedback from the critic. Given the artifact use learning problem in Section 3.2.3, the agents learn by maintaining a history of failed actions avoiding their repetition or by also employing reward-based learning using GAs. *LE* is responsible for developing these learning strategies and using them in collaboration with *PE* towards augmenting *PE*'s performance. To simplify the representation of a use action being learned by *LE,* use actions are reduced to their values specified in a predefined functional attribute sequence: $ua_j = \langle V, y\rangle$ where $V = \langle v_1, \ldots, v_n\rangle$ for n functional attribute values with y as the evaluated score. *CE* can provide two types of feedback to *LE*: non-utility feedback and utility feedback.

Non-utility Feedback An agent can select its use action simply by keeping track of unsuccessful actions that it has previously attempted. The critic evaluates the agent's performed action and *LE* is only advised on the success or failure of the action. The agent maintains a historical knowledge of failed attempts in its belief and selects subsequent use actions made up of functional attribute value combinations it has not yet tried. Agents that use this memory-based method for selecting actions do not have any utility attributed to the result of their actions. In other words, *CE* does not report on the fitness of the performed action and the agent as a result is unaware of how badly the action fails.

Utility Feedback Agents can also learn by obtaining a better evaluation of their actions. As with the non-utility option, the agent maintains a history of failed attempts in its belief. In addition the agent also maintains a score or fitness of each failed action. In choosing a use action the agent selects and modifies a single attribute value of a failed action selected based on its fitness. *CE* uses fitness functions to test a performed action against *PS*, with the action score dependent on its proximity to successful values. Agents that use the fitness-based action selection method are aware of both the success and failure of chosen actions as well as the utility of actions that fail.

4.3.1 Individual Learning

Individual learning involves agents learning from observations of only their own behavior. The agent learns as though it existed in a single-agent system. These agents can learn with either goal or utility feedback. *LE* formulates a learning goal g for *PE* to pursue that constitutes learning to use the artifact t. If *LE* learns with utility feedback, it randomly generates an initial pool of a predefined number x use actions. *PE* initializes the belief set and a use plan: $B = \emptyset$, $up = \langle \emptyset, -\infty \rangle$, an empty belief and a new capability plan with no actions and an undefined score. *PE* randomly generates a use action ua_j for up (with $k = 1$ when learning the first action) or is offered one by a utility-based *LE*, and applies it. *CE* evaluates ua_j against the available *PS* and provides feedback to *LE*. Regardless of the feedback *PE* will only generate actions it has not tried before, that is for any new action generated ua_j, $\langle t, g, ua_j \rangle \notin B$.

If *LE* is learning from non-utility feedback and the action failed *LE* advises *PE* to add the failed action to its belief: $B \cup \langle t, g, ua_j \rangle$, and randomly generate a functional attribute value combination for ua'_j that it has not been attempted before. If the action succeeded *LE* advises *PE* to update the action score of ua_j in up(any score that indicates success), and reinitialize the belief set. If the success meant the agent has reached its goal, *LE* advises *PE* to calculate the average score for up, and inactivate goal g. If the goal is not yet achieved and there are more actions needed, *LE* advises *PE* to generate a new action ua_{j+1} and learning continues. A use plan up with a score of $-\infty$ indicates that it is incomplete since the average score over all use actions is only calculated when the goal is achieved.

The search space of the algorithm is a function of the number of functional attributes and the performance standard. When evaluating a use action against a fixed standard for example, it is necessary to compare each generated value with its associated standard value in order to obtain its fitness. Therefore the evaluation grows linearly with the number of attributes.

Genetic Algorithm In order to learn from utility feedback, *LE* employs a GA. This is sufficient for *LE* since it can advise *PE* on the generation of new use actions based on the utility of previous attempts. Although GAs do not guarantee an optimal solution, the agent is only interested in finding *one way* to successfully use t to realize g, not necessarily the best way. The GA uses a binary representation for candidate solutions. A candidate solution is a use action's values $V = \langle v_1, \ldots, v_n \rangle$ where v_i is a bit sequence equivalent to an integer value drawn from functional attribute i's domain. Each candidate solution will be given a score once evaluated. Given the number of pool solutions x, the GA begins by generating a random pool of x use action values and *LE* converts one in the pool to its equivalent integer values and offers it to *PE*. When *LE* receives utility feedback from *CE* with respect to action ua_j it assigns the received fitness to ua_j as y and continues to offer use actions to *PE* as long as actions are unsuccessful until the pool is exhausted. The GA then uses roulette wheel selection to choose two candidates at a time for reproduction. Genetic operators crossover and mutation are applied to the solutions using given rates. Crossover is applied by randomly choosing a single attribute then applying two point crossover to swap its bit values. Mutation is applied to the bits in the solution according to the mutation rate. *LE* communicates with *PE* to ensure that newly generated solutions are only added to the new pool if they are not an element of B and the generated values are within the domain of the respective attribute. Once the pool is regenerated, LE proceeds with offering a converted solution as a use action to PE. If an action succeeds LE clears the solution pool and randomly generates a new one if the goal is not achieved.

4.3.2 Social Learning via Observation

The artifact-capability learning agents can also benefit from learning in the presence of others using a social learning model. Agents employ a form of observational learning. The idea is that the learning agent has observed another agent performing the capability it wishes to learn. As a result the agent is able to duplicate some of the knowledge and commence learning with prior information. According to Formula (3.1.3) a functional attribute h is defined to have a visibility property $b_h \in \{0, 1\}$. The property specifies whether an observing agent can copy a value chosen for h in another agent's use action and apply it with some certainty for success. The social learning agent learns only with utility feedback. Its CE operates in the same fashion as its counterpart in the individual learning agent. The distinction is in the variation of the GA that LE employs.

Seeded Genetic Algorithm It is possible for the initially generated pool of a GA to contain candidate solutions with seeded values [93]. This means that the agent commences the learning problem with partially successful solutions. With a use action that would mean the action is initialized such that some of its values when evaluated against the given PS would always be deemed correct. The seeded GA selects a value for each seeded value that falls within the standard values used by the PS while unseeded values are randomly generated from the attribute value's domain. The seeded GA applies the same genetic operators as the individual learning agent, however when regenerating the pool genetic operators are applied only to the values of non visible attributes. Seeded values of the solutions remain fixed throughout the learning process. The expectation is that learning would be accelerated in observational learning agents since they are built to commence the learning process with partial knowledge.

4.4 Generic MABS Model

The artifact use learning model is used to build a generic MABS to conduct experiments with the learning strategies. The MABS model is defined as: $S \triangleq \langle AG, R \rangle$ where AG is the population of agents and R is the set of artifacts in the environment.

4.4.1 Model Parameters

Parameters supplied to the model are either fixed or variable. Fixed parameters are the same for all experiments conducted while variable parameters differ between test cases. It should be noted that it is possible to run other experiments with different values even for the fixed parameters. All agents are assumed to be learning with one of the same type of artifact, although artifact types may differ between experiments. The following are fixed parameters:

NumberOfAgents The number of learning agents in the model that constitute the set AG. There are four types of agents differing according to their employed learning strategy, if any. Henceforth they will be referred to as AG_NOMEM, AG_MEM, AG_GA, and AG_SOCIAL. AG_NOMEM does not employ any form of learning. It randomly generates use actions, maintains no memory of applied actions and gets no feedback at all. AG_MEM and AG_GA are agents that utilize individual learning. AG_MEM agents use non-utility feedback generating actions that are yet to be attempted while AG_GA agents go further using utility feedback and employing a GA. AG_SOCIAL agents employ the observational learning strategy. There are 100 members of each type of agent resulting in a total of 400 agents in the model.

NumberOfArtifacts The number of artifacts in the model that constitute the set R. Each agent is given a single artifact, therefore there are 400 artifacts and all agents have the same type of artifact.

NumberOfArtifactParts The number of artifact parts. This is fixed at 1.

FunctionalAttributeDomain The domain of the functional attribute's values fixed at $[1, 100]$ for all attributes.

ArtifactFD The function description describing the service the agent is to learn. All artifacts expose a single FD with external goal: *use*, an empty set of criteria objects and the artifact is assumed to be predictable.

AgentGoal The agent's goal. All agents share the same goal which matches the external goal of the artifact: *use*.

GACrossoverRate After conducting experiments with various rates the crossover rate for the GA was chosen as 0.7.

GAMutationRate After conducting experiments with various rates, the mutation rate for the GA was chosen as 0.01.

NumberOfUseActions The number of use actions that need to be learned to realize the goal (specified as part of the performance standard) and fixed at 5.

The following are variable parameters:

NumberOfFunctionalAttributes The total number of functional attributes for the artifact.

NumberOfVisibleAttributes The number of visible functional attributes for the artifact.

PerformanceStandard Either the *range-of-values PS* defined to cover 20% of the functional attribute's domain or a single value chosen in the domain for the *fixed-value PS.* This is defined for each of the 5 required use actions.

GAPopulationSize The number of use actions in the GA pool of solutions. This is fixed at 100 for all experiments conducted with the *range-of-values* standard. For tests conducted with the *fixed-value* standard, a pool size of 200 is used for tests with 2-attribute artifacts and increased to 1000 for tests where artifacts have more than 2 attributes. This is to allowed for a more varied initial population for the *fixed-value* standard.

4.4.2 Simulation Flow

The environment is a simple 20 x 20 toroidal grid world, in which each square contains an agent and a single type of artifact. The general pseudo-code for the simulation steps of the agents given artifact t, goal g and performance standard PS is presented in Algorithm 2.

At the start of the simulation, each agent gets the artifact at its location. LE formulates a goal for the artifact for PE. PE initializes the belief set, a new capability to learn and activates the goal. At each simulation step, PE generates an action possibly with help from values generated by LE. CE uses the action and PS to provide feedback to LE which generates changes for PE using its learning strategy. PE applies the changes. The simulation is run until all agents succeed in achieving their goal, in other words learn the sequence of use actions that correspond to one way to use the artifact.

Algorithm 2 General pseudo-code for learning artifact use

```
Begin
    t = artifact at agent's location
    g = LE→formulate_goal(t)
    PE→initialize_belief
    PE→activate_goal(g)
    PE→initialize_capability(g,t)
    time_step = 0
    repeat
        action = PE→generate_action()
        feedback = CE→evaluate(PS,action)
        changes = LE→generate_changes(feedback)
        PE→apply_changes(changes)
        time_step = time_step + 1
    until PE→goal_achieved(g) = true
End
```

4.4.3 Experiments and Results

Test cases vary in the number of functional attributes, the number of visible functional attributes and the type of *PS* (*fixed-value* or *range-of-values*). All agents in each simulation run use the same type of artifact. The same random seeds are used to initialize the random number generator for each agent type to ensure that the agent types begin the evolution process equally. Many test runs were also carried out to ensure that results were consistent. This means that at the start of the simulation there should be one agent for each agent type with the same randomly generated initial population of solution. Tests were conducted for an artifact with a single part and 2, 4 and 8 attributes. The different number of attributes were tested against the two *PS*. For AG_SOCIAL agents, tests were run with 1, 2 and 4 visible attributes. Average convergence times for each type of agent were computed. These represented the average number of time steps needed by the respective agents to learn the artifact capability. Although time constraints make it impossible to account for all test cases, we believe the selected cases make it feasible to evaluate the agent's performance.

Table 4.1: Average Convergence Times for Artifact with 2 attributes (1 visible attribute for AG_SOCIAL) © [2011] IEEE

Agent Type	*fixed-value PS*	*range-of-values PS*
AG_NOMEM	52179.99	128.46
AG_MEM	25471.12	128.16
AG_GA	4526.51	111.44
AG_SOCIAL	248.74	25.74

Table 4.2: Average Convergence Times for Artifact with 4 attributes (1 visible attribute for AG_SOCIAL) © [2011] IEEE

Agent Type	*range-of-values PS*
AG_NOMEM	3153.08
AG_MEM	3153.07
AG_GA	2095.53
AG_SOCIAL	616.10

Obtained results are depicted in Tables 4.1, 4.2 and 4.3.

The numbers indicate the average number of steps needed by each agent type to achieve its goal of learning the artifact capability. In all cases results are presented for both *PS.* Table 4.1 shows the average convergence times for agents learning to use an artifact with 2 attributes. For AG_SOCIAL agents, one of the attributes was made visible. Table 4.2 presents results for agents using 4-attribute artifact and the *range-of-values PS*, with one made visible for AG_SOCIAL agents. Finally Table 4.3 shows average convergence times for AG_SOCIAL agents learning a capability for artifacts with 4 and 8 attributes with different degrees of visibility.

Table 4.3: Average Convergence for AG_SOCIAL agents © [2011] IEEE

# Attributes/ # Visible	*fixed-value PS*	*range-of-values PS*
4 / 1	31627.88	616.10
4 / 2	14849.59	122.30
8 / 2	233977.65	17510.96
8 / 4	168238.91	2639.40

4.4.4 Discussion

In accordance with our expectations all learning agents Ag_mem, Ag_ga and Ag_social outperformed the random agents Ag_nomem. In Tables 4.1 and 4.2, it can be observed that Ag_nomem agents are the slowest to learn an artifact capability regardless of the *PS*. Interestingly Ag_social agents performed significantly better than the others. Even after observing 25% of the attributes, their speed to convergence was significantly better. This supports the notion that artifact capability learning should occur faster with a social species than one that learns on its own.

Table 4.3 depicts a significant reduction in average convergence times for Ag_social agents as more attributes were made visible to the agents. For the *fixed value PS*, average convergence times were cut in half when the visible attributes were doubled. For *range-of-values PS* average convergence times increased fivefold for 4 attributes with 1 visible compared to when 2 were visible. In the case of 8 attributes, the increase was almost 7 times between 2 visible attributes and 4 visible attributes.

In general there was little difference found in the results of the experiments conducted between agents Ag_nomem and Ag_mem. The techniques utilized by both types of agents differ only in that Ag_mem remembers attempts that have failed and does not repeat them. It makes sense that Ag_mem never does worse than Ag_nomem, and would only do better when there is a higher tendency of repeating the choice of attribute values. Although it would seem that this would be more likely to occur when there are fewer attributes, it would not necessarily be the case, as fewer attributes could also lead to faster convergence. One situation where Ag_mem performed notably better than Ag_nomem was with the *fixed value PS*. This can be observed in the *Fixed* column of Table 4.1, where Ag_mem converges much faster than Ag_nomem. This can be explained by the notion that agents are more likely to repeat their selection of attribute values, in their quest to find values that match a particular value, than when they are searching for values that fall in some range.

Results also seem to suggest that changing just a single attribute at each step and taking into account some idea of the progress towards learning a tool capability, would lead to faster convergence than simply changing some or all of the attributes. This is observed in Tables 4.1 and 4.2, where AG_GA agents outperform AG_MEM agents, regardless of the *PS*.

We also observed that when agent selections were being tested against the *fixed value PS*, agents had a very difficult time learning to use the artifact. Even when there were few attributes, agents still took much longer to achieve success, than when there were more attributes being tested against the *range-of-values PS*. Again, this is in accordance with our expectations, as one would expect to find it more difficult to figure out an exact way of doing something if there was only one way, compared to when there are a variety of ways.

4.5 Conclusions

In this chapter we have provided a model for learning artifact use from individual experience and one type of social experience namely observational learning. Results demonstrate the superiority of learned use over random use and that rational agents can learn more efficiently through social experience than through individual experience. In the next chapter we build on the model to address an additional form of social experience learning, introducing into the model the notion of culture. This is keeping in line with the overall objective of MABS agents taking advantage of the social dimensions in MABS to evolve.

Chapter 5

A Cultural Evolutionary Model for Artifact Use

In this unit the model for learning artifact use through individual and observational learning presented in the previous chapter is extended to include a cultural component. This extension permits the model to address cultural evolution of artifact capabilities. Cultural evolution is realized by integrating the prior model into a CA where collaborative agents can learn from a shared belief space. In this way, integrated patterns of behavior accumulate changes across generations of the social population. The collaborative social learning strategy which is an additional strategy implemented by the *LE* component of the agent's cognition is compared with individual and observational learning. The previously implemented generic MABS is extended and used to conduct experiments. The work presented here has been previously published in Mokom and Kobti [62].

5.1 Social Learning via Collaboration

For the performance standard, only the *range-of-values PS* from Section 4.1 is supported in this version of the model along with its corresponding *CE* fitness function from Section 4.2 for the evaluation of use actions.

The distinction between the strategy here and the individual and observational learning strategies lies primarily within *LE* which directs how *PE* will act. *PE*'s actions for agents learning collaboratively are influenced by knowledge extracted from the population at large. As in the previous chapter use actions that agents learn are simplified to values in a predefined functional attribute sequence: $ua_j = \langle V, y \rangle$ where $V = \langle v_1, \ldots, v_n \rangle$ for n functional attribute values with y as the evaluated score. The agent learns to solve the artifact use learning problem defined in Section 3.2.3. The search space of the algorithm is a function of the number of functional attributes and the performance standard. When evaluating a use action against a fixed standard for example, it is necessary to compare each generated value with its associated standard value in order to obtain its fitness. Therefore the evaluation grows linearly with the number of attributes.

Henceforth we will refer to the *LE* and *PE* components of the different strategies as follows: Le1 and Pe1 for individual learning with a GA, Le2 and Pe2 for social learning by observation with a seeded GA, Le3 and Pe3 for the new strategy, social learning by collaboration with a CA.

5.1.1 Cultural Algorithm Framework

The CA for the collaborative learning agents consists of a population space of artifact capability learning agents (P) and a global belief space (GB) that maintains extracted knowledge from the population. Following the procedure of a CA, selected individuals

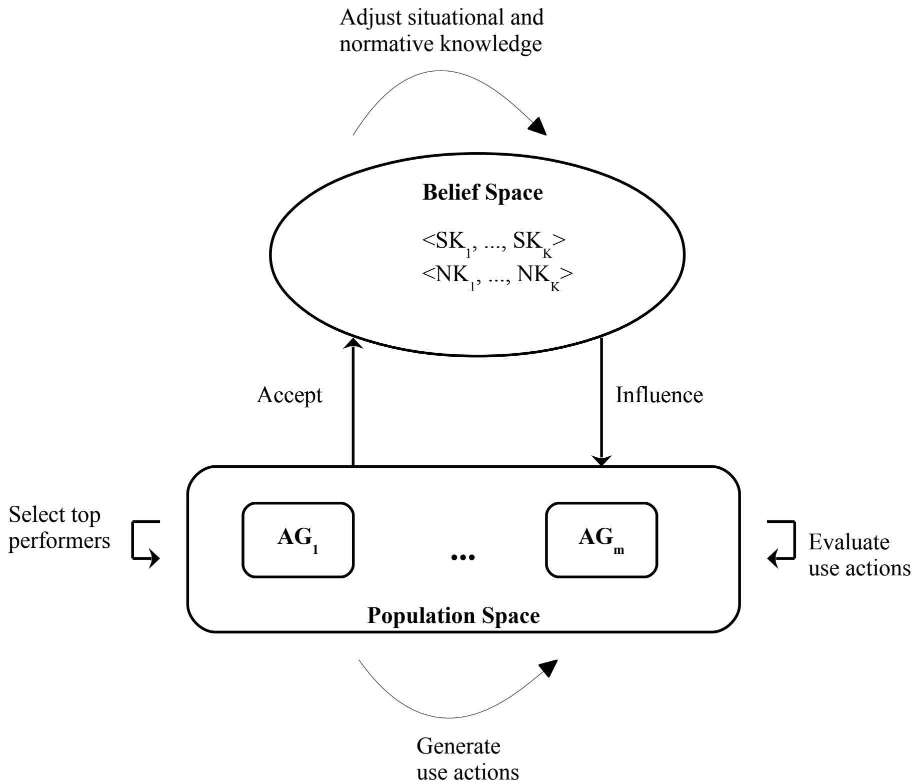

Figure 5.1.1: Cultural Algorithm for m social agents learning a k-action artifact capability

from P contribute their knowledge to GB which is used to influence the evolution of the knowledge of agents in P over time.

5.1.1.1 Belief Space Structures

Of the five knowledge sources offered for the CA's GB, the CA for collaborative learning by artifact capability learning agents supports situational knowledge (SK) and normative knowledge (NK) as depicted in Figure 5.1.1. The figure shows m agents collaboratively learning the required k use actions for an artifact capability .

SK maintains the best examples found in the evolving population so far. This constitutes the highest scoring use actions for each action learned so far. NK maintains encouraging ranges for each functional attribute in each use action learned so far. The combined influence makes it feasible for agents to follow the exemplar and strive to get into a desirable range [24]. GB should not be confused with the local belief that each PE3 maintains analogous to PE1 and PE2. While the local belief space maintains an agent's personal history of failed attempts, GB maintains knowledge extracted from the population at large.

GB is defined as: $GB = \langle SK, NK \rangle$, where $SK = \langle SK_1, \ldots, SK_k \rangle$ represents the situational knowledge and $NK = \langle NK_1, \ldots, NK_k \rangle$ represents the normative knowledge for k use actions of an artifact capability. SK_j maintains the single best exemplar found so far for use action j and is defined as:

$$SK_j = \langle XV, z \rangle \tag{5.1.1}$$

where XV denotes the exemplar use action's selected attribute values for n functional attributes : $XV = \langle xv_1, \ldots, xv_n \rangle$ and z represents the action's score. NK_j maintains favorable value ranges for each attribute in use action j and is defined as:

$$NK_j = \langle I_1, \ldots, I_n \rangle \tag{5.1.2}$$

Each I_i is a tuple specifying an interval or range and related scores for the i^{th} attribute:

$$I_i = \langle zl, zu, [l, u] \rangle \tag{5.1.3}$$

where l and u represent the favorable lower and upper bound values of attribute i, initialized with the boundaries of the attribute's domain. The other two elements zl and zu represent their respective scores given by the use action from which they were obtained.

5.1.1.2 Adjusting the Belief Space

GB is responsible for its own adjustment when knowledge is received from top performers in the population. The received use actions are sorted according to their scores. Let h represent the best example for use action j from the population: $h = \langle V_h, z_h \rangle$, then it is used to adjust the situational knowledge SK_j defined in Formula ((5.1.1)) as follows:

$$SK'_j = \begin{cases} h, & z_h > z \\ SK_j, & otherwise \end{cases} \tag{5.1.4}$$

Thus the current exemplar is only replaced when the proposed example has a better score.

The adjustment of NK is handled by dealing with one functional attribute at a time. For each attribute i, selected values by its top performers are obtained and sorted. The lowest selected value x_i and the highest selected value y_i, with their corresponding scores zx_i and zy_i can now easily be extracted. NK_j for a use action j as defined in Formulae ((5.1.2) and (5.1.3)) is updated for each attribute i using the following formulae:

$$l'_i = \begin{cases} x_i, & (x_i < l_i \; and \; zx_i = zl_i) \; or \; zx_i > zl_i \\ l_i, & otherwise \end{cases} \tag{5.1.5}$$

$$zl'_i = \begin{cases} zx_i, & (x_i < l_i \; and \; zx_i = zl_i) \; or \; zx_i > zl_i \\ zl_i, & otherwise \end{cases}$$

$$u'_i = \begin{cases} y_i, & (y_i > u_i \; and \; zy_i = zu_i) \; or \; zy_i > zu_i \\ u_i, & otherwise \end{cases}$$

$$zu'_i = \begin{cases} zy_i, & (y_i > u_i \; and \; zy_i = zu_i) \; or \; zy_i > zu_i \\ zu_i, & otherwise \end{cases}$$

Using these rules, the agents will progress towards learning the correct range required by the *PS*.

5.1.1.3 Belief Space Influence on the Population Space

The population space is implemented with a GA similar to LE1. The distinction stems from *GB*'S influence on the solutions in the GA pool. A bit representation is still utilized for candidate solutions ($\langle v_1, \ldots, v_n \rangle$ where v_i is a bit sequence equivalent to an integer value drawn from functional attribute i's domain). Influence from *GB* is applied when all solutions in the pool are evaluated and a new pool of solutions needs to be generated. Selection for reproduction is realized with roulette wheel selection. Two-point crossover is applied according to a given rate to swap the bits of a randomly chosen single attribute's values. With a GA integrated in a CA however, mutation is carried out differently although still according to a given rate. Instead of mutating the bits representing the chosen attribute value as is done by LE1 and LE2, *SK* and *NK* are used to determine direction and step size for LE3's mutation respectively. The direction determines whether the influence results in an addition or subtraction from the current value while the step size determines the value that is added or subtracted. Let q be a candidate solution for use action $j : q = \langle W, y \rangle$ where $W = \langle w_1, \ldots, w_n \rangle$, then the chosen attribute's value w_i is mutated using the following formula derived from Chung and Reynolds [24]:

$$w_i' = \begin{cases} w_i + \left| (u_i - l_i) \cdot N(0,1) \right|, & w_i < xv_i \\ w_i - \left| (u_i - l_i) \cdot N(0,1) \right|, & w_i > xv_i \\ w_i + (u_i - l_i) \cdot N(0,1), & otherwise \end{cases} \tag{5.1.6}$$

where xv_i represents the exemplar value in SK_j as defined in Formula (5.1.1), l_i and u_i correspond to the lower and upper bounds for attribute i in NK_j defined in Formulae ((5.1.2) and (5.1.3)), and $N(0,1)$ is a random value obtained using the standard normal distribution. Since the GA evolves a pool of solutions, it should be noted that the agent is also learning individually. Hence individually learning is integrated with the collaborative learning strategy employed. It is assumed that all agents are equally susceptible to influence from cultural beliefs.

5.1.2 Employing the Cultural Algorithm

Every agent learning by collaboration uses the CA to learn use actions. Given a pool size, each LE3 generates a random pool of use actions in bits then converts one in the pool to its equivalent integer values and offers it to PE3. LE3 uses the feedback obtained from the agent's *CE* to assign a fitness score and continues to offer use actions to PE3 until all actions are evaluated or a successful action is discovered. Once the pool is exhausted LE3 selects a designated number of top performers, offers them to *GB* and clears the pool. *GB* undergoes any necessary adjustments. A new pool of candidate solutions is then generated by LE3 with influence from *GB* and one is offered to PE3. If a successful action is discovered LE3 offers it for acceptance into *GB*, and randomly generates a new pool to learn a new action if the goal is not yet achieved.

The pseudo code for learning use actions with the CA is shown in Algorithm 3.

Algorithm 3 Pseudo-code for learning use actions with Cultural Algorithm

```
Begin

    if size(p) < P_SIZE

        initialize p with random actions
        pidx = 0

    else

        if pidx = size(p)

            Select top_performers from p
            Accept selected performers in GB
            Generate p' with influence from GB
            p = p'
            pidx = 0

    end
    action = get_action(p,pidx)
    pidx = pidx + 1

End
```

The agent's GA pool is represented by p, the pool index for traversing the pool is *pidx* and *get_ action* returns the action at the specified index from the pool.

5.2 Generic MABS Model

The MABS model in the previous chapter is extended to support the collaborative learning strategy. The MABS model here is defined as: $S \triangleq \langle AG, GB, R \rangle$ where AG is the population of agents, GB is the global belief space and R is the set of artifacts in the environment.

5.2.1 Model Parameters

Aside from the additions and exceptions noted in the parameter section below, the same parameters are used for the model. Hence fixed parameters NumberOfArtifactParts, ArtifactFD, AgentGoal, GACrossoverRate, GAMutationRate (for Ag_ga_pe1, Ag_social_pe2), NumberOfUseActions, FunctionalAttributeDomain and variable parameter NumberOfFunctionalAttributes are defined the same way. Experiments will once more differ according to the variable parameters. All agents still learn with one type of the same artifact and artifact types may differ between experiments.
The following are fixed parameters:

NumberOfAgents The number of learning agents in the model that constitute the set AG. There are three types of agents differing according to their employed learning strategy, if any. Henceforth they will be referred to as Ag_ga_pe1, Ag_social_pe2 and Ag_social_pe3. Ag_ga_pe1 and Ag_social_pe2 agents employ the individual learning and observational learning strategies from the previous chapter respectively. Ag_social_pe3 employ the collaborative learning strategy. There are 100 members of each type of agent resulting in a total of 300 agents in the model.

NumberOfArtifacts The number of artifacts in the model constituting the set R. Each agent is given an artifact, therefore there are 300 artifacts.

GAMutationRate The mutation rate for the GAs. It is set to 0.01 for Ag_ga_pe1, Ag_social_pe2 after experimenting with various rates. For Ag_social_pe3 it is set to $1/n$ for n functional attributes, so that per mutation, influence from the cultural space affects one attribute.

GAPopulationSize The number of use actions in all GA pools fixed at 100.

NumberOfTopPerformers The number of use actions that are offered by each agent for acceptance into *GB* fixed at 5% of the GA pool, that is, 5.

PerformanceStandard The only *PS* supported which is the *range-of-values PS* defined to cover 20% of the functional attribute's domain with all domains fixed at [1..100] as in the previous chapter.

NumberOfVisibleAttributes The number of visible functional attributes for the artifact set at 25% of the artifact's total functional attributes (which vary).

5.2.2 Simulation Flow

The simulated environment is a simple 20 x 15 toroidal grid world, in which each of the 300 squares contains an agent and the same type of artifact. For the most part, the simulation flows in the same manner as depicted in Algorithm 2. The only distinction is that *GB* is initialized at the beginning of the simulation so that AG_SOCIAL_PE3 agents can evolve using the collaborative learning strategy. All agents learn concurrently using their designated learning strategy. It should also be noted that once an agent succeeds it offers its solution to *GB.* This is important because solutions are normally only offered to *GB* when an agent has evaluated all the actions in its pool. This exception allows an agent to offer a good solution that it finds while it is still traversing the elements of its pool.

5.2.3 Experiments and Results

All agents use the same type of artifact. The same random seeds are used to initialize the random number generator for each agent type to ensure that the agent types begin the evolution process equally. This means that at the start of the simulation the randomly generated values of the initial population of solutions by one AG_SOCIAL_PE1 should be equivalent to one AG_SOCIAL_PE2 as well as one AG_SOCIAL_PE3. Test

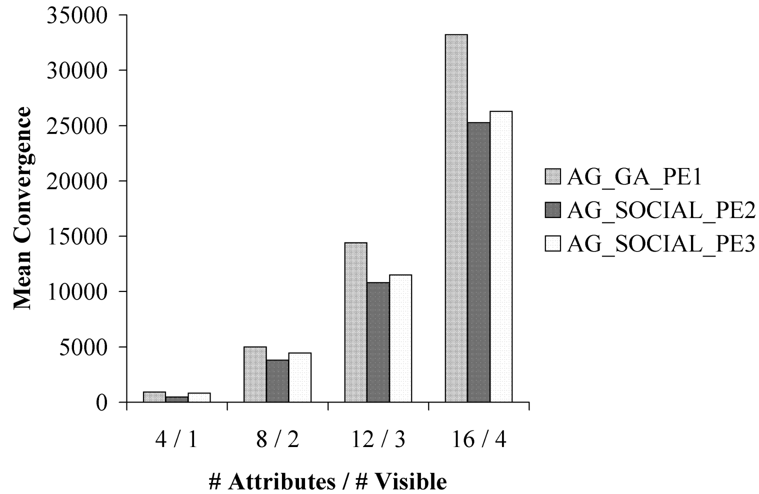

Figure 5.2.1: Mean convergence for all agents learning capability for 4, 8, 12 and 16-attribute artifacts (Attribute visibility only applies to AG_SOCIAL_PE2 agents

cases vary in the number of functional attributes and the corresponding number of visible functional attributes. Tests are conducted for an artifact with a single part and 4, 8, 12, 16, 20 and 24 attributes. At the end of each test run, the mean convergence times that is, the average number of simulation steps required to reach the goal for each agent type is computed. Results are depicted in Figures 5.2.1 and 5.2.2.

Figure 5.2.1 shows the mean convergence for all agent types learning capabilities for a single part artifact with 4, 8, 12 and 16 functional attributes with 1, 2, 3 and 4 visible attributes respectively for AG_SOCIAL_PE2 agents. Fig 5.2.2 shows the mean convergence social learning agents for a single part artifact with 8, 12, 16, 20 and 24 functional attributes with respective visible attributes 2, 3, 4, 5 and 6 for AG_SOCIAL_PE2 agents.

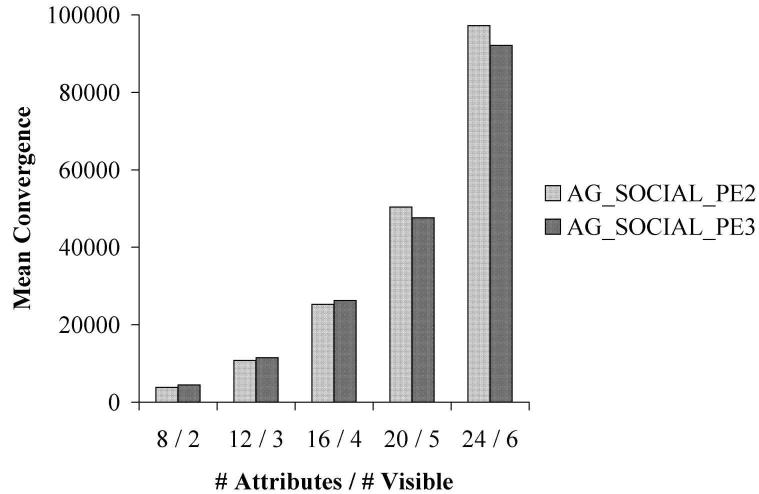

Figure 5.2.2: Mean convergence for social agents learning capability for 8, 12, 16, 20 and 24-attribute artifacts (Attribute visibility only applies to AG_SOCIAL_PE2 agents

5.2.4 Discussion

It can be observed in Figure 5.2.1 that AG_GA_PE1 agents were outperformed by both types of social agents AG_SOCIAL_PE2 and AG_SOCIAL_PE3 in all conducted experiments. As the number of attributes increased from 4 to 16 a difference in the convergence rates between individual and social learners is apparent, with individual learners needing more time to learn the capability. An interesting observation in Figure 5.2.2 is the difference in convergence rate between the two types of social learning agents. AG_SOCIAL_PE2 learn faster than AG_SOCIAL_PE3 agents for 8, 12 and 16 attributes. However at 20 attributes the collaborative learning agents outperform those learning by observation. The trend continues at 24 attributes as AG_SOCIAL_PE3 agents learn even faster.

The superiority of social learning by observation over individual learning was previously demonstrated so it is no surprise that AG_SOCIAL_PE2 agents do better than AG_GA_PE1 agents. The fact that AG_SOCIAL_PE3 agents also outperform AG_GA_PE1 agents supports the contention that learning through cultural evolution (with a CA) should proceed at a faster rate than through biological evolution (with a GA) [97]. It is understandable that AG_SOCIAL_PE2 agents perform better than AG_SOCIAL_PE3 agents when learning to use simpler artifacts or artifacts with fewer attributes since these agents begin the learning process with partial knowledge. As such AG_SOCIAL_PE2 agents have a head start in the learning process, whereas AG_SOCIAL_PE3 agents begin with no knowledge relying on the successes of their social group to improve over time. Knowledge compiled in the global belief space over time should guide the process of learning so that it improves with each successive trial [97]. As artifacts gain complexity in terms of the number of functional attribute values resulting in a much larger search space collaborators get better and eventually outperform those that began with prior knowledge obtained from observations. Although the observed threshold may vary or be problem dependent the results can be corroborated by other studies that demonstrate the use of CAs for optimizing complex applications [24]. In particular when the number of visible attributes of an artifact is low for observational learners, we suggest that the likelihood that learning by collaboration would be a better option increases.

5.3 Conclusions

In this chapter we have extended the learning capacities of the artifact capability learning agents from the previous chapter to include learning by collaboration through the integration of a GA into a CA. The cultural evolutionary model which included a population space of agents and a global belief space that maintained situational and normative knowledge was implemented in a generic MABS. The MABS included the prior two forms of learning: individual learning and social learning by observation.

The results confirmed that artifact capabilities are learned faster by social species than those operating in collaboration. The results also suggested a relationship between the complexity of the artifact (in terms of the number of functional attribute values) and the success of the type of social learning method employed. Collaborating agents seemed to outperform observational learning as artifacts gained complexity. Observational learners require the presence of another agent that has successfully performed the artifact capability within their vicinity from whom they can copy visible attributes, however agents learning through cultural evolution can commence learning without any knowledge at all.

Chapter 6

Adaptation Strategies for Artifact Use

In this chapter we direct our focus to social agents realizing their goals by exploiting artifacts in unpredictable environments. In Chapters 4 and 5 agents in a population with no particular social structure learned to exploit static artifacts in static environments. Here we address agents belonging to social networks learning to use unpredictable artifacts in dynamic environments. These extensions require agents to employ adaptation strategies. Unpredictable artifacts may be dynamic or heterogeneous in nature. An artifact is dynamic when the same action performed on it at different times of the evolutionary process produces different outcomes. Heterogeneity refers to an artifact that will produce different outcomes for the same action at the same time of the evolutionary process, when the action is performed by two different agents. The social population embedded in the CA in Chapter 5 is extended to support static and dynamic social networks. Static networks are constructed at the start of the MABS simulation and remain fixed throughout. Dynamic networks on the other hand change during the simulation. The agent's learning strategies are augmented to support real-valued functional attributes and additional strategies.

These include two additional types of learning from the cultural belief space, learning through social networks including evolving the members of the network and learning the extent of the network to generate at any given time. Agents can also learn using combinations of strategies and a meta-learning strategy that permits the evolution of the learning strategies themselves. Agents learn to adapt artifact use in environments where agents enter, leave and move around.

The integration of artifact use evolution into an existing MABS is demonstrated by incorporating the model into the MABS of the Village EcoDynamics Project developed to study the early Pueblo Indian settlers from A.D. 600 to 1300. In the Village MABS agents characterized as households use the paleoproductivity of the landscape to direct their decision of where to settle and farm. Eliminating the current presumption that this is known to the agents, the landscape is abstracted as an artifact and agents given the objective of farming for survival, are extended to employ artifact use learning strategies for its exploitation. The dynamic and heterogeneous nature of the landscape, the mobility of its inhabitants as well as agents entering and leaving the environment through marriages (new households) and deaths respectively provides a good test bed for the adaptability aspects of the artifact use model. Most aspects of the work presented here will appear in Mokom and Kobti [69].

6.1 Learning and Adaptation Strategies

Learning strategies are implemented by the LE component of the agent. The artifact use learning problem is defined in Section 3.2.3. Use actions that agents learn are simplified to values in a predefined functional attribute sequence as in the previous models however use actions have an additional element for the network radius: $ua_j \triangleq \langle V, r, y \rangle$ where $V = \{v_1, \ldots, v_n\}$ specifies a selected value for each of the artifact's n functional attributes, $r \in \mathbb{N}$ denotes the social network radius for social learning agents that evolve the extent of their dynamic social network and $y \in \mathbb{R}$ indicates a score attributed to the use action once its applied and evaluated. The model will also support real values for V as opposed to the previous restriction to integers and given the dynamic aspects being addressed agents will not maintain any record of failed actions in their local beliefs. Observational learning with visible attributes is also not supported. There are five distinct categories of strategies that agents are designed to employ in the model: individual learning, learning through social communication, learning through cultural belief space, combining various strategies and evolving strategies. With respect to the previous artifact use models provided strategies are either new or augmentations of previous ones.

In order to add more realism to the model, an additional rate is defined for agents that are influenced by others either through social networks or cultural beliefs:

- *Susceptibility rate:* the probability that an agent is susceptible to influence. For instance an agent with a *susceptibility rate* of 0.6 means there is a 60% chance that the agent will be influenced.

The rate increases the heterogeneity of the agents with respect to influence, as agents may now resist the adoption of knowledge from an influential source.

6.1.1 Individual Strategy

The individual learning strategy which facilitates agents learning artifact use through observations of their own behavior is implemented with a GA that uses a real-valued representation instead of a binary representation for the pool solutions. A candidate solution is a use action's values $V = \langle v_1, \ldots, v_n \rangle$ where each v_i is the real value of the functional attribute. The GA generates a random pool of solutions at the start and assigns a fitness score to each one after evaluation feedback. In order to regenerate the pool of solutions after evaluating all its elements, roulette wheel selection is used to select two candidates for reproduction. Crossover is applied at a given rate swapping a single randomly chosen attribute's value. Along with a specified mutation rate, real-valued mutation step-sizes are determined using the formula offered by the Breeder Genetic Algorithm (BGA) [72, 104]. BGA proposes to generate small step sizes with a high probability and large step sizes with a low probability. A functional attribute value v_i is mutated as follows: $v_i^{'} = v_i \pm r \cdot I_i \cdot \delta$ where $\pm$ is chosen uniformly at random, r is referred to as the mutation range with a standard value of 0.1 and I_i is the search interval or domain of the functional attribute with value v_i. δ is defined as: $\delta = 2^{-u \cdot k}$ where $u \in \{0, 1\}$ is chosen uniformly at random and k is referred to as the mutation precision, usually elements of the set $\{4, 5, \ldots, 20\}$ with 16 commonly used. We have used the common values 0.1 and 16 for the mutation range and mutation precision respectively.

In order to address adapting knowledge for unpredictable artifacts, the fitness score obtained after evaluating a performed action is used to update all pool solutions with values equivalent to the action. As a result identical pool elements that have been evaluated always have the same score, that is, the most recent one.

6.1.2 Social Network Strategy

The social learning strategies implemented here utilize a social network for agent communication towards learning. The social network may be an existing one or one that is constructed solely for learning artifact use. Social networks can remain fixed throughout the simulation or they can be dynamic in nature where agents continuously update them with different members. Social learning through networks is implemented with a GA similar to the individual learning strategy with a few distinctions. First the solution is extended with one more value representing the radius of the social network utilized by the agents that dynamically construct networks. This value can remain fixed for agents or given a predefined domain within which agents can randomly generate or evolve values with each action. The GA only requires a single-solution pool evolved with the influence from members of the agent's network when only social learning is employed, however if combined with individual learning the GA may maintain multiple solutions. For influence to occur, the agent searches its network for any performer whose current result for the use action is better. Once identified, the influence formula to influence solution $W = \langle w_1, \ldots w_{n+1} \rangle$ with a better performer's solution $X = \langle x_1, \ldots x_{n+1} \rangle$ is derived from Chung and Reynolds [24] characterization of influence from an exemplar:

$$w_i' = \begin{cases} w_i + \left| (x_i - w_i) \cdot N(0,1) \right|, & w_i < x_i \\ w_i - \left| (x_i - w_i) \cdot N(0,1) \right|, & w_i > x_i \\ w_i, & otherwise \end{cases} \tag{6.1.1}$$

where n represents the number of functional attribute values with the last value $n+1$ representing the radius. $N(0,1)$ is a random value obtained using the standard normal distribution. Attribute values are mutated according to a specified mutation rate and whether an agent is influenced at all depends on its susceptibility rate.

Agents that enter the environment during the evolutionary process do not commence learning with randomly generated actions. In the model, these agents use the latest evaluated use action of their nearest neighbor as an influence to initiate the learning process.

6.1.3 Cultural Algorithm Strategy

The CA utilized to implement learning from cultural beliefs makes some changes to the previously implemented CA in Chapter 5.

6.1.3.1 Belief Space Structure

Unlike the CA in Chapter 5 agents do not directly offer use actions to GB. Instead at specified intervals, the population is searched for a percentage of top performers and their use actions are offered to GB for acceptance. Although GB is still defined to maintain situational knowledge (SK) and normative knowledge (NK) it is augmented to support knowledge representing the social network radius and the maintenance of multiple exemplars in the belief space. The radius is added to support agents that combine learning from GB with social network learning (concurrently with radius). The set of k exemplars for use action j is defined as: $SK_j = sk_{j,1}, \ldots, sk_{j,k}$ with the i^{th} exemplar $sk_{j,i} \in SK_j$ defined as:

$$sk_{j,i} = \langle XV, r, z \rangle \tag{6.1.2}$$

where XV denotes the exemplar use action's selected attribute values for n functional attributes : $XV = \langle xv_1, \ldots, xv_n \rangle$, r which is assumed to be restricted by a domain in $\mathbb{N}$ denotes the value of the social network radius and z represents the action's score. NK_j which maintains favorable value ranges for each attribute in use action j is extended to include favorable ranges for the radius and is defined as:

$$NK_j = \langle I_1, \ldots, I_n, I_r \rangle \tag{6.1.3}$$

where each I_i with $1 \leq i \leq n$ is a tuple specifying an interval or range and related scores for the i^{th} attribute and in the case of I_r denoting the information for the radius. Each I_i and I_r is defined as in Formula (5.1.3).

6.1.3.2 Belief Space Adjustment

With respect to GB's adjustment when it accepts knowledge from top performers the fact that multiple exemplars are maintained in SK must be taken into account. The adjustment will also depend on the artifact's predictability defined by the function description which specifies the service the agent is trying to learn. When the artifact is predictable *GB* maintains the *best so far* and to facilitate adaptation for unpredictable artifacts *GB* maintains the *current best.* Since a predictable artifact always yields the same outcome for an action, agents can benefit from better examples even if they occurred in the past. However, when an artifact's outcome changes during the evolutionary process relying on a good result that is no longer useful would prove detrimental to the learning agents.

For predictable artifacts, when top performers are received in *GB* they are sorted according to their scores. Given use action j, let $h = \langle XV_h, r_h, z_h \rangle$ represent a contribution from performer h, and $x = \langle XV_x, r_x, z_x \rangle$ represent the worst performer (the exemplar with the lowest z score) in SK_j. SK_j is adjusted as follows:

$$SK'_j = \begin{cases} (SK_j \cup \{h\}) - \{x\}, & z_h > z_x \\ SK_j, & otherwise \end{cases} \tag{6.1.4}$$

Basically, every contributed performer replaces the worst exemplar if it has a better score. NK_j is adjusted with the same formula in (5.1.5).

For unpredictable artifacts, the adjustment is less complicated. Basically *SK* and *NK* are cleared. The top performers replace the exemplars in the belief space and Formula (5.1.5) is used to construct a new NK.

6.1.3.3 Belief Space Influence on the Population Space

The GA used to implement the population space can support a multiple-solution pool as before or a single-solution pool for each agent. An agent with a multiple-solution pool is actually implementing individual-learning concurrently with cultural influence. An agent with a single solution pool learns solely as a result of influence from *GB*. Previously *SK* was combined with *NK* to influence the population, so agents followed the exemplar and concurrently tried to jump into the normative range. Two additional types of belief space influence are added here. Agents can be influenced only by *SK* or only by *NK*. For agents influenced by *SK* only, an exemplar is randomly chosen from the set of exemplars in *GB*. The influence formula to influence solution $W = \langle w_1, \ldots w_{n+1} \rangle$ with the chosen exemplar's solution $X = \langle x_1, \ldots x_{n+1} \rangle$ is identical to Formula (6.1.1) where the *GB* exemplar is equivalent to the better network performer. For agent's influenced by *NK* only, the following formula derived from Chung and Reynolds [24] is used to influence W:

$$w_i' = w_i + (u_i - l_i) \cdot N(0, 1) \tag{6.1.5}$$

where u_i and l_i denote the upper and lower bounds for attribute i or the radius in *NK*. For agents influenced by the combined *SK* and *NK, W* is influenced using Formula (5.1.6) with a randomly chosen exemplar. As in the case of learning through social networks, attribute values are mutated according to a specified mutation rate and whether an agent is influenced by knowledge from *GB* depends on the agent's susceptibility rate.

6.1.4 Combining Strategies

Agents can decide upon any combination of learning strategies to employ. For instance an agent that wishes to learn on its own as well as socially or culturally would maintain a pool of solutions rather than a single one, using crossover as specified in individual learning and mutation on each solution with influence from better performers in its network or the cultural belief space respectively. An agent can also combine the social and cultural strategies choosing any of the three belief space influence types and randomly alternating its influence between better performers in its network and knowledge from the belief space.

6.1.5 Evolving Strategies

While agents can learn with any designated learning strategy the model supports agents that wish to evolve learning strategies as part of the learning process. At a minimum agents that learn which strategy to employ should outperform those that employ strategies at random. Two meta-learning strategies are supported for evolving the strategies: individual strategy evolution and social strategy evolution.

Individual strategy evolution is realized with a GA that uses a binary representation for the candidate solutions. A binary string of 5 bits is used to represent the strategies to be evolved: $[b_1b_2b_3b_4b_5]$. The first bit b_1 is set to '1' when individual learning is on and '0' when it is off. The next two bits $[b_2b_3]$ represent social learning with influence from cultural beliefs: '00' - no learning from cultural space, '01' - influence from *SK* only, '10' - influence from *NK* only and '11' - combined influence from *SK* and *NK*. The last two bits $[b_4b_5]$ represent social learning through social networks: '00' - no learning through networks, '01' - invalid, '10' - learning through social networks with randomly generated radius and '11' - learning through social networks concurrently learning the radius. The bit string where no learning occurs [00000] is considered invalid along with any bit string where $[b_4b_5] = [01]$ resulting in 23 possible learning strategies. For instance the bit string [11011] represents the combined strategies: individual learning, learning through *NK* influence and learning through social/social network radius while the bit string [01100] represents learning through the combined *SK* and *NK* influence. With a given pool size, roulette wheel selection is used for selecting solutions for reproduction. Crossover occurs at a given rate with two point crossover applied to swap a single type of influence, that is, b_1,$[b_2b_3]$ or $[b_4, b_5]$. Mutation occurs at a given rate. Invalid solutions are rejected and not considered for the next population of solutions. It may be useful in future work to attempt repairing these solutions in the event that there are too many invalid solutions that end up skewing the algorithm towards a random search.

Social strategy evolution is realized with a GA that extends the GA for individual strategy evolution with an integer value to represent the radius and reduces the pool to a single solution. The bits representing the strategies along with the radius are altered with influence from the agent's evolving social network members that have a better performance. Whether an agent is influenced depends on its susceptibility rate. As for the mutation rate, it is defined by a rate that controls influence on the bit strings:

- *Learning rate*: the probability that an agent copies the bit from the influential source correctly. For instance an agent with a *learning rate* of 0.8 means that there is an 80% chance that the bit will copied correctly.

This individualizes the influence from an influential source instead of all agents sharing a fixed mutation rate. Whether an agent is influenced at all depends on its susceptibility rate. The influence on the radius occurs in the same manner as when a social network member influences an agent's radius in the social network learning strategy.

Agents that evolve strategies are equipped with the GA for evolving strategies and a separate evolutionary algorithm (GA or CA) for each possible strategy. When evolving strategies the agent first determines a strategy to use. The selected strategy is then matched to its evolutionary algorithm which is used to learn the use action.

6.2 Case Study: Artifacts in the Village Multi-Agent Simulation

In this section a case study implementation of the model demonstrating its integration into an existing MABS system is presented. The Village EcoDynamics Project (VEP) [56, 57] is a significant part of a broader study of the history of the American

Southwest that has been well funded by the National Science Foundation (NSF) for over ten years. The project which involves researchers from several disciplines including Anthropology, Geology, Economics and Computer Science was developed to study the early Pueblo Indian settlers from A.D. 600 to 1300. A major component of the project, the Village MABS henceforth referred to as VillageSim models households constituting families as agents, as they farm for maize, hunt for protein, gather water and wood and employ various exchange models for trade [51, 52, 53, 58]. Births occur, marriages result in the formation of new households and death is the result of natural causes or agent's failure in meeting their needs. Many other aspects of the region are modeled including soil productivity, rainfall, forest density and animal density. Each household maintains plans adaptable to changes in the environment for obtaining resources and trading food with other households. Objectives of the project include understanding what led to the depopulation that occurred at the end of period, settlement distributions, violence and demography.

Although the region has many different ruins and real artifacts the objective of our research is to demonstrate the exploitation of any object that can be abstracted as an artifact using our artifact representation towards achieving the agent's goals. As such we strive to augment the agent's plans with adaptable plans involving objects that can provide essential functionality.

6.2.1 The Landscape Artifact

We focus on the farming task carried out by the agents in VillageSim. In VillageSim the landscape is divided into cells and agents are presumed to know the soil productivity of every cell throughout the years. As such agents automatically choose the more productive areas to settle and farm upon. A time step in the simulation is a year characterized by four seasons: spring, summer, fall and winter. Agents consume maize during all seasons however they plant in the spring and harvest in the fall.

Agents self-evaluate and will move when necessary or plant additional plots either in their settled cell or other productive cells. Many factors are utilized to measure soil productivity which changes over time and declines depending on how long and how often it has been cultivated.

Artifact exploitation is incorporated into VillageSim by abstracting the landscape as an artifact (Landscape) and eliminating the presumption that agents know how to best exploit it. Instead agents are stripped of all tasks except farming and expected to learn and adapt using Landscape over time in order to survive. Five features are selected to describe Landscape representing its functional attributes: the average elevation (*dem*), the average slope (*slope*), the average direction of slope (*aspect*), the average depth to bedrock (*depth*) and the average proportion of its biomass consisting of any subspecies of big sagebrush prior to any agricultural clearing (*artr*). Predefined domains are given by the VEP archaeologists for each attribute. Landscape is an unpredictable artifact. Choosing the same values for its five features is likely to produce a different outcome in different years. Moreover, Landscape is heterogeneous. Since agents occupy different cells, two agents in different locations choosing the same attribute values at the same time could also possibly experience different outcomes.

Agents in Villagesim are extended to employ the learning strategies towards exploiting Landscape. Learning algorithms are applied when the agent decides to move or plant additional plots in other cells. The agent learns to select productive cells for farming. Over time it evolves and adapts its plans as necessary to changes in settlement distribution, landscape productivity, demographics of households and the population at large, its social networks and emergent cultural beliefs. Along the way it maintains its primary objective which is to produce enough to feed its family for survival as agents that do not produce enough maize will ultimately die.

6.2.2 Generating and Evaluating Use Actions

Unlike the generic MABS implementations, it is not necessary to provide fitness functions for the agent's *CE* element nor aspects of *PS* as agent's already self evaluate in VILLAGESIM once the results of their actions are perceived. The results of the agent's performed action is characterized by the harvest obtained once a year. There is only one action to adapt which constitutes a combination of values selected for each of the five features. Unlike the generic MABS implementations where chosen functional attribute values directly map to the action performed by the agents, agents exploiting LANDSCAPE need to convert their selected values to a single cell. Since selected values will not necessarily be identical to those in any particular cell, an interpretation layer is needed to convert the use action's values to the closest matching cell. It is important to note that this occurs within *PE*, prior to the action being performed or evaluated, that is, it should not be confused with the fitness of the agent's action which is received as feedback in the form of the agent's harvest. Basically, *PE* needs to choose a cell once *LE* has supplied it with a combination of attribute values. Choosing the closest matching cell constitutes what *PE* needs to do in order to be able to apply the action formed by the values from *LE*. There are other domains where the use action values from *LE* can be directly applied without any further interpretations.

To match selected values to a cell a simple distance measure is used averaging over all attribute values to select the cell that is closest to the generated values. For a given cell with functional attribute values $CV = \{cv_1, \ldots, cv_5\}$ and a generated use action with functional attribute values $V = \{v_1, \ldots, v_5\}$ the following function is applied:

$$
\begin{aligned}
dst\left(v_i, cv_i\right) &= \begin{cases} 1.0, & v_i = cv_i \\ \frac{1}{|cv_i - v_i|}, & otherwise \end{cases} \\
Dst(V, CV) &= avg\left(\sum_{i=1}^{5} dst\left(v_i, cv_i\right)\right)
\end{aligned}
\tag{6.2.1}
$$

If multiple cells have the same distance measure one is randomly selected.

6.2.3 Relevant Model Parameters

While fixed parameters remain the same for all experiments, variable parameters take on different values. It should be noted that it is possible to conduct other experiments with different values even for the fixed parameters and that VILLAGESIM has many more parameters, however only those that directly relate to the exploitation of LANDSCAPE are identified. Many of these parameters are constrained by the case study (as required by the VEP archaeologists).

The following are the fixed parameters:

NumberOfAgents The number of learning agents at the beginning of the simulation. This is fixed at 600.

NumberOfArtifacts The number of artifacts. There is a single artifact LANDSCAPE to which all agents have access.

NumberOfArtifactParts The number of artifact parts. LANDSCAPE is an artifact with a single part.

NumberOfFunctionalAttributes The total number of functional attributes. LANDSCAPE has five functional attributes.

FunctionalAttributeDomain The domain of the landscape features (functional attributes) provided by the VEP archaeologists:
artr[0.008421053,0.5198181], *aspect*[0.002658795,359.9978],
dem[1438.436,3008.686], *depth*[25.2,182.7], *slope*[0.0,49.36105].

ArtifactFD The function description describing the service the agent is to learn. LANDSCAPE has one *FD* with external goal: *farm*, an empty set of criteria objects and unpredictable set to 1.

AgentGoal The agent's goal. All agents share the same goal which matches the external goal of LANDSCAPE: *farm*.

GACrossoverRate The crossover rate for GAs with multiple solution pools (any strategy that includes individual learning) set to 0.7.

GAMutationRate The mutation rate for the GAs. A mutation rate of 1/5 is used for mutating attribute values. For agents evolving strategies on their own (without social influence) a mutation rate of 0.01 is used.

GAActionPopulationSize The number of use actions in all multiple solution GA pools fixed at 4 . This number is kept small as agents in VILLAGESIM will often die before all actions in the pool are evaluated.

GAStrategyPoolSize The number of strategies in GA pool when evolving strategies with individual strategy evolution. This is set to 10.

LearningRate The learning rate for each agent evolving strategies through social influence. It is randomly generated when the agent is created.

SusceptibilityRate The susceptibility rate for each agent evolving through any influential source. It is randomly generated when the agent is created.

NumberOfTopPerformers The number of top performers offered for acceptance into *GB* fixed at 5% of the current agent population. At the start this is 30 (given the 600 agents) but will change according to the number of surviving agents.

GBUpdateInterval The interval defining when top performers are contributed to *GB* fixed at 5 to indicate *GB* is updated every five years.

NumberOfUseActions The number of use actions the agent is learning and adapting specified as part of *PS*. This is fixed at 1.

SocialNetworkRadiusDomain The domain of the social network radius used by agents evolving the network radius. This is set to [1,40] which is the same radius currently used in VILLAGESIM for moving.

PerformanceStandard The *PS* used for evaluating the agent's actions. Only specifies a single action is required for the capability plan. VILLAGESIM handles the rest of the *PS* through its self evaluation.

The following are the variable parameters:

LearningStrategies Indicates which strategy or strategy combination is being employed (when the agent is not evolving strategies).

EvolvingStrategies Indicates whether the agent is evolving strategies.

MetaStrategy Indicates which meta strategy the agent is employing if any: individual strategy evolution or social strategy evolution.

6.2.4 Simulation Implementation

Given that VILLAGESIM has over 20,000 lines of code, the integration of the artifact use model into VILLAGESIM required very few modifications to its existing code. The following modifications were made to VILLAGESIM:

- Added variable to turn artifact use learning on and off.

- Added function to create LANDSCAPE as a single part artifact, with the five features as functional attributes and their predefined domains. Each agent was given access to LANDSCAPE.

- Added a variable to store a list of attributes denoting features for the *Cell* object, with each one represented as a <name,value> pair.

- Added code to load data for the cell features of all landscape cells.

- Added code to create agents as artifact use learning agents.

- Modified agents to use function in the artifact use learning model for scoring cells in order to choose cell to settle upon or to plant additional plots.

- Included the artifact model as a new package.

Modifications to the artifact use model to accommodate integration into VILLAGESIM:

- Defined the artifact use learning agent as a subclass to VILLAGESIM'S *Agent.*

- Added function to match the generate values of use actions to a cell, given a set of cells.

- Added function to update the score of a use action with obtained harvest.

The general flow of VILLAGESIM depicting artifact use learning is provided in Algorithm 4.

At the start cell data is loaded, the LANDSCAPE artifact is created and randomly generated agents are placed on the landscape. Agents are initialized by formulating and activating a goal for farming with LANDSCAPE, and initializing the capability. The simulation begins in year 600 A.D. and runs through 1280 A.D. There are many aspects to the simulation however we have focused on the activities that directly relate

Algorithm 4 Pseudo-code for agents exploiting LANDSCAPE in VILLAGESIM

```
Begin

    Load data for the five attributes of all cells
    Create cell objects setting its features with loaded data
    Create LANDSCAPE artifact
    Generate/Initialize random learning agents
    Randomly place agents on landscape
    Apply 1,2,3,4 to settle each agent in a cell
    year = 600
    repeat
        for each agent in each season
            if moving or planting in other cells

                1. C = selected cells within a defined radius
                2. C = PE→score_cells(C)
                3. c = best_cell(C)
                4. Settle/Farm on c
            end
            if season = fall

                LE→updateActionScore(CE→get_harvest())
            end
            Consume maize (if not enough for family, die)
        end
        Remove dead agents
        If moving, apply 1,2,3,4
        Create new households as per current simulation
        year = year + 1
    until year = 1280

End
```

to learning the exploitation of LANDSCAPE. Every year the agent goes through the four seasons, planting if necessary in the spring, harvesting in the fall and moving when necessary. Agents may move during planting season or at the end of the year according to their self evaluation. When the agent needs to move or plant in additional cells, the agent selects a set of cells within a predefined radius. It then decides on a cell by scoring all cells against a use action and choosing the best scoring cell. The pseudo code for scoring cells is shown in Algorithm 5. Cells are scored using the distance function Dst provided in Formula (6.2.1) to measure how close a cell's values are to the latest unevaluated use action provided by LE's algorithm. In the fall CE obtains the agent's harvest and passes it on to LE as feedback for the latest use action. LE updates the use action's score and maintains its learning algorithms.

Algorithm 5 Pseudo-code for agents scoring cells in VILLAGESIM against generated use actions

```
Begin

    if no current_action or current_action evaluated
        current_action = LE→get_action()
    end
    for each cell in C

        current_action_values = values(current_action)
        cell_score = Dst(current_action_values,cell_values)
        update cell_score for cell in C
    end

End
```

6.2.5 Experiments and Results

The village simulated in conducted experiments is VEP IIN which models a larger region than the original VEP I. Spatially, agents occupy a landscape represented as 114,240 cells (VEP I occupied 45,400 cells [57]). All conducted experiments track the survival of the agents. It should be noted that the model can still be used to investigate other aspects other than survival, such as settlement distribution etc. To facilitate the identification of no learning and the various learning strategies employed they are henceforth referred to as follows: No learning or randomly choosing strategies (*Random),* Individual learning (*Indv*), Social learning with randomly generated radius (*SocRRad*), social learning with learned radius *(SocLRad*), learning from the cultural belief space with situational knowledge (*CulS*), normative knowledge (*CulN*) and combined situational and normative knowledge (*CulB*). In the case of *Random* the agent randomly generates attribute values for a use action every time it needs to choose a cell for moving or planting or selects a random learning strategy if evolving strategies. Next, *EvStrategy-Indv* is used for meta strategy learning with individual strategy evolution and *EvStrategy-SocLRad* is used for social strategy evolution. Finally *Original* refers to the original simulation where the knowledge for soil productivity is presumed.

Results are aggregated over the four study periods obtained from the Pecos classification [55] currently used by the Village Ecodynamics Project researchers. They are *Basketmaker III (A.D. 600-750), Pueblo I (A.D. 750-900), Pueblo II (A.D. 900-1150)* and *Pueblo III (A.D. 1150-1280).* The objective is to provide results in a manner that enables archaeologists and anthropologists to analyze their findings. For every conducted experiment results show the number of agents that survived at the end of each classified phase.

In the first set of experiments agents learn with specified strategies. The first test case compares the three cultural belief strategies: *CulS, CulN* and *CulB*. Results are shown in Figure 6.2.1. The next test case compares social learning agents: *SocRRad* and *SocLRad*. Results are shown in Figure 6.2.2. Next we compare agents that are not learning with a strategy with the *Indv* agents and the most successful in the social learning strategies: *Random, Indv, CulN* and *SocLRad*. Results are shown in Figure 6.2.3. In the last of the first set of experiments, a comparison is done between combined strategies: *Indv + SocLRad, Indv + CulB, SocLRad + CulB* and *Indv + SocLRad + CulB*. Results are shown in Figure 6.2.4.

The next experiment investigates agents evolving strategies. For this there is a single test case where we compare the results of agents evolving with randomly chosen strategies to those with learned ones. Results are depicted in Figure 6.2.5.

In the final experiment, the original simulation is compared to *SocLRad*. The original simulation is run with agents stripped of all tasks except farming with agents knowing the best cells for settling and farming. Results are shown in Figure 6.2.6.

6.2.6 Discussion

The conducted experiments examine the survival rate of the population throughout the evolutionary process by tracking the number of agents that survive at the end of each of the classification phases. Although statistical testing such as significance tests were not formally conducted, many test runs were conducted (over 50) in order to ensure that the results remained consistent.

Figure 6.2.1 shows the results of our initial test case which compares *CulS, CulN* and *CulB.* The population barely survives by the end of *Pueblo III.* Agents learning through *GB* are influenced by the best performers chosen from the population at large. The struggle for survival may be explained by the heterogeneity of LANDSCAPE which makes use action values that are good for top performers not necessarily beneficial to most members of the population. As a result agents lose sight of strategies that would be successful at the local level and follow popular strategies that are detrimental. *CulN* is slightly more successful that the other strategies indicating that agents may be better off following good ranges than specific performers.

Comparing *SocLRad* with *SocRRad,* Figure 6.2.2 demonstrates that the size of the agent's network plays a significant role in its chances for survival. *SocRRad* agents significantly underperform when compared to *SocLRad* agents. *SocLRad* agents evolve the social network radius alongside attribute values. As a result they are able to learn how far to go to find partners to learn from. This notion is very important as success related to the size of the radius depends on the heterogeneity of the area the agent is in. While it should be beneficial for an agent in a homogeneous area to maximize the size of its network radius, this would most likely prove detrimental in heterogeneous areas. Both social network learning strategies are better than the *GB* strategies. This can be explained by the fact that compared to the *GB* strategies, the social network approach allows agents to learn what is best for them at a local level.

Figure 6.2.3 shows very little difference between *Random* and *Indv* agents, even though *Indv* agents are slightly better. Both are poor performers and *Random* agents are practically gone by the end of *Pueblo II.* It is likely that many *Indv* learning agents die before they get a chance to learn, even with a small pool of solutions. As expected though, a learning agent should never perform worse than one that is not learning and that is reflected in the results. Agents learning from *GB* influence are also shown to do better than *Indv* agents which also makes sense since some of the population should at least benefit from the top performers in *GB.* The results show the major difference between *SocLRad* agents and the others revealing *SocLRad* when employed on its own as the best adaptive strategy. A wider gap can be observed between *SocLRad* and the other strategies in *Pueblo II* and *Pueblo III* when compared to the gaps in *BasketMaker III* and *Pueblo I.* Apparently this is consistent with archaeological findings which identify *Pueblo II* and *Pueblo III* as periods when the landscape showed the highest variability from year to year. According to their analysis it is during these periods that wider gaps between social network, particularly those learning the radius and other learners should emerge. Comparing the results for the other network learners *SocRRad* to non-social network learners it should be noted that although the gap does not widen, *SocRRad* learners improve slightly through the high variable periods (the last two phases) while performance for the other learners drops.

Results with various learning strategy combinations in Figure 6.2.4 once again depict the importance of social network learning on the LANDSCAPE artifact. Although lower than *SocLRad* employed on its own the survival count remains significantly better for any combined strategy that includes it as opposed to *IndvCulN* which does not. Agents obtain better results when *SocLRad* is combined solely with *Indv,* than when the combination includes influence from *GB.* When agents combine *SocLRad* with *Indv* they are essentially evolving solutions possibly with influence from multiple better network members, resulting in the best combination. Adding *CulN* causes the agent to evolve multiple solutions with influence from *either GB* or the network, hence the performance reduces. When agents combine *SocLRad* and *CulN* a single solution is evolved with a randomly chosen exemplar *or* network member. As a result they are the worst performers amongst agents evolving combined strategies that include *SocLRad.* For agents that combine *Indv* and *CulN* the agent evolves a pool of solutions which are influenced by *GB.* Evidently the results are somewhat similar to agents employing *CulN* on its own, as it should make little difference which *GB* performers influence the agent when a heterogeneous artifact is involved.

Results depicted in Figure 6.2.5 demonstrate that agent performance can be positive without a learning strategy known *a priori.* The population barely survives when strategies are selected randomly. Unlike the *Random* agents in Figure 6.2.3 that employ no learning strategy at all, agents here are always learning although the employed learning strategy is chosen at random. As a result unlike their no learning counterparts, the population does survive through *Pueblo III.* Social learning through the network once again proves beneficial even at the meta-level as *EvStrategy-Social* agents outperform *EvStrategy-Indv* agents.

Figure 6.2.6 shows the results for *Original* agents that are presumed to know the productivity of the soil compared to agents employing the most successful adaptive strategy *SocLRad.* First the results show that the low performance of agents in *BasketMaker III* is not solely attributed to learning. Agents in *Original* know the landscape productivity however population growth and survival is much lower compared to the last three phases in the simulation. The more relevant aspect of the results is what happens in these phases. It is already expected that results in *Original* should always outperform those in *SocLRad*. However, while results in *Original* indicate that survival is for the most part unchanged between *Pueblo I, Pueblo II* and *Pueblo III* the results in *SocLRad* are apparently more consistent with archaeological findings that suggest adaptation to be relevant during these periods. Moreover it confirms once more the earlier contention that the wider gap between social network learning agents and other learning agents during the last two phases as depicted in Figure 6.2.3 can be attributed to the variability during these periods. If variability did not play a role in agent performance then given the results by *Original* that depict similar performances in the last three phases, the other agent learning types in Figure 6.2.3 should not perform worse in *Pueblo II* and *Pueblo III* compared to *Pueblo I*. However, not only do they perform worse but Figure 6.2.2 shows that the other social network learners *SocRRad* do not. The results therefore provide some insight into the role that the landscape variability may have on agent performance with various learning strategies employed.

6.3 Conclusions

In this chapter we have provided a more extensive artifact use learning model for adapting the exploitation of unpredictable artifacts in dynamic environments through various learning strategies. The model was integrated into the existing Village multi-agent based simulation which models the lives of the ancient Pueblo Indians spanning almost 700 years. The simulation provided a real complex social system suitable for conducting experiments with our artifact and capability concepts. With agents rep-

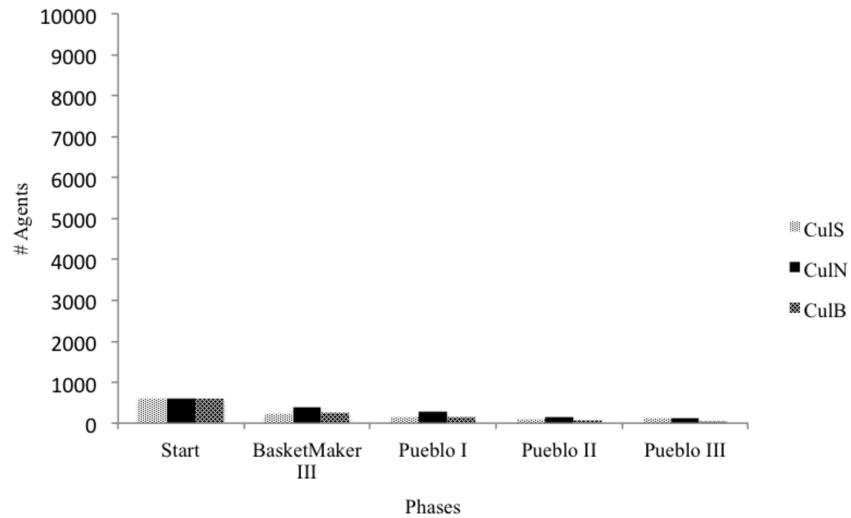

Figure 6.2.1: Agent survival for cultural-learning agents at the end of each classification phase

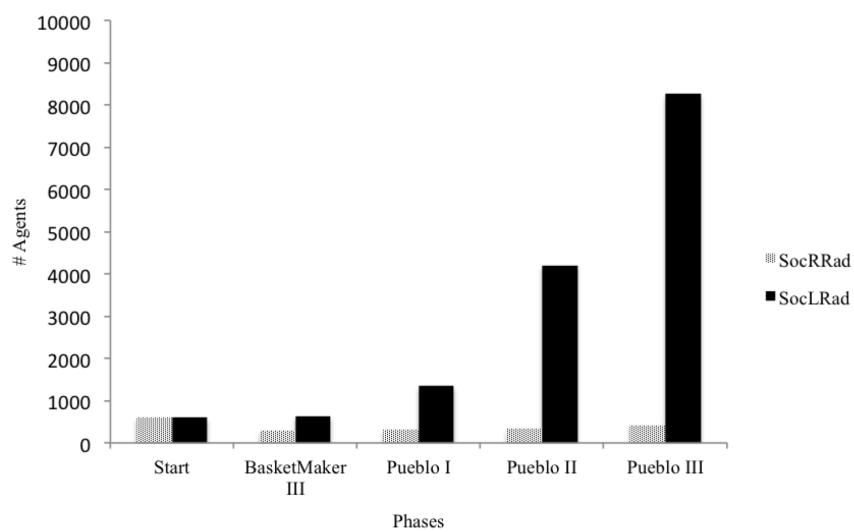

Figure 6.2.2: Agent survival for social-learning agents collected at the end of each classification phase

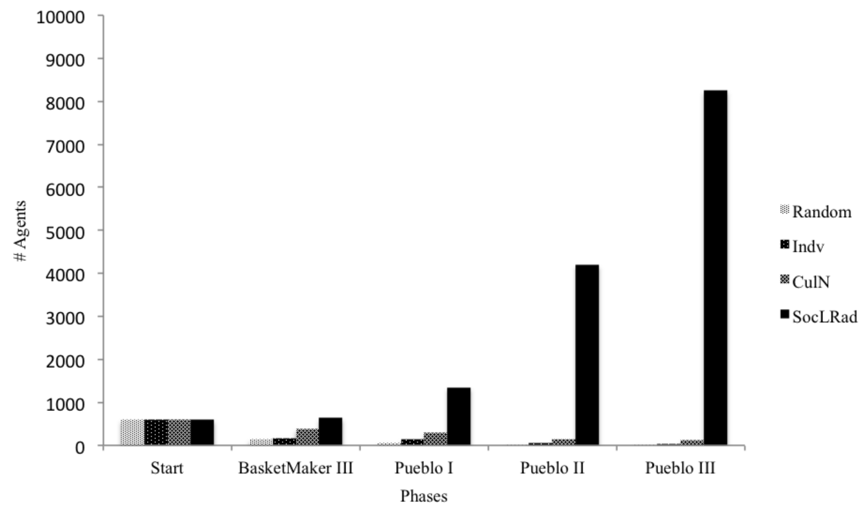

Figure 6.2.3: Agent survival for agents not learning compared with learning agents at the end of each classification phase

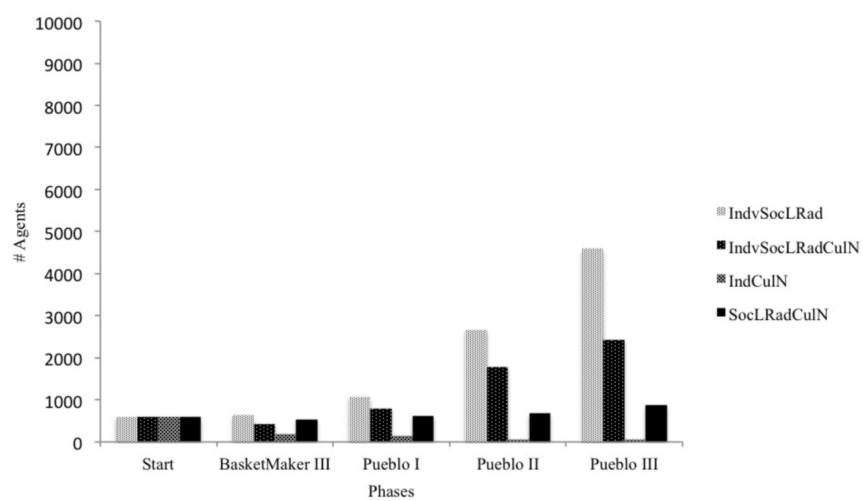

Figure 6.2.4: Agent survival for combined learning strategies at the end of each classification phase

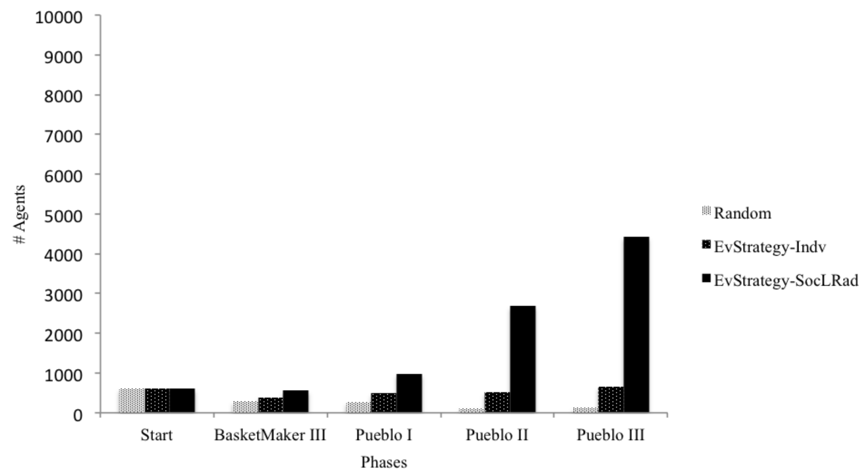

Figure 6.2.5: Agent survival for agents randomly choosing strategies compared with agents evolving strategies

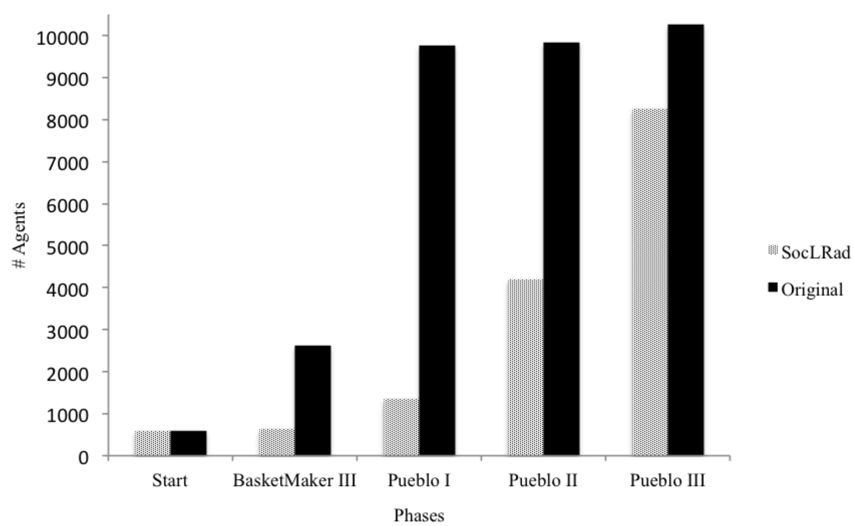

Figure 6.2.6: Agent survival for agents in the original simulation that know productivity compared with the best learning agent

resented as households farming for survival the landscape was modeled as an artifact abstraction that agents learned to exploit from a representation of a few of its attributes. Given its dynamic and heterogeneous nature and an environment characterized by agents entering, leaving and relocating as they strive to survive, learning and adaptation is essential for every agent. Experiments conducted track the survival of the agents aggregated over archaeological phases defined by the Pecos classification. Social learning through social networks while evolving the network radius is revealed as the best adaptive strategy. Results are considered consistent with archaeological findings that identify periods when the landscape showed high variability expected those to be the phases when social network learning with an evolved radius should prove most beneficial. Accordingly the widest gaps between that strategy and the others are observed during these phases. The superiority of learning from the network over the cultural belief space can be explained by the heterogeneity of the landscape and the poor performance of individual learners demonstrates that social strategies can be valuable in dynamic environments. Although the experiments tracked the survival of the agents many other aspects such as settlement distribution can be investigated. The study revealed that with just a few essential functional attributes represented, the artifact use learning model can be used to gain insight into a social complex system.

Chapter 7

A Multi-Population Evolutionary Model for Artifact Selection

In this unit, three essential contributions to the thesis are made. First artifact selection, the other facet of artifact exploitation is addressed. Second the scalability of the model is increased by implementing artifact exploitation in a multi-population setting using an MPCA, an extension of CAs. Finally a case study in the Transportation and Injury Prevention domain constituting child auto safety restraints is used to demonstrate artifact exploitation in a new domain dependent MABS.

The model uses the artifact and agent representations from Chapter 3 and provides three learning strategies implemented by LE for artifact selection. The model addresses only artifact selection thus agents are assumed to know how to properly use the selected artifact for an adopted goal, but need to learn which artifact to select. Artifact use plans are therefore assumed to exist while agents must learn artifact selection plans. The learning strategies are developed using GAs, social networks and an MPCA. Agent migration between subpopulations embedded in an MPCA is used to evolve artifact selection knowledge in the social agents. Migration is only supported from advanced subpopulations to underperforming ones where agents in advanced subpopulations are assumed possess knowledge for additional relevant artifacts that agents in underperforming ones do not. Agents in each independently evolving subpopulation are connected via social networks through which knowledge propagates. Each subpopulation is embedded in its own CA whose belief space can further influence its evolution. An implemented MABS constituting of two subpopulations where one subpopulation consistently outperforms the other due to the presence of knowledge about certain restraints is used to conduct experiments with the model. Agent migration with novel restraint selection knowledge from the advanced subpopulation to the underperforming one is investigated. The major aspects of the work here have been previously published in Mokom and Kobti [68].

7.1 Performance Standard and the Critic Element

In order to evaluate an agent's knowledge a domain dependent *PS* is defined as the source of correct knowledge. The correct knowledge is used by *CE* to measure the correctness of the selection. *PS* for artifact selection contains standard knowledge defined with the same structure as the agent's selection knowledge in definition (3.2.6). The standard knowledge sets bits to '1' for each artifact that is supposed to be selected for any given range of a physical attribute of a criteria object. We assume fixed predetermined splits for the criteria ranges given by *PS* and used by all agents. For example, it is possible for artifact selection knowledge data to be extracted from conducted surveys where artifact users provided their knowledge according to given ranges. When an agent's knowledge is compared against the standard selection knowledge in *PS* it is measured according to the number of bits that match. In addition *PS* may provide other domain dependent information that measure the consequences of the artifact selected. This can include for instance the injury level incurred when the wrong artifact is selected or a measure for the positive outcome of selecting the correct one.

7.2 Learning Strategies for Artifact Selection

The artifact selection learning problem is defined in Section 3.2.3. Given a set of artifacts that can be used for realizing its goal and one or more criteria objects related to the artifacts *FDs*, the agent learns or evolves the knowledge K_{sp} for a selection plan *sp* with which it can choose the proper artifact from the set, that is one that will yield an acceptable y_{sp} score.

Artifact selection learning agents are designed to improve through social and cultural mechanisms implemented by *LE*. Learning occurs in the framework of an MPCA consisting of two or more subpopulations. Agents can improve artifact selection through social communication by being members of social networks, through cultural belief space influence or as a result of influence from migrants into their population. In order to accommodate agent heterogeneity with respect to the various influences on the agent's selection knowledge, two rates (similar to the previous chapter) are defined for each agent:

- *Learning rate*: the probability that an agent copies the bit from the influential source correctly.
- *Susceptibility rate:* the probability that an agent is susceptible to influence by an influential source.

7.2.1 Multi-Population Cultural Algorithm Framework

The MPCA for agents learning artifact selection consists of two or more independently evolving social populations. Each population P_i is embedded in its own CA with its own belief space GB_i. Selected individuals from each population P_i contribute their

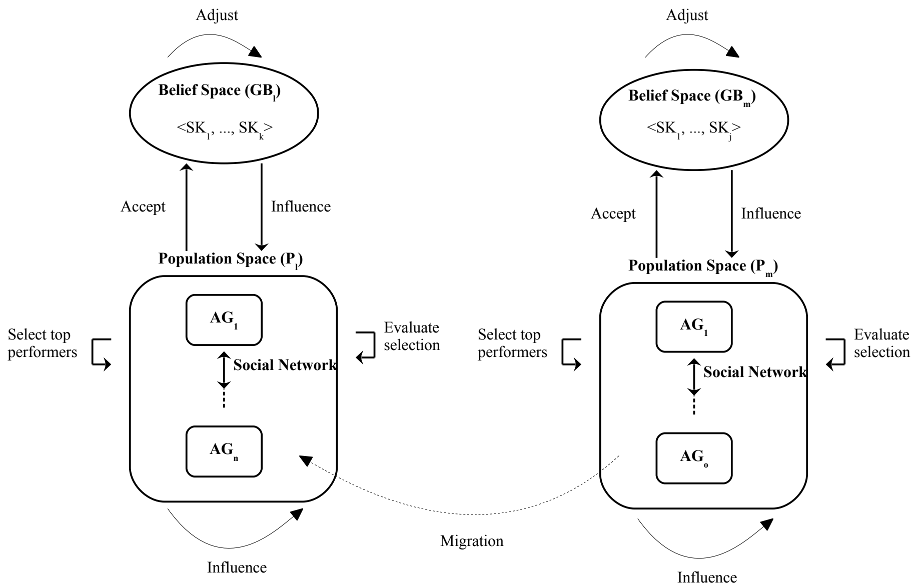

Figure 7.2.1: Multi-Population Cultural Algorithm for social agents belonging to two isolated subpopulations P_l and P_m with respective belief spaces GB_l and GB_m and migration supported from P_m to P_l.

knowledge to their respective belief space GB_i which in turn influences the evolution of the agents in P_i. It is assumed that at least one population will be considered more advanced than others. As such agents from advanced populations may migrate into underperforming ones and affect the evolution of the agents there. An example of the MPCA framework is depicted in Figure 7.2.1 .

The sample framework shows two independently evolving social populations P_l and P_m embedded in CAs with respective GB's GB_l and GB_m. With P_m considered more advanced than P_l, migration is supported from P_m to P_l. Social networks are maintained in each population.

7.2.2 Social Learning via Cultural Algorithms

7.2.2.1 Belief Space Structure

Of the five knowledge sources offered for a CAs belief space, all CAs for agents learning artifact selection maintain only situational knowledge (SK) which consists of the best examples extracted from the corresponding population space. Each CA defines how many exemplars are maintained in its respective GB. A belief space GB_i is defined as: $GB_i = SK_i$. $SK_i = sk_1, \ldots, sk_k$ specifies knowledge for k exemplars with each exemplar $sk_j \in SK_i$ defined as:

$$sk_j = \langle K_j, s_j \rangle \tag{7.2.1}$$

where K_j represents the knowledge for exemplar j selected from P_i and s_j is j's performance score. GB_i is updated at specified intervals when the top k performers are selected from P_i to contribute their knowledge and their performance scores. At any time, GB_i may be probed for the average performance of its exemplars as well as how close their knowledge is to the correct knowledge as defined by PS.

7.2.2.2 Belief Space Adjustment

Each GB_i is responsible for its own adjustment when it accepts knowledge contributed by top performers. The received knowledge is sorted according to its scores. Let $h = \langle K_h, s_h \rangle$ represent a contribution from performer h, and $x = \langle K_x, s_x \rangle$ represent the worst performer (the exemplar with the lowest score) in SK_i then SK_i is adjusted as follows:

$$SK_i' = \begin{cases} (SK_i \cup \{h\}) - \{x\}\,, & s_h > s_x \\ SK_i, & otherwise \end{cases} \tag{7.2.2}$$

Thus each contributor replaces the worst exemplar if it is better or the exemplars are unaffected.

7.2.2.3 Belief Space Influence on the Population Space

Agents in population P_i can be influenced by any randomly chosen exemplar from GB_i. This occurs at any specified intervals during the evolutionary process.When an agent is influenced by an exemplar, LE mutates each bit of the agent's current selection knowledge according to the agent's *susceptibility rate* and *learning rate.* If the agent is susceptible according to the associated rate, then the bit is mutated according to the learning rate.

7.2.3 Social Learning via Social Networks

Agents can improve their knowledge as a result of knowledge propagation through social networks. The social network in the model is defined with two subnetworks that denote relatives or associates and neighbors respectively. An agent may communicate with members of its network which are allowed to contain only agents belonging to its subpopulation. The neighbor subnetwork is defined as other agents residing within a certain distance from the agent. An agent's network of relatives can be any subset of agents up to a predefined maximum. During communication an agent may influence a percentage of its network members. This happens in two possible scenarios: when the agent's knowledge is altered by knowledge from the belief space or when it is a recent migrant to the subpopulation. The LE component of influenced agents accepts new knowledge by mutating the agent's current knowledge according to the agent's *susceptibility* and *learning rates.*

7.2.4 Social Learning via Migration

Assuming two isolated populations in the model as depicted in Figure 7.2.1, agents in P_m are assumed to be more advanced than agents in P_l in terms of knowing and having access to additional useful artifacts. They are therefore expected to maintain a higher performance level on average. At a minimum agents with real criteria objects that require the unknown artifact in P_l should underperform relative to their counterparts in P_m. Agent migration occurs when a randomly selected percentage of agents from P_m migrate to P_l taking their knowledge of the *novel* artifact with them. These migrants are randomly chosen from P_m at large rather than GB_m, therefore their performance is not taken into consideration. The migrants replace an equal number of agents in P_l. Migration is only supported in one direction since the objective is to observe what happens in P_l as knowledge of the *novel* artifact spreads. Once an agent migrates it automatically inherits the networks of the agent it replaces and influences a percentage of its members. When an agent in P_l receives influence that includes knowledge pertaining to a *novel* artifact, it is assumed to now have access to the artifact and so LE adds a bit of knowledge for its representation to the bit sequences of all ranges in its knowledge structure. Agents therefore do not resist the awareness of a new artifact they are exposed to. The value of the added bit of knowledge will however depend on the agent's *learning rate*.

7.3 Case Study: Learning Child Auto Safety Restraint Selection

In this section an application of the artifact selection model to a case study in the domain of Transportation and Injury Prevention is presented. Some aspects of the restraint model in Kobti *et al.* [50] are adopted. The motivation for choosing this particular case study for the model comes from a 2012 report by the Canadian Paediatric Society [109]. The study reported that while motor vehicle collisions were still considered the primary cause of death among Canadian children over one year of age and restraint use had been shown to lower the risk of severe injury by 40% to 60%, the use of incorrect restraints remained high. In particular booster seat use was very low at 30% as children were found to graduate too soon to seat belts. Part of the problem according to the study, was the lack of proper booster seat legislation in 6 out of 13 Canadian provinces and territories, with provinces such as Alberta and Saskatchewan identified as the worst. We explore the idea that while using a booster may not be law in a province or assumed to be unknown to the agent population, a migrated agent from another province where proper legislation exists may bring with it the knowledge and help improve performance of its new population as it evolves.

7.3.1 MABS Model Definition

The MABS model is defined as:

$$S \triangleq \langle P, GB, E \rangle \tag{7.3.1}$$

where $P = \langle P_{booster}, P_{nobooster} \rangle$ specifies two subpopulations with $GB = \langle GB_{booster}, GB_{nobooster} \rangle$ as their corresponding belief spaces. Let $A_{booster}$ be the set of agents belonging to $P_{booster}$ and $A_{nobooster}$ be the set of agents belonging to $P_{nobooster}$ with $|A_{booster}| = |A_{nobooster}|$ and $A_{booster} \cap A_{nobooster} = \emptyset$. Thus the two subpopulations have the same number of agents and an agent can only belong to one subpopulation. Also both GB*'s* maintain the same number of exemplars. The environment is described in terms of artifacts and concrete criteria objects that correspond to the criteria object category abstractions in the FD of artifacts in T as per Formula (3.1.6): $E = \langle T, CNR \rangle$, where each concrete criteria object $cnr \in CNR$ corresponds to a criteria object category. It is assumed that all agents in a subpopulation are aware of and have access to the same set of the environment's components.

Driver agents in both populations $P_{booster}$ and $P_{nobooster}$ know about three types of child auto safety restraints: REAR-FACING, FORWARD-FACING and SEATBELT. The primary distinction between the two populations is that agents in $P_{booster}$ know about one more child restraint BOOSTER. A CHILD criteria object category is defined with three physical attributes namely AGE, WEIGHT and HEIGHT. Each physical attribute is given a specific domain. It is assumed that any restraint the agent chooses for use with a concrete CHILD object is available.

7.3.2 Restraint Selection Knowledge and Assignment Function

The restraint selection knowledge is defined according to Formula (3.2.6). An example of a unit of knowledge for an agent in $P_{nobooster}$ with age ranges given in months is:

$$\begin{aligned} k\left(CHILD, AGE\right) &= \left\langle \left[0,12\right], 0100 \right\rangle \\ & \quad \left\langle \left[13,48\right], 0010 \right\rangle \\ & \quad \left\langle \left[49,96\right], 0001 \right\rangle \\ & \quad \left\langle \left[97,145\right], 0001 \right\rangle \end{aligned} \tag{7.3.2}$$

The first bit in each bit string specifies the option for no artifact selection and each remaining bit corresponds to artifacts the agent knows about in a predefined sequence. Therefore, the second, third and fourth bits correspond to REAR-FACING, FORWARD-FACING and SEATBELT respectively. The first bit for no selection can be interpreted as the driver transporting the child on its lap [50]. In the example, the agent knows that a child whose age inclusively falls 0 and 12 belongs in REAR-FACING, between 13 and 48 in FORWARD-FACING and between 49 and 145 is transported with SEATBELT. An example of a unit of knowledge for an agent in $P_{booster}$ with weight ranges given in pounds is:

$$
\begin{aligned}
k\left(CHILD, WEIGHT\right) \quad = \quad & \left\langle \left[0, 20\right], 01000 \right\rangle \\
& \left\langle \left[21, 40\right], 00101 \right\rangle \\
& \left\langle \left[41, 80\right], 10010 \right\rangle \\
& \left\langle \left[81, 121\right], 00011 \right\rangle
\end{aligned} \qquad (7.3.3)
$$

Agents in $P_{booster}$ have one additional bit of knowledge representing BOOSTER. In the example the agent knows that a child weighing inclusively between 0 and 20 belongs in REAR-FACING, between 21 and 40 in FORWARD-FACING or BOOSTER, between 41 and 80 on the lap or in SEATBELT and between 81 and 121 in BOOSTER or SEATBELT. All agents have additional bits of knowledge for the other CHILD physical attribute HEIGHT. It should be noted that, it is possible for the agent to have units of knowledge for other relevant criteria objects. For instance an agent's restraint selection may vary between 2-door and 4-door vehicles. If vehicle characteristics were considered part of the criteria for selecting a restraint, vehicle would be a criteria object and its corresponding physical attributes would form additional units of knowledge. However the represented criteria object attributes are assumed to be consistent among all agents. The model also stores all agent bit string knowledge in the same predefined sequence using the same predefined ranges for the child's attributes. This is simply to facilitate the implementation of the learning process.

Since an agent will end up with three possible restraint bit strings based on the particular child's age, weight and height values a function is needed as defined in Formula (3.2.7) to determine the artifact the agent finally selects. Given the three bit strings *sage*, *sweight*, and *sheight* an artifact represented by bit i will be selected according to the following formula given by Kobti *et al.* [50]:

$$
\begin{aligned}
sel\,(i) \;\; &= \;\; (sage\,(i)\;\; AND\, sweight\,(i))\;\; OR \\
&\quad\;\; (sage\,(i)\;\; AND\, sheight\,(i))\;\; OR \\
&\quad\;\; (sheight\,(i)\;\; AND\, sweight\,(i))\;\; OR
\end{aligned}
$$

7.3.3 Restraint Selection Evaluation

The simulation is given the correct source of knowledge specifying appropriate artifact selection for children with various age, weight and height. The standard knowledge in *PS* is the one provided in Kobti *et al.* [50]. Agents are evaluated when they have simulated accidents and the result is a score that is updated throughout the simulation indicating the agent's overall performance. *CE* uses the custom scoring function from Kobti *et al.* [50] that associates an injury with each accident according to the agent's artifact selection and calculates a driver performance score (DPS):

$$DPS = \frac{KnowledgeScore}{TotalInjury} * NumAccidents \tag{7.3.4}$$

Essentially drivers who have not been in an accident have an undefined performance. *CE* calculates the knowledge score by matching the agent's knowledge bit strings to *PS* and generating a score between 0 and 1 inclusively: $KnowledgeScore = \frac{\#matchingbits}{total\#bits}$. The injury level for an accident is the sum of the injury for each involved child. *PS* provides 3 possible values for each CHILD injury: 0.1 if the correct restraint was selected, 0.5 if an incorrect restraint was selected and 0.9 if no restraint was selected. $TotalInjury$ reflects the cumulative injury incurred over all accidents. Over time a driver that improves its knowledge and minimizes its injury with each accident should improve its overall performance score.

7.3.4 Model Parameters

Parameters are either fixed or variable according to whether they differ between experiments conducted. As in previous chapters the fixed parameters can be altered to conduct different experiments. The fixed parameters in the model are:

NumberOfAgents The number of learning agents. There are a total of 800 agents with 400 in each population.

NumberOfArtifacts The number of artifacts available in a population. Each agent in $P_{booster}$ gets one of each of the four artifacts. Each agent in $P_{nobooster}$ gets one of each of the three artifacts.

NumberOfCriteriaObjects The number of concrete criteria objects. Each agent gets four objects, one corresponding to each type of restraint.

FunctionalAttributeDomain The domain of the functional attributes: *artr*[0.008421053,0.5198181], *aspect*[0.002658795,359.9978], *dem*[1438.436,3008.686], *depth*[25.2,182.7], *slope*[0.0,49.36105].

ArtifactFD The function description describing the service the agent is to learn and criteria object categories. Each artifact has one *FD* with external goal: *safe_child_transport*, a CHILD criteria object and unpredictable set to 0. CHILD defines physical attribute ranges for AGE, WEIGHT and HEIGHT.

AgentGoal The agent's goal. All agents share the same goal which matches the artifacts' external goal : *safe_child_transport*.

LearningRate The learning rate of an agent, randomly generated at the start.

SusceptibilityRate The susceptibility rate of the agent, randomly generated at the start.

MaxNumberOfRelatives The maximum number of members of an agent's network randomly selected from $[1, 10]$.

NetworkInfluencePercent The percentage of its network members that an agent can influence, fixed at 10%.

NumberOfTopPerformers The number of top performers offered for acceptance into *GB* fixed at 2% of the agent population.

GBUpdateInterval The interval defining when top performers are contributed to *GB* fixed at 7 to mimic every seven days.

PerformanceStandard The *PS* used for evaluating the agent's selection, constituting the correct restraint selection knowledge with predefined ranges used by learning strategies.

MigrationTimeStep Indicates the simulation time step when migration occurs. This is fixed at time step 100 for all migration experiments.

NumberOfTimeSteps The number of time steps in the simulation. The simulation is run for 500 time steps in all experiments.

SelectionRate The probability that an agent selects an artifact in a time step. This denotes the driving probability of a driver in a time step and is fixed at 0.3 [38].

AccidentRate The probability that there is an occurrence which results in the agent's selection being evaluated. This is the probability that a driver gets into an accident fixed at 0.007 as per Canada Motor Vehicle Traffic Collision Statistic 2011 [114].

The variable parameters in the model are:

LearningStrategies Indicates which strategy or strategies are being employed.

MigrationCount The number of agents allowed to migrate. In migration experiments this is either 2% or 5% of the population.

7.3.5 Simulation Flow

The simulation environment is a simple 20 x 20 toroidal grid defined for each subpopulation with 400 agents placed on each grid. Each agent occupies its own square with four children, one corresponding to each possible restraint. With $P_m = P_{booster}$, $P_l = P_{nobooster}$, $GB_m = GB_{booster}$ and $GB_l = P_{nobooster}$, the pseudo code for the simulation is depicted in Algorithm 6.

The simulation begins with the initialization of the subpopulations and their GBs. This involves randomly generated agents given randomly generated CHILD objects and one of each type of artifact known to their respective population. The 8 agents occupying each agents Moore neighborhood are used to construct its neighbor subnetwork. Each agent's relative subnetwork is formed by randomly members of its subpopulation. Agents formulate the *safe_ child_ transport* goal, initialize their selection plan and learning begins. Every time step an agent decides to drive according to the driving probability, selects a restraint or none for each child based on its current knowledge and may get in an accident according to the accident rate. An agent can be influenced by any exemplar from the belief space and in turn influence a percentage its social network members. At given intervals the belief spaces are updated with knowledge obtained from a percentage of their corresponding population according to the DPS scores. Migration can occur at any chosen time step, after which migrants inherit the social network of the agents they replace and immediately influence a percentage of the network members.The simulation runs for any specified duration.

Algorithm 6 Pseudo-code for learning artifact selection with Multi-Population Cultural Algorithm

1. Initialize P_m,P_l,B_m,B_l
2. Generate social networks in P_m,P_l
3. Each agent's *LE* formulates goal and *PE* activates goal
4. Each agent's *PE* initializes its capability with knowledge (*possibly random*) and shares copy with *LE*
5. Each agent is given a set of concrete criteria objects and the same set of artifacts
6. The following is repeated for every *time_ step* in the evolutionary process
   (a) Each *PE* selects artifacts according to its current knowledge (*can depend on selection rate*)
   (b) If artifact selection occurs in (a), then the agent may be evaluated (*can depend on an accident rate*)
   (c) If the agent is to be evaluated per (b), *CE* evaluates artifact selection against *PS* and gives feedback to *LE*
   (d) *LE* updates agent's performance score
   (e) *LE* knowledge is influenced by a randomly chosen exemplar in GB_i (*according to retention and learning rates*)
   (f) If influence occurred in (e), *LE* influences a percentage of its social network
   (g) If agent is influenced by another network member, *LE* mutates knowledge (*according to retention and learning rates*)
   (h) If migration allowed this *time_ step*, migrate r agents from P_m to P_l
   (i) If migration occurred in (h):
      i. Each migrant inherits social network of agent it replaces
      ii. Each migrant's *LE* influences a percentage of its inherited social network
   (j) If GB_i updates allowed this *time_ step*:
      i. Search P_i for top performers (using performance scores in *LE*)
      ii. Accept top performers in GB_i and adjust
   (k) If termination *time_ step*: END.

7.3.6 Experiments and Results

Experiments are performed for six different settings. The variation is based on the presence or absence of migration, the percentage of agents that migrate and the source of influential knowledge. The two primary sources of influence are the belief space only or a combination of the belief space and social network.

In the first two test cases the model is tested in the absence of migration capturing the performances of agents in $P_{booster}$ and $P_{nobooster}$. The objective is to observe whether the knowledge of the BOOSTER artifact has a positive effect on the performance of agents in $P_{booster}$ when compared to those in $P_{nobooster}$ as the subpopulations independently evolve. This is important since we have presumed that performance on average in $P_{booster}$ should exceed that of $P_{nobooster}$. Agents are only influenced by the belief space in the first test case and in the second they are influenced by both the belief space and their social network. Results are depicted in Fig. 7.3.1 and Fig. 7.3.2 respectively.

The remaining four test cases measure migration effects on agents in $P_{nobooster}$. In the experiments the subpopulations evolve without migration for the first 100 time steps. At time step 100 a percentage of agents migrate from $P_{booster}$ to $P_{nobooster}$. In the first scenario 2% of agents migrate with belief space as the only source of influence. In the second the social network is enabled and agents receive influence from both the belief space and their network. Results for test cases 3 and 4 are shown in Fig. 7.3.3. The last two experiments involve 5% of agents migrating with belief space influence only and a combined belief space and social network influence respectively. The results are depicted in Fig. 7.3.4.

For all test cases the cumulative average of driver performance scores (DPS) for a subpopulation over time are calculated and plotted. In addition we measure and plot the average $KnowledgeScore$ of the exemplars in the belief space for the respective population.

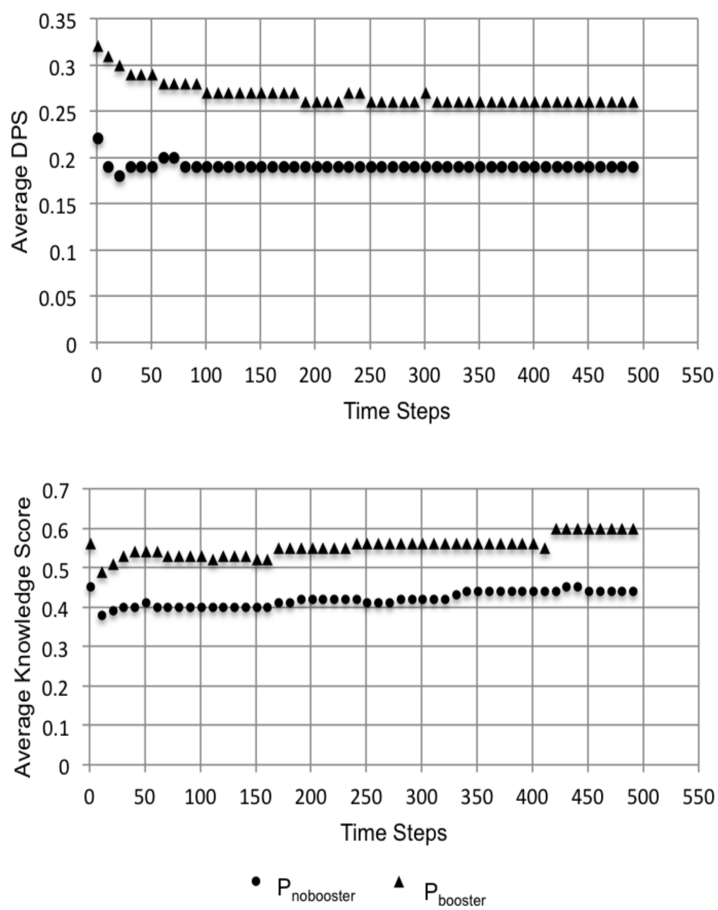

Figure 7.3.1: The population's average driver performance score (top) and the average knowledge score of the belief space (bottom) over time with belief space influence in the absence of migration. © [2014] IEEE

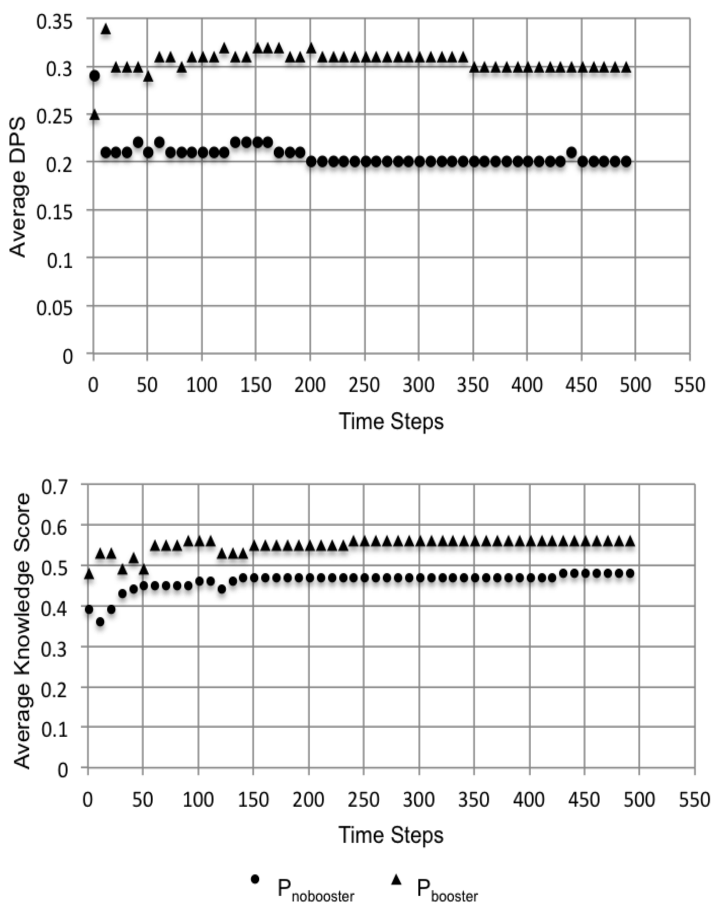

Figure 7.3.2: The population's average driver performance score (top) and the average knowledge score of the belief space (bottom) over time with belief space and social network influence in the absence of migration. © [2014] IEEE

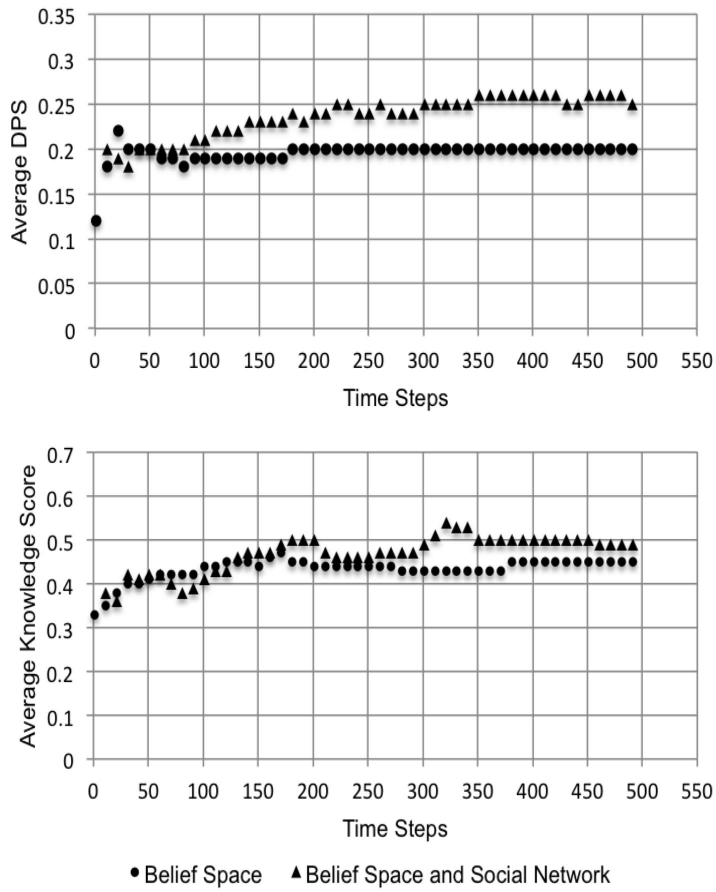

Figure 7.3.3: The population's average driver performance score (top) and the average knowledge score of the belief space (bottom) over time for $P_{nobooster}$ with 2% migration at time step 100. © [2014] IEEE

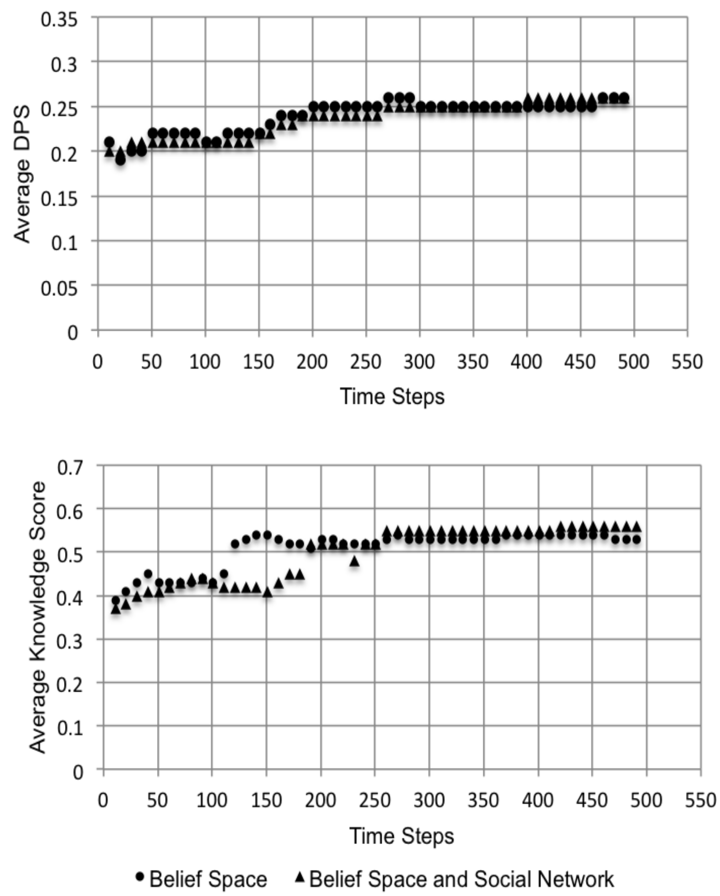

Figure 7.3.4: The population's average driver performance score (top) and the average knowledge score of the belief space (bottom) over time for $P_{nobooster}$ with 5% migration at time step 100. © [2014] IEEE

7.3.7 Discussion

The driver performance score (DPS) considers the correctness of an agent's knowledge as compared to the standard knowledge source, the combined injury levels for its children as accidents occur and the number of accidents that the agent has been in. Undefined DPS (for agent's that have not been in an accident) are neither contributed to the belief spaces nor included in the cumulative averages. The knowledge scores depicted in the results are obtained from the exemplars in the respective population's belief space. Although DPS is used to determine which examples denote the top performers for acceptance into the belief space, the knowledge scores provide a measurement for the knowledge level of the exemplars.

In Fig. 7.3.1 which depicts results for evolution in the absence of migration it can be observed that agents in $P_{booster}$ consistently outperform their counterparts in $P_{nobooster}$. These agents maintained a higher DPS throughout the simulation regardless of the source of influential knowledge supporting our presumption that with respect to overall performance $P_{booster}$ should be superior to $P_{nobooster}$. This is expected as children that belong in a booster will always suffer above minimal injuries in $P_{nobooster}$ since its agents are unaware of the restraint. It can also be observed that the knowledge score of the belief space exemplars is higher for $P_{booster}$ than for $P_{nobooster}$. This is in accordance with our expectations since agents in $P_{booster}$ have knowledge about an additional useful artifact. The agent's performance and exemplar knowledge seems to find and maintain a steady level throughout for both subpopulations. This correlates with other studies [50, 52, 53] that demonstrate the emerged resistance of culturally evolving agents to change, in the absence of external intervening sources.

Figs. 7.3.3 and 7.3.4 depict results obtained from migration experiments. In Fig. 7.3.3 there is no improvement in average DPS when agents are only influenced by the belief space as 2% of $P_{booster}$ agents migrate to $P_{nobooster}$ at time step 100. In contrast agents show an improvement when they are influenced by a combination of the belief space and social network. This can be explained by the fact that when the social network is disabled and the belief space is the only source of influence, a migrant must be selected as an exemplar before any non migrant agent in the population can learn about the booster. Since the selection of agents for migration from $P_{booster}$ is random and from the subpopulation at large rather than $P_{booster}$'s belief space, it is possible that the migrants may not be good enough for $P_{nobooster}$'s belief space or may have undefined DPSs. When the social network is enabled, migrants influence members of their inherited network upon migration. This results in the immediate propagation of booster restraint knowledge and increases the chances of an agent with that knowledge to be selected as an exemplar thus affecting the rest of the population. The average knowledge scores of $P_{nobooster}$ exemplars also improves with better scores observed for the combined influence than the influence from belief space only. In Fig. 7.3.4 we observe an improvement of DPS scores for both types of influences. In this scenario 5% of agents migrate thus increasing the chances that a migrant agent is selected as an exemplar. Accordingly knowledge scores for exemplars in the belief space also shows an improvement. Subsequent to the observed increases in DPS and knowledge scores the effects of cultural beliefs start to become evident as once again the resistance to change starts to show. However the agents continue to perform better than they did prior to the occurrence of migration. It is important to note that the observed trends emerge despite the fact that the agent's restraint knowledge along with their learning and retention rates were generated randomly at the start of all six experiments.

7.4 Conclusions

In this chapter we have provided a model for artifact selection evolution in MABS. As a result in combination with the previous chapters we have addressed the proposed aspects of artifact exploitation: artifact selection and artifact use. Agents learn artifact selection through social networks, at population levels using CAs and at multi-population levels using MPCA. The MPCA consists of two independently evolving subpopulations where agents in one subpopulation consistently outperform their counterparts in the other due to an enhanced knowledge of artifacts in a particular domain. The effects of agents migrating from the more knowledgeable population to the other are examined. In particular the objective is to determine whether agents in the underperforming subpopulation can improve their performance as a result of the arrival of migrants with knowledge of useful artifacts. A domain dependent MABS characterized by the selection of child auto safety restraints has been utilized to implement the model.

Results from conducted experiments show that in the absence of migration, the performance of agents aware of all useful artifacts consistently surpass that of agents with a missing artifact as both subpopulations independently evolve over time. When migration occurs it is shown that the arrival of knowledge about the novel artifact may not have an effect when the CA belief space is the only source of influence. The effect may depend on the percentage of agents that migrate and the quality of their knowledge and performance. With both the belief space and social network as sources of influence, knowledge about the novel artifact is propagated and the agents performance scores improve accordingly. Overall the results demonstrate that social agents can improve their knowledge about artifact selection in the absence of interventions from a standard external correct knowledge source as in Gupta *et al.* [38] and Kobti *et al.* [50, 54].

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this dissertation we presented a computational model for integrating artifacts into MABS in order to enhance the abilities of its embedded social agents to realize their goals. The model is based on established theories in agent-oriented computing that propose artifacts as an abstraction for functional system components that proactive agents with reasoning and planning capacities can exploit. The theories supply the necessary concepts for accommodating exploitable objects in complex systems, which can provide significant benefits especially to systems where tool use is directly related to the evolution of the society and its overall performance. Our model promotes an evolutionary approach to address artifact exploitation by MABS agents so that better insight can be gained into their effects on the system over time. This is realized by extending the artifact theories to support agents that can evolve artifact selection and use by employing various learning and adaptation strategies. We emphasize on those strategies that take advantage of the social dimensions of MABS, where agents evolve through influence from others in the environment. Artifacts are reduced to a set of functional attributes whose values can be evolved by applying computational intelligence methods, specifically genetic algorithms, cultural algorithms and multi-population cultural algorithms.

Prior models that have proposed an evolutionary approach to artifacts in MABS have been restricted to a particular domain and have addressed limited aspects of artifact exploitation. The scalability of our model is evident in its support for artifact exploitation at individual, population and multi-population levels. The model is versatile accommodating heterogeneous agents, static and dynamic artifacts, dynamic environments and agent influence through static or dynamic social networks and cultural beliefs. Consequently artifact exploitation is modeled in the context of social and cultural evolution. Although many different parameters are used, the model is flexible as they are only fixed in conducted experiments and can be altered for studies with different objectives.

A generic MABS was built to conduct various experiments with different aspects of the model. The superiority of learned artifact use over random use was demonstrated and social learning was shown to consistently outperform individual learning, similar to findings by social scientists [32]. Two case studies were provided to demonstrate the applicability of the model.

In the first case study, the model was incorporated into the existing MABS of the Village EcoDynamics Project developed over decades to study the lives of the ancient Pueblo Indian settlers in the American Southwest during a period spanning 700 years. The Village MABS models a very complex social system where artificial social agents represent households that farm, hunt, gather wood and water and employ various exchange models for trading. This is all realized by agents using reasoning and planning capabilities. The Village MABS models many environmental aspects such as soil productivity, rainfall, animal and forest density and is used for studying settlement distribution, violence and the demography of the population. The landscape exploited by agents for farming was abstracted as an artifact. In collaboration with a team of archaeologists working on the project, we eliminated the presumption that knowledge for the productive farming areas of the landscape was embedded in agents. Instead agents were made to use the learning and adaptation strategies provided by our artifact model to evolve their knowledge for exploiting the landscape over time. The dynamic and heterogeneous nature of the landscape, the mobility of its inhabitants and the dynamic environment where agents enter and leave through marriages and deaths respectively provided an excellent testbed for our model. Conducted experiments that tracked agent survival through archaeological phases have already been shown to be consistent with some of the archaeological findings. For instance results depicted by the most successful adaptation strategy where agents learned socially while concurrently evolving the radius of their social network was noted to align with some of their observations. More importantly with the model in place other objects can now be abstracted as artifacts and many other aspects of the system can be investigated.

In the second case study our model was used to build a new domain dependent MABS. Artifact exploitation was explored for child auto safety restraints where agents learned to select the proper restraints for particular children. The model experimented with agent migration between independently evolving populations of social agents as a means for improving knowledge in underperforming populations. Experiments were conducted to observe the propagation of knowledge in each population in the presence and absence of migration. Results showed that the knowledge level of the migrants as well as the number of migrants played a role in the impact of migration.

It is my belief that our model for integrating artifacts in new and existing MABS systems is a valuable contribution to agent-oriented computing. In addition to the case studies depicted in this thesis we have demonstrated the application of the model in the investigation of social phenomena such as social norms [66, 67] and social inhibition [65]. To the best of my knowledge it is the first domain independent model for integrating artifacts in MABS, that is based on artifact theories and specifically addresses the evolution of artifact exploitation by social agents.

8.2 Limitations

It is worth noting that it will not always be practical to use our model. One of the fundamental aspects of the model is that in order to exploit an artifact it is reduced to a set of functional attributes whose values can be evolved over time. Consider the artifact *pen.* Its functional attributes include *hold, point, press, move* and so on. A combination of these attributes that forms a use action in our model would constitute a single value selected from each of the variables. The study of tool use is inherently complicated by the fact that these attributes can be numerous. If one were to take account all the possibilities available to the holder of a pen, or other objects that may impact its use as well as spatial and temporal considerations the result would be a very large problem space. In the first implementation of our model presented in Chapter 4 it was observed that on average an agent learning to use an artifact represented with 8 attributes where each required a fixed value for success, needed 168,239 simulation time steps to reach its goal. Although these numbers are greatly reduced when the social dimensions are utilized for learning, it is apparent that learning artifact exploitation for artifacts with numerous attributes may prove intractable with the model. We have shown with the case studies however that the model can be very useful with just a few essential attributes selected by experts in the modeled domain. Moreover the primary objective of the model is to explore how artifact exploitation evolves as a result of the different ways agents could learn.

There are also many parameters in the model that could play a role in the outcome of agent performance. To circumvent this we have used averages as much as possible in our experiments to ensure that results are consistent. Also, many of these parameters are not fixed. They can be altered with different experiments and tuned by social scientists to examine their impact, if any.

8.3 Future Work

The cultural algorithms implemented in our model have only used two of the five suggested knowledge sources to characterize belief knowledge extracted from the population. Research has shown that these knowledge sources when combined can inform on how evolving populations actually learn to solve problems [99]. It would be useful to augment our artifact model with the other knowledge sources, historical, topographical and domain. The integration of the latter, domain knowledge, may be facilitated with an ontology that characterizes the centralized knowledge of artifacts in the environment and the complementary artifact capabilities local to agent. We have proposed such a hybrid ontology [64] and intend to implement and integrate it into our model.

We have really only addressed the use of a single artifact, however artifacts are usually used in conjunction with others. This concerns instances where both artifacts provide functionality towards realizing a goal. For instance the use of a bow and arrow requires functional attribute values selected from both artifacts. It may be possible to implement this in our model by including all attributes of all involved artifacts. However many other aspects such as temporal or artifact collaboration concerns may come into play.

Another aspect that might be worth exploring with the model is how agents could generalize artifact use knowledge across different artifacts. Agents could learn to use particular artifacts then apply the knowledge to new artifacts and extract knowledge about similarities and differences both in the artifacts' structures and their behavior.

An agent could then gain competence against a sets of artifacts learning how to combine or substitute them at different phases of realizing its goal.

Another possible direction for future work could be to deepen the representations of other components of the performance element. Goals and beliefs could be extended such that learning methods could be developed for improving them along with capabilities over time. For instance, behavioral traits could be used to define an agent's beliefs. This would extend the model to define an agent's choice of actions according to its behavior and knowledge hence paving the way for employing methods towards altering an agent's behavior with respect to artifact exploitation.

It may also be useful to explore reducing the cognitivity of the artifacts even further. For instance, agents could be built to learn the functional attributes of a usage interface or learn external goals and relevant criteria objects. Effectively this would result in a larger search space.

Finally the third facet of artifact exploitation namely artifact construction and manipulation has not been addressed. It should be useful to explore this leading to studying innovation with respect to artifacts. This is intertwined with economic aspects such as demand as the failure of artifacts to meet agent needs creates a need for a new artifact or the modification of an existing one.

Bibliography

[1] The repast suite. `http://repast.sourceforge.net/`, 2013.

[2] Daghan L. Acay, Philippe Pasquier, and Liz Sonenberg. Extrospection: Agents reasoning about the environment. In *Proceedings of the 3rd International Conference on Intelligent Environments*, pages 220–227, 2007.

[3] Daghan L. Acay, Liz Sonenberg, Alessandro Ricci, and Philippe Pasquier. *How Situated Is Your Agent? A Cognitive Perspective*, pages 136–151. Springer-Verlag, Berlin, Heidelberg, 2009.

[4] Daghan L. Acay, Gil Tildar, and Liz Sonenberg. Extending agent capabilities: Tools vs Agents. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 259–265. IEEE Computer Society, Los Alamitos, CA, 2008.

[5] J. Alami, A. El Imrani, and A. Bouroumi. A multipopulation cultural algorithm using fuzzy clustering. *Applied Soft Computing*, 7(2):506–519, 2007.

[6] Kathrin Bacher, Simon Allen, Anna K. Lindholm, Lars Bejder, and Michael Krützen. Genes or culture: Are mitochondrial genes associated with tool use in bottlenose dolphins (*Tursiop* sp.)? *Behavior Genetics*, 40:706–714, 2010.

[7] Lynne Rudder Baker. The ontology of artifacts. *Philosophical Explorations*, 7(2):99–112, 2004.

[8] Albert Bandura. *Social foundations of thought and action: a social cognitive theory*. Prentice Hall series, 1986.

[9] Albert-László Barabási. Scale-free networks: A decade and beyond. *Science*, 325(5939):412–413, 2009.

[10] Ricardo Landa Becerra and Carlos A Coello Coello. Cultured differential evolution for constrained optimization. 195(33):4303–4322, 2006.

[11] Benjamin B. Beck. *Animal Tool Use Behavior: The Use and Manufacture of Tools by Animals*. Garland Press, New York, NY, 1980.

[12] Sergio Bernardes, Allison Howard Eury, Andrea Presotto, Marguerite Madden, Thomas Jordan, Dorothy M Fragaszy, Patricia Izar, and Yuri Tavares-Rocha. An agent based modeling approach for representing capuchin (sapajus spp.) behavior in brazil. In *Proceedings Of The Annual Conference- American Society For Photogrammetry AND Remote Sensing*, pages 41–47, 2011.

[13] Christopher Best, Xiangdong Che, Robert G Reynolds, and Dapeng Liu. Multi-objective cultural algorithms. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–9, 2010.

[14] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies–a comprehensive introduction. *Natural Computing*, 1(1):2–52, 2002.

[15] Ergun Bicici and Robert St. Amant. Reasoning about the functionality of tools and physical artifacts. *Department of Computer Science, North Carolina State University, Tech Rep*, 22, 2003.

[16] William Bluethmann, Robert Ambrose, Myron Diftler, Scott Askew, Eric Huber, Michael Goza, Fredrik Rehnmark, Chris Lovchik, and Darby Magruder.

Robonaut: A robot designed to work with humans in space. *Autonomous Robots*, 14(2):179–197, 2003.

[17] Olivier Boissier, Rafael H Bordini, Jomi F Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with JaCaMo. *Science of Computer Programming*, 78(6):747–761, 2013.

[18] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3):7280–7287, 2002.

[19] Rafael H Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*, volume 8. John Wiley & Sons, Inc., 2007.

[20] Michael E. Bratman. *Intentions, Plans and Practical Reason*. Harvard University Press, Cambridge, MA, 1987.

[21] Solly Brown and Claude Sammut. A relational approach to tool-use learning in robots. In *Inductive Logic Programming*, pages 1–15. Springer, 2013.

[22] Beth S. Bruce, Anne W. Snowdon, Charles Cunningham, Carolyn L. Cramm, Krista Whittle, Heather Correale, Melanie Barwick, Caroline Piotrowski, Lynne Warda, and Jessie Harold. Predicting parents' use of booster seats. *Injury Prevention*, 17(5):313–318, 2011.

[23] Richard W. Byrne. The manual skills and cognition that lie behind hominid tool use. In Anne E. Russon and David R. Begun, editors, *The Evolution of Thought*. Cambridge University Press, Cambridge, England, 2004.

[24] Chan-Jin Chung and Robert G. Reynolds. CAEP: an evolution-based tool for real-valued function optimization using cultural algorithms. *International Journal on Artificial Intelligence Tools*, 7(3):239–291, 1998.

[25] Carlos A. Coello Coello and Ricardo Landa Becerra. Efficient evolutionary optimization through the use of a cultural algorithm. *Engineering Optimization*, 36(2):219 – 236, April 2004.

[26] Carlos A Coello Coello and Ricardo Landa Becerra. Evolutionary multiobjective optimization using a cultural algorithm. In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, pages 6–13. IEEE, 2003.

[27] Jason G Digalakis and Konstantinos G Margaritis. A multipopulation cultural algorithm for the electrical generator scheduling problem. *Mathematics and Computers in Simulation*, 60(3):293–301, 2002.

[28] Alexis Drogoul and Jacques Ferber. Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. In *Artificial Social Systems*, pages 2–23. Springer, 1994.

[29] Agoston E Eiben and James E Smith. *Introduction to evolutionary computing*. Springer Science & Business Media, 2003.

[30] Luca Gardelli, Mirko Viroli, Matteo Casadei, and Andrea Omicini. Designing self-organising environments with agents and artefacts: a simulation-driven approach. *International Journal of Agent-Oriented Software Engineering*, 2(2):171–195, 2008.

[31] Luca Gardelli, Mirko Viroli, and Andrea Omicini. Combining simulation and formal tools for developing self-organizing MAS. *Multi-Agent Systems: Simulation and Applications, Computational Analysis, Synthesis, and Design of Dynamic Systems*, pages 133–165, 2009.

[32] Amy K. Gardiner, David F. Bjorklund, and Marissa L. Greif ad Sarah K. Gray. Choosing and using tools: Prior experience and task difficulty influence preschoolers' tool-use strategies. *Cognitive Development*, 27(3):240–254, 2012.

[33] Kathleen R. Gibson and Tim Ingold. *Tools, language and cognition in human evolution*. Cambridge University Press, 1994.

[34] Nigel Gilbert. Agent-based social simulation: dealing with complexity. *The Complex Systems Network of Excellence*, 9(25):1–14, 2004.

[35] Leilani Gilpin, Laurent Ciarletta, Yannick Presse, Vincent Chevrier, and Virginie Galtier. Co-simulation solutions using AA4MM-FMI applied to smart space heating models. In *Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques*, pages 153–159. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[36] Yi-nan Guo, Dandan Liu, and Jian Cheng. Multi-population cooperative cultural algorithm. In *Bio-inspired Computing and Applications*, pages 199–206. Springer, 2012.

[37] Ritwick Gupta. Validated agent-based model using predictive data mining and intervention policy testing framework: A case study in child vehicle safety. Master's thesis, University of Windsor, Windsor, Canada, 2013.

[38] Ritwick Gupta, Ziad Kobti, and Robert Kent. Child vehicle safety simulation using regression analysis and predictive data mining. In *Proceedings of the 5th KES Internation Conference on Intelligent Decision Technologies*, volume 255, pages 94–103, 2013.

[39] Lynne Hamill and Nigel Gilbert. Social circles: A simple structure for agent-based social network models. *Journal of Artificial Societies and Social Simulation*, 12(2), 2009.

[40] Andrew Hlynka and Ziad Kobti. Knowledge sharing through agent migration with multi-population cultural algorithm. In *The Twenty-Sixth International FLAIRS Conference*, 2013.

[41] Catherine Hobaiter, Timothée Poisot, Klaus Zuberbühler, William Hoppitt, and Thibaud Gruber. Social network analysis shows direct evidence for social transmission of tool use in wild chimpanzees. *PLoS Biology*, 12(9):e1001960, 2014.

[42] John H. Holland. *Adaptation in natural and artificial systems*. Univ of Michigan Press, 1975.

[43] Jomi F Hubner, Jaime S Sichman, and Olivier Boissier. Developing organised multiagent systems using the moise+ model: programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3):370–395, 2007.

[44] Raghvendra Jain and Tetsunari Inamura. Bayesian learning of tool affordances based on generalization of functional feature to estimate effects of unseen tools. *Artificial Life and Robotics*, 18(1-2):95–103, 2013.

[45] Hong Jiang, Waldemar Karwoski, and Tareq Z. Ahram. Applications of agent-based simulation for huma socio-cultural behavior modeling. *Work: A Journal of Prevention, Assessment and Rehabilitation*, 41:2274–2278, 2012.

[46] Xidong Jin and Robert G. Reynolds. Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algo-

rithm approach. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1672 – 1678, 1999.

[47] Kenneth A. De Jong. *Evolutionary Computation: A Unified Approach.* MIT Press, 2005.

[48] Robert D Kent, Ziad Kobti, Anne Snowdon, and Akshai Aggarwal. Towards a unified data management and decision support system for health care. In *Intelligent Interactive Multimedia Systems and Services*, pages 205–220. Springer, 2010.

[49] David Kirsh. Explaining artifact evolution. In Lambros Malafouris and Colin Renfrew, editors, *The Cognitive Life of Things*, pages 1–37. Cambridge University Press, Cambridge, England, 2010.

[50] Z. Kobti, A.W. Snowdon, S. Rahaman, T. Dunlop, and R.D. Kent. A cultural algorithm to guide driver learning in applying child vehicle safety restraint. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1111–1118, 2006.

[51] Ziad Kobti and Robert G Reynolds. Modeling protein exchange across the social network in the village multi-agent simulation. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 4, pages 3197–3203, 2005.

[52] Ziad Kobti, Robert G. Reynolds, and Tim Kohler. A multi-agent simulation using cultural algorithms: the effect of culture on the resilience of social systems. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1988–1995, 2003.

[53] Ziad Kobti, Robert G Reynolds, and Tim Kohler. The effect of kinship cooperation learning strategy and culture on the resilience of social systems in the

village multi-agent simulation. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1743–1750. IEEE, 2004.

[54] Ziad Kobti, Anne Snowdon, Shamual Rahaman, Tina Dunlop, and Robert Kent. Modeling the effects of social influence on driver behavior in applying child vehicle safety restraint. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 145–150. IEEE, 2006.

[55] Timothy A Kohler. News from the northern american southwest: Prehistory on the edge of chaos. *Journal of Archeological Research*, 1(4):267–321, 1993.

[56] Timothy A Kohler. The final 400 years of pre-hispanic agricultural society in the mesa verde region. *Kiva*, 66:191–264, 2000.

[57] Timothy A Kohler, C David Johnson, Mark Varien, Scott Ortman, Robert Reynolds, Ziad Kobti, Jason Cowan, Kenneth Kolm, Schaun Smith, and Lorene Yap. Settlement ecodynamics in the prehispanic central mesa verde region. *The model-based archaeology of socionatural systems*, pages 61–104, 2007.

[58] Timothy A Kohler and Mark D Varien. *Emergence and collapse of early villages: Models of central Mesa Verde archaeology*, volume 6. Univ of California Press, 2012.

[59] Wolfgang Köhler. *The mentality of apes.* Harcourt, Brace, and Co., New York, NY, 1931.

[60] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[61] Vishwanathan Mohan and Pietro Morasso. How past experience, imitation and practice can be combined to swiftly learn to use novel tools: Insights from

skill learning experiments with baby humanoids. In *Biomimetic and Biohybrid Systems*, pages 180–191. Springer, 2012.

[62] Felicitas Mokom and Ziad Kobti. A cultural evolutionary model for artifact capabilities. In Tom Lenaerts, Mario Giacobini, Hugues Bersini, Paul Bourgine, Marco Dorigo, and René Doursat, editors, *Advances in Artificial Life, ECAL 2011: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, pages 542–549, Paris, France, August 2011. MIT Press.

[63] Felicitas Mokom and Ziad Kobti. Evolution of artifact capabilities. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 476–483, New Orleans, Louisiana, June 2011. IEEE Press.

[64] Felicitas Mokom and Ziad Kobti. Learning artifact capabilities via a hybrid ontology. In *Twenty-Fifth International FLAIRS Conference*, pages 367–370. AAAI Press, 2012.

[65] Felicitas Mokom and Ziad Kobti. Effects of social inhibition on selection of artifact capabilities. In *The Twenty-Sixth International FLAIRS Conference*, pages 100–105. AAAI Press, 2013.

[66] Felicitas Mokom and Ziad Kobti. Interventions via social influence for emergent suboptimal restraint use. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 1(5):25–35, 2013.

[67] Felicitas Mokom and Ziad Kobti. Suboptimal restraint use as an emergent norm via social influence. In *Distributed Computing and Artificial Intelligence*, pages 545–552. Springer, 2013.

[68] Felicitas Mokom and Ziad Kobti. Improving artifact selection via agent migration in multi-population cultural algorithms. In *Swarm Intelligence (SIS), 2014 IEEE Symposium on*, pages 1–8. IEEE, Dec 2014.

[69] Felicitas Mokom and Ziad Kobti. Exploiting objects as artifacts in multi-agent based social simulations. To appear in *Proceedings of The International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2015.

[70] Ambra Molesini, Enrico Denti, and Andrea Omicini. Homemanager: Testing agent-oriented software engineering in home intelligence. In *Agents and Artificial Intelligence*, pages 205–218. Springer, 2010.

[71] Sara Montagna, Alessandro Ricci, and Andrea Omicini. A&a for modelling and engineering simulations in systems biology. *International Journal of Agent*, 2(2):222–245, 2008.

[72] Heina Muhlenbein. The breeder genetic algorithm-a provable optimal search algorithm and its application. In *Applications of Genetic Algorithms, IEE Colloquium on*, pages 5–1. IET, 1994.

[73] Yi nan Guo, Jian Cheng, Yuan yuan Cao, and Yong Lin. A novel multi-population cultural algorithm adopting knowledge migration. *Soft Computing*, 15(5):897–905, 2011.

[74] Jason Noble and Daniel W. Franks. Social learning mechanisms compared in a simple environment. In *Proceedings of the 8th International Conference on Artificial Life*, pages 379–385. MIT Press, Cambridge, MA, 2002.

[75] Jason Noble and Daniel W. Franks. Social learning in a multi-agent system. *Computing and Informatics*, 22(6):561–574, 2003.

[76] Alberto Ochoa, Arturo Hernández, Fernando Montes, Julio Ponce, Laura Cruz, Lenka Janacek, and Liang Li. *Artificial societies and social simulation using ant colony, particle swarm optimization and cultural algorithms*. INTECH Open Access Publisher, 2010.

[77] Andrea Omicini and Stefano Mariani. Agents & multiagent systems: en route towards complex intelligent systems. *Intelligenza Artificiale*, 7(2):153–164, 2013.

[78] Andrea Omicini, Michele Piunti, Alessandro Ricci, and Mirko Viroli. Agents, intelligence and tools. In *Artificial Intelligence An International Perspective*, pages 157–173. Springer, 2009.

[79] Andrea Omicini, Alessandro Ricci, and Mirko Viroli. Agens Faber: Toward a theory of artefacts for MAS. *Electronic Notes in Theoritical Computer Sciences*, 150(3):21–36, 2006.

[80] Andrea Omicini, Alessandro Ricci, and Mirko Viroli. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17(3):432–456, 2008.

[81] Wendell H Oswalt. *Habitat and technology: the evolution of hunting*. Holt, Rinehart and Winston New York, 1973.

[82] Lin Padgham and Patrick Lambrix. Formalisations of capabilities for bdi-agents. *Autonomous Agents and Multi-Agent Systems*, 10:249–271, May 2005.

[83] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

[84] Bin Peng, RG Reynolds, and J Brewster. Cultural swarms. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 3, pages 1965–1971. IEEE, 2003.

[85] PP González Pérez, A Omicini, and M Sbaraglia. A biochemically inspired coordination-based model for simulating intracellular signalling pathways. *Journal of Simulation*, 7(3):216–226, 2013.

[86] Henry Petroski. *The evolution of useful things.* Knopf, New York, NY, 1992.

[87] Michele Piunti. *Designing and Proramming Organizational Infrastructures for Agents Situated in Artifact-Based Environments.* PhD thesis, Universita di Bologna, 2010.

[88] Michele Piunti, Andrea Santi, and Alessandro Ricci. Programming soa/ws systems with bdi agents and artifact-based environments. *Proceedings of Agents, Web Services and Ontologies, Integrated Methodologies Important Dates and Instructions (AWESOME-09)*, 2009.

[89] Thomas Plummer. Flaked stones and old bones: biological and cultural evolution at the dawn of technology. *American Journal of Physical Anthropology*, 125:118–164, 2004.

[90] Thomas G. Power. *Play and Exploration in Children and Animals.* Lawrence Erlbaum Associates, Mahwah, NJ, 2000.

[91] Beth Preston. Cognition and tool use. *Mind and Language*, 13(4):513–547, 1998.

[92] Anand S. Rao and Michael P. Georgeff. BDI agents: From theory to practice. In *ICMAS*, pages 312–319, 1995.

[93] Colin R. Reeves and Jonathan E. Rowe. *Genetic algorithms: principles and perspectives: a guide to GA theory*. Number 20. Springer Science & Business Media, 2003.

[94] Robert Reynolds and Dapeng Liu. Multi-objective cultural algorithms. In *Evolutionary Computation, 2011., IEEE International Conference on*, pages 1233–1241, 2011.

[95] Robert G. Reynolds. *An Adaptive Computer Model of the Evolution of Agriculture for Hunter-gatherers in the Valley of Oaxaca*. PhD thesis, Dept. of Computer Science, University of Michigan, 1979.

[96] Robert G. Reynolds. An introduction to cultural algorithms. In *Proceedings of the third annual conference on evolutionary programming*, pages 131–139, 1994.

[97] Robert G. Reynolds. *Time, Process, and Structured Transformation in Archaeology*, chapter Why Does Cultural Evolution Proceed at a Faster Rate Than Biological Evolution, pages 269–282. Routledge Press, New York, NY, 1997.

[98] Robert G Reynolds and ChanJin Chung. Knowledge-based self-adaptation in evolutionary programming using cultural algorithms. In *Evolutionary Computation, 1997., IEEE International Conference on*, pages 71–76. IEEE, 1997.

[99] Robert G. Reynolds and Bin Peng. Cultural algorithms: Modeling of how cultures learn to solve problems. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 166–172. IEEE Computer Society, Washington, DC, 2004.

[100] Alessandro Ricci, Mirko Viroli, and Andrea Omicini. CArtAgO: An infrastructure for engineering computational environments in mas. In *Proceedings of the 3rd International Workshop on Environments for Multi-Agent Systems*, pages 102–119, 2006.

[101] Allessandro Ricci, Michele Piunti, and Mirko Viroli. Environment programming in multi-agent systems: an artifact-based perspective. *Autonomous Agents and Multi-Agent Systems*, 23:158–192, 2011.

[102] Nicole Ronald, Virginia Dignum, Catholijn Jonker, Theo Arentze, and Harry Timmermans. On the engineering of agent-based simulations of social activities with social networks. *Information and Software Technology*, 54(6):625–638, 2012.

[103] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 2010.

[104] Dirk Schlierkamp-Voosen and H Mühlenbein. Predictive models for the breeder genetic algorithm. *Evolutionary*, 1(1):25–49, 1993.

[105] Julien Siebert, Laurent Ciarletta, and Vincent Chevrier. Agents & artefacts for multiple models coordination: objective and decentralized coordination of simulators. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 2024–2028, 2010.

[106] Julien Siebert, Laurent Ciarletta, and Vincent Chevrier. Agents and artefacts for multiple models co-evolution: Building complex system simulation as a set of interacting models. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 509–516, 2010.

[107] Matthew Denman Smith and Courtney J Conway. Collection of mammal manure and other debris by nesting burrowing owls. *Journal of Raptor Research*, 45(3):220–228, 2011.

[108] Anne Snowdon, Abdul Hussein, Carol Kolga, Beth Bruce, Pau Boase, Rebecca Purc-Stevenson, and Andrew Howard. Are we there yet? Canada's progress

towards achieving road safety vision 2010 for children travelling in vehicles. *International Journal of Injury Control and Safety Promotion*, 16(4):231–237, 2009.

[109] Canadian Paediatric Society. *Are we doing enough? A status report on Canadian public policy and child and youth health.* Canadian Paediatric Society, Ottawa, ON, 2012.

[110] Alexander Stoytchev. Behavior-grounded representation of tool affordances. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3060–3065, 2005.

[111] Carlos M. Toledo, Rafael H. Bordini, Omar Chiotti, and María R Galli. Developing a knowledge management multi-agent system using JaCaMo. In *Programming Multi-Agent Systems*, pages 41–57. Springer, 2012.

[112] Michael Tomasello. *The cultural origins of human cognition.* Harvard University Press, 2009.

[113] Michael Tomasello, Ann C. Kruger, and Hilary H. Ratner. Cultural learning. *Behavioral and Brain Sciences*, 16:495–552, 1993.

[114] Ministry of Transportation Transport Canada. Canada motor vehicle traffic collision statistics 2011, 2011.

[115] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards.* PhD thesis, University of Cambridge, England, 1989.

[116] Christopher John Cornish Hellaby Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[117] Alexander B. Wood, Thomas E. Horton, and Robert St. Amant. Effective tool use in a habile agent. In Ellen J. Bass, editor, *Proceedings of the IEEE Systems and Information Engineering Design Symposium*, pages 75–81. IEEE, 2005.

[118] Michael J. Wooldridge. *Reasoning about rational agents.* Intelligent Robots and Autonomous Agents. The MIT Press, Cambridge, MA, 2000.

[119] Yan Wu and Yiannis Demiris. Learning dynamical representations of tools for tool-use recognition. In *IEEE International Conference on Robotics and Biometics*. IEEE, 2011.

[120] Shinya Yamamoto, Tatyana Humle, and Masayuki Tanaka. Basis for cumulative cultural evolution in chimpanzees: social learning of a more efficient tool-use technique. *PLoS One*, 8(1):e55768, 2013.

[121] Yumiko Yamazaki, Chieko Echigo, Masakado Saiki, Masayuki Inada, Shigeru Watanabe, and Atsushi Iriki. Tool-use learning by common marmosets (callithrix jacchus). *Experimental brain research*, 213(1):63–71, 2011.

Appendix A

Copyrights

In the next few pages, I provide permissions granted to reproduce aspects of my published material as chapters within this dissertation. The first refers to the publication in the proceedings of the IEEE Congress on Evolutionary Computation 2011 and the second to the publication in IEEE Symposium on Swarm Intelligence 2014. The third refers to the European Conference on the Synthesis and Simulation of Living Systems 2011. The final copyright permission is for the paper that has been accepted for publication to appear in The International Conference on Autonomous Agents and Multiagent Systems 2015. The permission is noted in the REUSE section from ACM.

Copyright Clearance Center RightsLink®

Home | Create Account | Help | Live Chat

IEEE
Requesting permission to reuse content from an IEEE publication

Title: Evolution of artifact capabilities
Conference Proceedings: Evolutionary Computation (CEC), 2011 IEEE Congress on
Author: Mokom, F.; Kobti, Z.
Publisher: IEEE
Date: 5-8 June 2011

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK | CLOSE WINDOW

Home | Create Account | Help | Live Chat

Title:	Improving artifact selection via agent migration in multi-population cultural algorithms
Conference Proceedings:	Swarm Intelligence (SIS), 2014 IEEE Symposium on
Author:	Mokom, F.; Kobti, Z.
Publisher:	IEEE
Date:	9-12 Dec. 2014

Requesting permission to reuse content from an IEEE publication

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK | CLOSE WINDOW

From: **Pamela L Quick** quik@mit.edu
Subject: Re: Permission to use material from publication for thesis
Date: March 2, 2015 at 11:15 AM
To: Felicitas Mokom mokom@uwindsor.ca

PL

Dear Felicitas Mokom,

Thank you for your message. You are free to reprint your contribution to ECAL 2011 in any way that you wish. Please indicate that the article appeared in "Advances in Artificial Life, ECAL 2011: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems," edited by Tom Lenaerts, Mario Giacobini, Hugues Bersini, Paul Bourgine, Marco Dorigo, and René Doursat, published by The MIT Press.

Please let me know if you have any questions.

Very best,

Pamela Quick
Permissions Manager
The MIT Press
One Rogers Street
Cambridge, MA 02142-1209
USA

quik@mit.edu
617-253-0080

On 2/27/15 7:53 PM, "Felicitas Mokom" <mokom@uwindsor.ca> wrote:

> Dear Mrs. Quick,
>
> I would like to use a paper that I published in Advances in Artificial Life, ECAL 2011 titled ³A Cultural Evolutionary Model for Artifact Capabilities² as a part of my dissertation. I was the primary author on the paper and it was co-authored by my supervisor. From your website, I found information about getting permission for Journals and Books, but not for conferences.
>
> May I have permission to use it? If more information is required, please kindly advise.
>
> sincerely,
>
> Felicitas Mokom

ACM Information for Authors

Author Rights	FAQ

ACM Author Rights

ACM exists to support the needs of the computing community. For over sixty years ACM has developed publications and publication policies to maximize the visibility, impact, and reach of the research it publishes to a global community of researchers, educators, students, and practitioners. ACM has achieved its high impact, high quality, widely-read portfolio of publications with:

- Affordably priced publications
- Liberal Author rights policies
- Wide-spread, perpetual access to ACM publications via a leading-edge technology platform
- Sustainability of the good work of ACM that benefits the profession

CHOOSE

Authors have the option to choose the level of rights management they prefer. ACM offers three different options for authors to manage the publication rights to their work.

- Authors who want ACM to manage the rights and permissions associated with their work, which includes defending against improper use by third parties, can use ACM's traditional copyright transfer agreement.
- Authors who prefer to retain copyright of their work can sign an exclusive licensing agreement, which gives ACM the right but not the obligation to defend the work against improper use by third parties.
- Authors who wish to retain all rights to their work can choose ACM's author-pays option, which allows for perpetual open access through the ACM Digital Library. Authors choosing the author-pays option can give ACM non-exclusive permission to publish, sign ACM's exclusive licensing agreement or sign ACM's traditional copyright transfer agreement.

POST

Authors can post the accepted, peer-reviewed version prepared by the author—known as the "pre-print"—to the following sites, with a DOI pointer to the Definitive Version in the ACM Digital Library.

- On Author's own Home Page *and*
- On Author's Institutional Repository *and*
- In any repository legally mandated by the agency funding the research on which the work is based.
- Prior to submission to ACM for peer-review, authors may post their original work in any informal, non-peer-reviewed aggregation or collection.

DISTRIBUTE

Authors can post an *Author-Izer* link enabling free downloads of the Definitive Version of the work permanently maintained in the ACM Digital Library

- On the Author's own Home Page *or*
- In the Author's Institutional Repository.

REUSE

Authors can reuse any portion of their own work in a new work of *their own* (and no fee is expected) as long as a citation and DOI pointer to the Version of Record in the ACM Digital Library are included.

- Contributing complete papers to any edited collection of reprints for which the author is *not* the editor, requires permission and usually a republication fee.

Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected).

- Commercially produced course-packs that are *sold* to students require permission and possibly a fee.

CREATE

ACM's copyright and publishing license include the right to make Derivative Works or new versions. For example, translations are "Derivative Works." By copyright or license, ACM may have its publications translated. However, ACM Authors continue to hold perpetual rights to revise their own works without seeking permission from ACM.

- If the revision is minor, i.e., less than 25% of new substantive material, then the work should still have ACM's publishing notice, DOI pointer to the Definitive Version, and be labeled a "Minor Revision of"
- If the revision is major, i.e., 25% or more of new substantive material, then ACM considers this a new work in which the author retains full copyright ownership (despite ACM's copyright or license in the original published article) and the author need only cite the work from which this new one is derived.

Minor Revisions and Updates to works already published in the ACM Digital Library are welcomed with the approval of the appropriate Editor-in-Chief or Program Chair.

RETAIN

Authors retain all *perpetual rights* laid out in the ACM Author Rights and Publishing Policy, including, but not limited to:

- Sole ownership and control of third-party permissions to use for artistic images intended for exploitation in other contexts
- All patent and moral rights
- Ownership and control of third-party permissions to use of software published by ACM

Have more questions? Check out the FAQ.

back to top

Vita Auctoris

NAME:	Felicitas Anyicha Mokom
PLACE OF BIRTH:	Tiko, Cameroon
YEAR OF BIRTH:	1971
EDUCATION	University of Texas, Austin, Texas USA B.A., Computer Science 1990
	Wayne State University, Detroit, Michigan USA M.Sc., Computer Science 1993
	University of Windsor, Windsor, ON, Canada Ph.D., Computer Science 2015