Aspects of Self-awareness: An Anatomy of Metacreative Systems

Simo Linkola, Anna Kantosalo, Tomi Männistö, Hannu Toivonen
Department of Computer Science
University of Helsinki
{simo.linkola, anna.kantosalo, tomi.mannisto, hannu.toivonen}@helsinki.fi

Abstract

We formulate a model of computational metacreativity. The model consists of various aspects of creative self-awareness that potentially contribute, in various combinations, to the metacreative capabilities of a creative system. Our model is inspired by a psychological view of metacreativity promoting the awareness of one's thoughts during the creative process, and draws from the field of self-adaptive software systems to explicate different viewpoints of metacreativity in creative systems. The model is designed to help in analyzing metacreative capabilities of creative systems, and to guide the development of creative systems in a more autonomous and adaptive direction.

Introduction

Metacreativity, the capability to reflect on one's own creative processes and to adjust them, is an essential part of any creative system that could be claimed to have creative autonomy or intrinsic motivation. For instance, Jennings (2010) argues that autonomous change, the capability of a system to modify its standards by its own decision, is a requirement for creative autonomy. Metacreativity and creative autonomy allow the system to evolve and, eventually, to create artifacts outside the control of the programmer or other agents.

Metacreativity is the subject of various discussions in the field of computational creativity (see, e.g. Buchanan (2001), Colton (2009), Grace and Maher (2015), Jennings (2010) to name a few). Despite a general aspiration towards building systems with greater creative autonomy and a general interest in metacreativity, explicit models of what metacreativity is and how it can be achieved have been scarce. Our goal is to add to the understanding of computational metacreativity by proposing concepts, components and processes useful in characterizing and also building creative systems.

The ability of a computational system to modify itself is known in software architecture research as self-adaptivity (Salehie and Tahvildari 2009; Lewis et al. 2015). We draw from this field of research to derive concepts for metacreative systems.

A key concept for self-adaptivity and metacreativity, both in humans and machines, is *self-awareness*, the ability to be the target of one's own attention. For a system to be self-aware of some aspect of itself requires that the system explicitly knows of that particular aspect, is able to monitor it and to control itself in relation to it. We formulate these complementary sub-aspects of self-awareness as *(self-)reflection* and *(self-)control*, loosely following concepts used in autonomous computing (Kephart and Chess 2003). The goal is that reflection and control allow a (creative) system to make justified decisions about its own behavior.

Our emphasis on explicit self-awareness implies that the view of metacreativity is "top-down": metacreative control is located in specific components of the system that have awareness of other parts of the system. The advantage of this modular approach is that it is easier to explicate self-awareness and to discuss how and where a system is self-aware, and the concepts directly suggest possible components and their relations for a creative system. The downside is that the model is not well-suited for systems where creativity is an emergent "bottom-up" property arising from the interaction of multiple (simple) agents or other actors and where there is no explicit self-awareness.

The contributions and structure of this paper are as follows. We start by briefly reviewing the background of metacreativity and self-awareness in computational creativity, psychology and software systems. We then move on to our main contribution: a model for metacreativity, consisting of six different types of creative self-awareness in systems that aim to produce creative artifacts. These aspects of self-awareness can be used to describe existing metacreative systems or to design new ones. After introducing the model, we illustrate how different interesting designs of creative systems can be derived from it. We conclude the paper with a discussion of the model and its relationship to some existing concepts related to metacreativity.

Background

We provide here a brief background on three topics relevant to self-aware metacreative systems: metacreativity as discussed within the computational creativity community, self-awareness in psychology, and self-adaptive software system design.

Metacreativity in Computational Creativity

In the computational creativity literature, the term metacreativity has been used in two kinds of contexts that give it largely opposite meanings. The first meaning refers to the programmer of the system as a metacreator and the created system as a metacreation, to use Veale's (2015) terminology. This view is common especially in the musical metacreation literature (see e.g. Pasquier et al. (2017)). The second meaning of metacreativity refers to a process by which creative systems autonomously evolve their creative capabilities. For example, Ventura (2016) characterises metacreativity as the capability of a system to "change its domain knowledge/summative criteria through learning, interaction, environmental effects". In this paper we talk about metacreative systems, i.e., our use of the term 'metacreation' coincides with the latter meaning.

The few attempts to explicitly model computational metacreativity that we have encountered are based on the idea of transformational creativity. Transformational creativity was defined by Boden (1992) and more formally described in a search-based model by Wiggins (2006). In Wiggins' model, creativity is viewed as search in a space of possibly creative concepts. The search is governed by rules which effectively define a system's search space, the method of traversing it, and a function for evaluating concepts. A system is transformationally creative if it changes any of the rules that govern this search. Grace and Maher (2015) elaborate on Wiggins' model by adding features for modeling an agent's expectations and curiosity during the creative search, and by using this model to evaluate potential rules to direct the transformational search.

Implemented metacreative systems are usually based on the transformational creativity paradigm, by invention of new rules for defining, traversing or evaluating search spaces. For instance, Colton's (2001) HR system, originally designed for mathematical discovery, was extended to discover meta-theories about its own theory formation. Colton argues that these meta-theories could be used to extend HR's creative capabilities by implementing them as new rules for the original program. Similarly, in a more recent attempt Morris, Burton, and Ventura (2012) discuss a culinary system that could achieve meta-level capabilities by transforming its evaluation function over time. This type of computational metacreativity is often implied in works describing transformation of creative search, such as Liapis et al. (2013).

Metacreativity and Self-Awareness in Psychology

Self-awareness is a central concept to our model of metacreativity. Throughout the paper, we use the term as it is defined in psychological literature: self-awareness is the capacity to become the object of one's own attention (Morin 2006). The potential to also change oneself is an important aspect of self-awareness, even if implicit in some discussions. In our model, we make it explicit by talking about self-control in relation to self-reflection.

Bruch (1988) has identified metacreativity as being aware of thoughts and feelings during creative experiences. This relates metacreativity to self-awareness as defined above. Bruch also relates metacreative acts to Sternberg's (1982) nine element model of problem solving, which takes into account metacomponential executive processes allowing a creator to plan, monitor and evaluate its own creative process.

One form of executive monitoring is constant overseeing of the solution forming process and being aware of new problems – and solutions – which arise during the creative act, e.g., to recognize possibly serendipitous incidents. This kind of monitoring comes close to ideas of solution-focused strategy known as "design thinking", where a designer uses synthesis of previous (sub)solutions to find new solutions, or redefines the problem based on the accumulated information in order to find an acceptable solution (Cross 1982).

A metacreative person needs to be self-aware of his/her own creative processes. Self-awareness, more generally, is an eminent functionality of human cognition. For example, Morin (2006) characterizes the contemporary neurocognitive models of consciousness by the perception of self in time, and by complexity of self-representations. Neisser (1997) formulates five levels of self-awareness which describe the development of higher cognitive functions: (1) ecological self consists of perceptual information, (2) interpersonal self describes raw awareness of social interactions, (3) extended self is able to reflect on itself over time with no explicit focus on mental states, (4) private self can process private information, such as thoughts and feelings and (5) self-concept is made of abstract and symbolic representations of oneself, such as role and identity, in the end encompassing also meta-self-awareness. Neisser's self-awareness levels have served as inspiration for self-adaptive software systems, from which we draw results.

Self-Adaptive and Self-Aware Software Systems

Metacreativity, as the capability of a creative system to change its own creative behavior, is closely related to self-adaptivity in software systems. The challenges of making software systems self-adaptive have been primarily addressed in the domain of software systems (Salehie and Tahvildari 2009; Camara et al. 2016). Self-adaptive software systems are usually closed-loop systems with a feedback loop connecting changes in the operating conditions back to the system (Maes 1987; Salehie and Tahvildari 2009). The changes can originate from the software systems itself or its context, e.g., an operating environment. Self-adaptation requires the system to be able to reflect on the changes, which requires suitable structures of a self-representation of the system (Maes 1987).

Self-adaptation of software is often architecturally based on a general control structure such as the MAPE-K model (Kephart and Chess 2003). The model has two main components: a base system that is being managed and a self-adaptive system taking the control. The self-adaptive system has the capability to reflect on the controlling actions executed over the managed system. The reflection is based on a monitoring component (the M in MAPE-K) that gauges the managed system. The input from the monitor feeds the analyzer (A), which conducts the analyzes for the purposes of planning (P) the adaptive action required. An executor (E) then carries out the action in the managed system. The monitor-analyze-plan-execute loop uses a shared knowledge-base (K) to inform the self-adaptation.

Lewis et al. (2015) elaborate on the basic MAPE-K kind of model using inspiration from the above-mentioned Neisser's (1997) five-level model of consciousness. For conceptualizing self-awareness, they separate and define levels of awareness for self-aware software (Lewis et al. 2015): stimulus-awareness, interaction-awareness, time-awareness, goal-awareness and meta-self-awareness. For the purpose of constructing self-aware software systems, they propose an overall architecture with architectural elements explicitly separating the responsibilities regarding self-awareness. The architecture also distinguishes between private and public forms of self-awareness based on whether a change originates and the awareness relates to something within or outside the system itself, respectively. However, their private awareness concerns the hardware of the systems itself and is not purely awareness about the software, whereas the definitions of self-adaptation focusing more clearly on software may speak of an evaluation on how well the software is accomplishing its task, e.g., in terms of performance (Laddaga 1997).

A Self-Awareness Model for Metacreativity

We next define our self-awareness-based model for metacreativity. We start by defining six possible aspects of creative self-awareness in systems that create artifacts. We then show how different types of systems can be constructed using different combinations of these self-awareness aspects.

Overview of the Model

The model is grounded on the concept of self-awareness as the basis of meaningful self-change in a creative system. We define six aspects of self-awareness for creative systems and outline how these aspects contribute to a creative system's ability to reason about (reflect) and make decisions of (control) its own behavior. These types of self-awareness will be elaborated on later in this section.

Artifact-awareness A creative system that is artifact-aware is able to monitor the artifacts it creates and to adjust what kind of artifacts are generated based on the observed information.

Generator-awareness A creative system that is generator-aware is able to monitor its generator's behavior and adjust it based on the observed information, possibly redesigning parts of the generator.

Goal-awareness A goal-aware system can observe how well it reaches its creative goals, and can modify its behavior and the goals themselves if needed.

Interaction-awareness An interaction-aware creative system knows that some of its actions constitute interactions with other agents or its environment, and can decide how to interact with others in order to influence them or obtain influences.

Time-awareness A time-aware creative system is informed of its behavior in time. It can observe historical development and anticipate likely future phenomena, and modify its behavior based on these observations.

Meta-self-awareness A meta-self-aware system can observe its own self-awareness aspects and influence how they are exercised.

These self-awareness aspects have both implicit and explicit mutual relationships, which are shown in Figure 1. Artifact-awareness and generator-awareness are in a direct hierarchy, as the generator creates artifacts (Figure 1, bottom middle part). Goal-awareness is prominently related to artifacts, but it can also be related to any other element in the system depending on its design. Similarly, interaction-awareness can be in relation to any component of the system, but often deals with exchanging artifacts — or information about them — with external actors. Time-awareness is different from the other aspects, it can only be observed indirectly and it can not be controlled. Meta-self-awareness means awareness about any of the above types of self-awareness; it is also used as an all-encompassing concept meaning the system's ability to reflect and control itself and its own self-awareness aspects.

Self-awareness over an aspect is composed of reflection and control:

Reflection Reflecting on an aspect means tapping into it, monitoring it to gain information about it and possibly processing that information e.g. by generalizing it.

Control Controlling an aspect means adjusting or modifying that aspect.

Both reflection and control are needed for self-awareness: one is meaningless for metacreativity without the other one. They both need a *target*, a component that is monitored and controlled, and another component (or a set of components) that is in charge of reflection and control, dubbed here as the *manager* of the target.

Each of the self-awareness aspects has a variety of possible reflection and control types. They range from simple to complex and, accordingly, we informally talk about *weak* and *strong reflection* and *weak* and *strong control*.

Weak reflection refers to severely limited capability to gain information about the target, e.g. through a black box function. Strong reflection requires analysis of the observations and general attention towards how and what is monitored. A strong form of reflection can be "perceiving": what is observed is externalized from the system (Grace and Maher 2015).

Weak control covers limited adjustments, e.g. parameter changes and other actions with low impact. On the other hand, a system that has strong control over a component may redesign it by changing, adding or removing parts of it.

Aspects of Creative Self-Awareness

Next, we elaborate on each of the self-awareness aspects and argue how they are intertwined with a creative system's capability to gain information of and make decisions about itself.

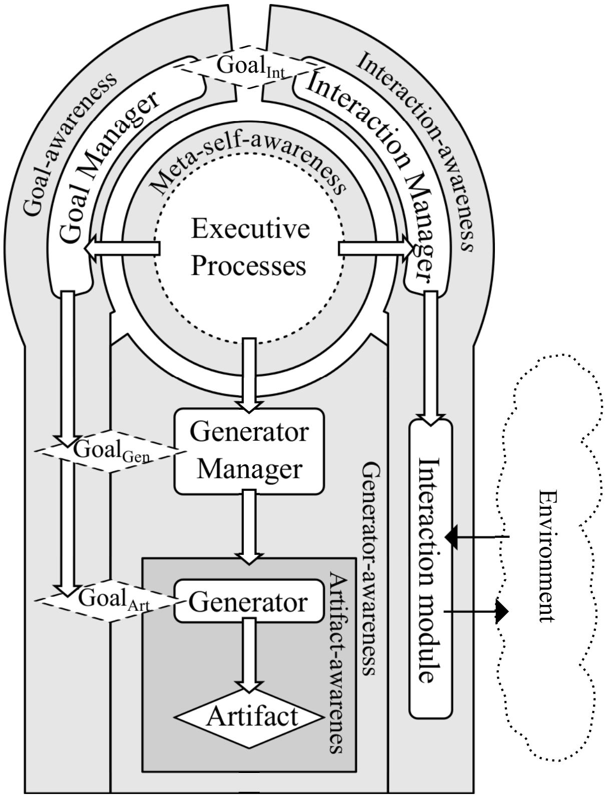

Figure 1: Conceptual drawing of the self-awareness aspects in a creative system and their relations. Reflection and control are indicated by white arrows pointing in the direction of reflection and control. A system can have goals w.r.t. other aspects of awareness and interaction can relate to various other components depending on the system's design. Time-awareness is omitted from the figure, but could be a property of any aspect.

Artifact-awareness A property that distinguishes creativity from mere generation is awareness of the artifacts one generates (Ventura 2016): a system can hardly be called creative if it is not able to assess its own artifacts or adjust what kind of artifacts it generates. An artifact-aware system is not necessarily yet metacreative.

Many creative systems have an evaluation function for artifacts, i.e., they reflect on their products. Some other systems rely on external evaluation; such a system may benefit from building an internal model, e.g. by machine learning, of the external evaluation function. This allows the system then to reflect on artifacts before publishing them, or to modify generation of artifacts to better fit that model.

Consider, for example, a system that generates metaphors, figures of speech where an object (tenor) borrows properties from another object (vehicle). A good metaphor is generally understandable, interesting and tells more about the tenor – or from a different angle – than is usual. If the system is artifact-aware it could, e.g. try to interpret the metaphors it generates using different background knowledge (strong reflection), and modify which object categories (animal, arts, etc.) the vehicle is drawn from during the generation (weak control).

Generator-awareness The next logical step to increase the creative potential of a system is to make the system aware of its artifact generation process. A system that is generator-aware is able to monitor the artifact generation process and to adjust it if needed. This functionality is called Generator Manager (cf. Figure 1). Generator-awareness gives the system transformational capabilities, allowing it to reach a different space of potentially creative artifacts from what it could reach before.

Returning to the metaphor generator used as an example above, a generator-aware version could monitor the combinations of tenor and vehicle produced (weak reflection), and further analyze if the procedure to select the combination performs poorly (strong reflection). It could then adjust how the combination is selected from candidate sets of tenors and vehicles, e.g. by adjusting weights of matching criteria (weak control).

Goal-awareness Creative systems can have various goals (e.g. maximizing value of artifacts or just exploring a space of artifacts), and the goals can consist of conflicting criteria making their use non-trivial (e.g. how to balance novelty and value if they tend to be negatively correlated?). A goal-aware system knows of its own goal(s) and can use that knowledge to solve some of the issues (e.g. deciding how to strike a useful balance), or it can even modify its own goal(s).

At the most concrete level, the goal is defined as a function over created artifacts (Goal$_{Art}$ in Figure 1). The generator could use this function to reflect on the artifacts it has generated, but mere black-box use of an evaluation function does not count as goal-awareness. For a system to be goal-aware, it also needs to be able to change the evaluation standards.

Depending on the system's design, it can have various other goals potentially associated with other aspects of self-awareness (e.g. Goal$_{Gen}$ in Figure 1). An illustrative example requiring interaction-awareness would be a partial goal (Goal$_{Int}$) on pleasing or provoking the user or other agents in the environment. Goal-awareness therefore is actually an umbrella term for many different possible goals.

Goal-awareness allows the system to further loosen its chains from the developer. The system is then not only able to adjust how the artifacts are generated (if it has artifact and generator-awareness), but it also has the ability to change how it perceives artifacts and what it considers as good or interesting artifacts.

In metaphor generation, the system might monitor the interestingness of tenor and vehicle combinations, along other measures, and its relation to the overall evaluation covering a variety of different measures (strong reflection). This reflection could then have an impact on how interestingness affects the overall evaluation (weak control). For example, in a situation where the current formulation of interestingness is seen counterproductive, its effect could be diminished in the overall evaluation.

Interaction-awareness Interaction-awareness allows the system to reason and make informed decisions about its environment and how it behaves with respect to the environment. The system knows that it has components that can be used to communicate with outside sources, either directly (e.g. messaging other agents) or indirectly (e.g. leaving pheromones in the environment). Without interaction-awareness, the system does not have an explicit notion of the outside environment, and cannot comprehend that some of its actions constitute communications with others.

A system with strong interaction-awareness can have a primitive *Theory of mind*, meaning, it can model the world outside, including other creative agents and their properties, such as their assumptions about the original system itself. Importantly, it can distinct these from its own properties.

An interaction- and artifact-aware metaphor generation system could model artifact preferences of other agents, and e.g. observe that a certain agent seems to prefer metaphors where the vehicle is drawn from the animal kingdom. It could then use this information to please the agent by communicating to it only with metaphors which have an animal vehicle.

Time-awareness Time-awareness is distinct from the other self-awareness aspects in several mutually related respects. Time cannot be reflected on its own, nor can it be controlled. Time can only be observed indirectly via changes in artifacts, the system's components or its interactions. Accordingly, time-awareness occurs in conjunction with the other types of awareness and, more specifically, their reflections and controls.

For example, a system can be time-aware with respect to its reflection of artifacts. This allows the system to have an understanding of its generation history and to anticipate its future development. Time-awareness with respect to control of artifacts, in turn, allows the system to make plans for what to generate over time (if the anticipated future was not satisfactory). For example, the system could design a strategy of how to approach a perfect instantiation of a certain type of artifact, e.g, an expressionist painting, over time. It could deliberately allocate its resources so that they maximize the strategy's effectiveness, e.g., it can plan to first explore the artifact space more broadly, and to later on focus in subspaces with more promising artifacts. Further on, the system is able to monitor and adjust its strategies when needed.

In a metaphor-generation system, time-awareness could be utilized to monitor how novelty or interestingness of the artifacts behaves as a function of time, or how other agents appreciate the generated artifacts during the system's lifespan. This information could then be used to form a plan for what kind of artifacts are communicated to each specific agent and how this can be achieved in a timely manner.

Meta-self-awareness Meta-self-awareness is awareness of one's own self-awareness, so it encompasses all other aspects of awareness in a creative system. A meta-self-aware system is able to monitor its own awarenesses and can potentially merge information from different aspects into a unifying view of the system's state.

A metacreative system does not necessary need to have meta-self-awareness, as the system can be metacreative with respect to a single awareness aspect (except artifact-awareness). However, systems without meta-self-awareness have less control over their own behavior.

Consider a meta-self-aware metaphor generation system which has previously generated metaphors with vehicles in the category of animals. Assume that the system observes that it has run out of feasible metaphors with animal vehicles, and that it has previously observed that an agent in the environment prefers metaphors with vehicles from arts. The system could then change its generation of artifacts by fixing the vehicle's category to arts, and change the system's interaction to prefer communication with the art-oriented agent. Further, the system could change its evaluation standards to accommodate the metaphor feedback it receives specifically from this agent.

Connecting Reflection and Control

Self-awareness allows the system to evolve meaningfully during its lifespan. However, it has to have apt connections between reflection and control. We make a distinction between exogenous and endogenous connection types.

An *exogenous connection* is given by an outside source (typically a system's developer), e.g. as a direct mapping between observed and controlled parameters or as a prelearned model. Exogenous connections do not change during the system's lifespan, and they require that the original source of this connection has a substantial understanding of the problem space at design time, something which is not always feasible – or even desirable. Further, exogenous connections implicitly impose reflection and control types to be well known in advance.

An *endogenous connection* is obtained by actively modeling how reflection and control are related, e.g using machine learning or other adaptive models. Unfortunately, due to the complex or even chaotic nature of many creative applications and phenomena, learning relationships between controls and effects can be hard. Deciding right control procedures for certain reflected elements can require a substantial amount of accumulated information from the problem space, some of which can be obtained by experimentation.

Both exogenous and endogenous connection types are perfectly valid in metacreative systems and can co-exist in the same system. The connections can range from simple, e.g. reflecting and controlling a few parameters in the same component, to sophisticated structures connecting reflected elements from various self-awareness aspects to the controls of others. The most eminent connections operate also on meta-self-awareness and execute complex reasoning about how certain reflection should be effected in control.

Example Configurations

We next outline some example configurations of self-awareness aspects and give name suggestions for them.

Creative We call a system creative (as opposed to merely generative) if it is aware of its own artifacts and uses strong reflection and control over them in an exogenously or endogenously connected manner. The behavior of such a system changes over time as it controls new artifacts based on what it already has generated. This does not necessarily imply time-awareness, however.

Self-transforming A self-transforming system modifies its generator in a way that allows it to reach previously unattainable areas of a conceptual space. That is to say, the system is generator- and artifact-aware, and can utilize an exogenous or endogenous connection between them to come up with new generators enhancing the system's capability to explore the creative space.

Self-guiding A system is said to be self-guiding, if it is able to make justified long-term plans and change its way of generating the artifacts if the plan is failing. This requires the system to have generator-awareness and time-awareness. A self-guiding system looks at its own generation process history and estimates how it will be doing in the future if the generator stays unchanged. If the system predicts that the current way of generating artifacts is going to be inadequate in the future, it modifies its generator and starts to make new estimations about its generator's future competence.

A self-guiding metaphor generation system may plan to try to generate as many valuable metaphors as possible with a generator that specifies a set of properties the vehicle must have. If the system anticipates that it is not able to generate any more valuable metaphors with such constraint in its generator, it may change the constraints, e.g. by specifying another set of properties.

Autonomously creative A system is autonomously creative if it changes its goal(s) (Jennings 2010) based on the information it gains from reflecting over the artifacts. Such a system is both artifact-aware and goal-aware. The connection between the artifact reflection and the goal changes can be either exogenous or endogenous. However, an endogenously connected system could be seen to have greater internal motivation.

The metaphor generation system could learn via reflection that vehicles with some properties never score highly in the overall evaluation with current constraints (reflecting on artifacts). It could then directly modify its goal and generator to avoid such vehicles (control of goals, control of generator).

Collaborative A collaborative system interacts with other agents in the environment to advance a common creative goal. It requires interaction-awareness and goal-awareness from the system, as the system has to be able to control its communication and adjust its own goals based on the exchanged messages.

A set of collaborative metaphor generation systems could, for example, divide the metaphor search space so that each agent operates on a distinct subspace. This requires communication-aware coordination of how to set the individual goals of systems.

Self-driven A self-driven system is both self-guiding and autonomously creative. The system makes long-term plans to acquire feasible strategies for future behavior, and changes its goals based on its behavior (e.g. by reflecting on the artifacts or the current generator) and the plan it currently adheres to. To rigorously adapt to new plans and goals, a self-driven system needs to have endogenous connections between reflection and control, requiring meta-self-awareness to connect different self-awareness aspects.

A self-driven metaphor generation system could form a similar plan as in the self-guiding example: fixing a set of properties a vehicle must have and trying to generate as many valuable metaphors as it can. Then, when it anticipates that it cannot generate any more metaphors with current constraints, it may form a new plan with a new set of properties the vehicle must have. At this point, it also can directly modify its goal and generator to avoid vehicles with the former set of properties.

All the above configurations except for "Creative" are metacreative in a meaningful manner. For example, a self-guiding system could be said to have a primitive form of intent as it operates proactively and carries out previously made plans, and an autonomously creative system satisfies the requirements for creative autonomy by Jennings (2010).

However, there still is a gap between these configurations and a system which is in complete control of its own development as a creator, as even a self-driven system is not aware of its own self-awareness and thus cannot change how it is aware of artifacts, the generator, or its own goals. The next level of metacreativity thus involves meta-self-awareness, an executive process that manages other self-awareness aspects. It regulates their operation in order to best fulfill the system's current goals, and allowing it to temporarily – and intentionally – concentrate on specific creative subspaces. We return to this topic in Discussion and Conclusions.

Discussion and Conclusions

We have presented a model for metacreative systems, heavily based on the concept of self-awareness characterized by self-reflection and self-control as its components. In our model, a metacreative process evolves through apt connections between reflection and control, the connections being either exogenous (given, static) or endogenous (achieved or learned by the system). The proposed model is modular in the sense that the six self-awareness aspects introduced can be combined in numerous ways to reach various configurations of metacreative systems.

The proposed model does not cover all possible types of metacreativity, and even within the covered aspects there is a large variety of nuances that could be characterized in more detail. However, we believe that the model contains a representative and diverse set of high-level aspects in which creative systems are potentially metacreative. We believe that the concepts introduced here help computational creativity researchers better analyze and describe how their systems are metacreative.

Designing metacreative systems The model can also be used as a software architecture for metacreative systems. In Figure 1, all boxes with rounded corners are possible components of a metacreative system. For instance, the Generator Manager stands for the functionality that observes and controls the generator. Implementing it as an explicit component (or a set of them) allows, in turn, a meta-self-awareness component to observe and control it.

For the design of metacreative systems, two main lessons can be taken from software architecture design: separation of concerns and means for increasing flexibility.

Separation of concerns is of importance for explicating the responsibilities and functionality related to different aspects of metacreativity and, in particular, clearly defining the levels of metacreativity in software.

On the other hand, to implement metacreativity, the target elements to be modified, e.g., the Generator Manager, need to provide the means for modifiability. A principle mechanism towards modifiability is modularization that allows access to the mechanisms controlling the element. The actual modification mechanisms may include manipulation of the code at runtime, parametrization of computation, dynamic (re)configuration of the functionality, e.g., by means of plug-in components or more sophisticated means of model-based configuration.

MAPE-K Drawing from research in self-adaptive and autonomous systems, our model is inspired by MAPE-K (Kephart and Chess 2003). Reflection in our model can be seen as a mix of monitoring (M) and analysis (A), control as execution (E), and connections between reflection and control as a form of planning (P). However, we talk about reflection and control since they simplify the model, and since they also better support describing situations where reflection and control span multiple self-awareness aspects, possibly including meta-self-awareness capabilities.

Aspects of self-awareness Two of the self-awareness aspects, artifact-awareness and generator-awareness, are specific to and especially interesting for creative systems. A similar separation of artifact and process levels has been previously brought up in computational creativity literature, e.g. by Ritchie (2006), O'Donoghue et al. (2014) and Wiggins (2006). The other four aspects are more general in nature and are inspired by self-awareness levels in self-adaptive systems (Lewis et al. 2015). However, they are formulated here by taking into account specific points of view related to creativity, and have been augmented with a control requirement.

Relation to other models in computational creativity Even though specific self-awareness aspects are not usually presented in computational creativity models, our model can be used in conjunction with existing models of computational creativity. For example, Colton, Charnley, and Pease (2011) argue in their FACE model that creative systems should be able to explain and justify their creations or processes to their audience by generating framing information. The self-aware aspects of our model offer concrete topics and processes for generating such framing information. For instance, an artifact-aware system can explain its artifacts, a generator-aware can further reflect on how it obtained them, a goal-aware system can talk about its goals and can relate the results and process to the goals.

Wiggins (2006) models creativity as search and metacreativity as transformational creativity. Wiggins' meta-level could be seen parallel to our generator-awareness, focused on changing the generation process based on information about the process itself. Our model gives several additional concepts that help analyze and describe metacreativity beyond artifact or process levels. Further on, with our model we can directly point out meta-level elements which allow a system to escape unwanted creative behaviors described by Wiggins (2006). For example, a self-aware system can get over *generative uninspiration* — the system's inability to find valued artifacts or concepts — by intentionally interacting with others to obtain new seed artifacts, by modifying its own goals, or by changing how the generator traverses the creative space.

Serendipity A creative system arrives at a *serendipitous incident* when the system realizes that it has unintentionally done something valuable. With our model, we can describe systems which may take advantage of these situations. For example, a self-driven system is able to recognize them: it is able to assess the value of the incident, and assess if following it pays off better in the long run than the current plan. That is, it does not follow the serendipitous incident blindly, but can form a new action plan from it.

Role of meta-self-awareness In our model, meta-self-awareness is the birthplace for many high level creative phenomena. For example, specific curiosity, characterized by Grace and Maher (2015) as a driving force of intentional transformational behavior, would need elaborated meta-self-awareness. Without meta-self-awareness, a system cannot have a unified view of its state and where it should direct its attention. This may cause the system to behave erratically as the fragmentary curiosity of different system components is rendered contentious.

To give a system the ability to fully be in charge of its own development, the connections in the meta-level executive processes have to be (partially) endogenous. A particularly suitable class of machine learning models for these connections are intrinsic motivation models (see, e.g. Oudeyer and Kaplan (2007)). If appropriately applied, they impose on a system an evolving attention towards its own operation and goals. As such, they inherently enable experimentation and are a good fit to the open-endedness of many creative tasks.

Some intrinsic motivation models may give a system a conation not only to try out new reflection and control connections but also allow it to build up competence towards temporarily fixed goals. For example, these models can be applied to naturally inhibit some sensory stimuli (e.g. specific measurements from artifacts) for a while and concentrate on others (e.g. communication responses of specific agents) and continuously re-evaluate and modify these configurations during the system's lifespan. In this sense, they are promising candidates to offer means for a meta-self-aware creative system to, e.g., exhibit diversive and specific curiosity as discussed by Grace and Maher (2015).

To conclude, we have described six different aspects of self-awareness that creative systems possibly exhibit. As a conceptual tool, these aspects can be used to describe, analyze and compare creative systems. These aspects also directly suggest potential building blocks for metacreative behavior in creative systems.

The next obvious step is to use this model to analyze and describe existing metacreative systems, to test its value as a descriptive and analytical tool. An interesting topic for future work is also designing example systems to concretely illustrate and test some of the self-awareness aspects presented in this paper. Both the process of creating such a system as well as the end result will be important in showing the value of the model as an architecture for metacreative software.

Acknowledgements

This work has been supported by the Academy of Finland under grant 276897 (CLiC).

References

Boden, M. 1992. *The Creative Mind*. London: Abacus.

Bruch, C. B. 1988. Metacreativity: Awareness of thoughts and feelings during creative experiences. *The Journal of Creative Behavior* 22(2):112–122.

Buchanan, B. G. 2001. Creativity at the metalevel: AAAI-2000 presidential address. *AI Magazine* 22(3):13–28.

Camara, J.; Lopes, A.; Garlan, D.; and Schmerl, B. 2016. Adaptation impact and environment models for architecture-based self-adaptive systems. *Science of Computer Programming* 127:50–75.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*, 90–95.

Colton, S. 2001. Experiments in meta-theory formation. In *Proceedings of the AISB01 Symposium on AI and Creativity in Arts and Science*.

Colton, S. 2009. Seven catchy phrases for computational creativity research. In *Computational Creativity: An Interdisciplinary Approach*, number 09291 in Dagstuhl Seminar Proceedings.

Cross, N. 1982. Designerly ways of knowing. *Design Studies* 3(4):221 – 227.

Grace, K., and Maher, M. L. 2015. Specific curiosity as a cause and consequence of transformational creativity. In *Proceedings of the Sixth International Conference on Computational Creativity*, 260–267.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

Kephart, J. O., and Chess, D. M. 2003. The vision of autonomic computing. *Computer* 36(1):41–50.

Laddaga, R. 1997. DARPA broad agency announcement on self-adaptive software (BAA-98-12).

Lewis, P. R.; Chandra, A.; Faniyi, F.; Glette, K.; Chen, T.; Bahsoon, R.; Torresen, J.; and Yao, X. 2015. Architectural aspects of self-aware and self-expressive computing systems: From psychology to engineering. *Computer* 48(8):62–70.

Liapis, A.; Martínez, H. P.; Togelius, J.; and Yannakakis, G. N. 2013. Transforming exploratory creativity with DeLeNox. In *Proceedings of the Fourth International Conference on Computational Creativity*, 56–63.

Maes, P. 1987. Concepts and experiments in computational reflection. In *Conference Proceedings on Object-oriented Programming Systems, Languages and Applications*, OOPSLA '87, 147–155.

Morin, A. 2006. Levels of consciousness and self-awareness: A comparison and integration of various neurocognitive views. *Consciousness and Cognition* 15(2):358 – 371.

Morris, R.; Burton, S.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the Third International Conference on Computational Creativity*, 119–125.

Neisser, U. 1997. The roots of self-knowledge: Perceiving self, it, and thou. *Annals of the New York Academy of Sciences* 818(1):19–33.

O'Donoghue, D. P.; Power, J.; O'Briain, S.; Dong, F.; Mooney, A.; Hurley, D.; Abgaz, Y.; and Markham, C. 2014. Can a computationally creative system create itself? Creative artefacts and creative processes. In *Proceedings of the Fifth International Conference on Computational Creativity*, 146–154.

Oudeyer, P.-Y., and Kaplan, F. 2007. What is intrinsic motivation? A typology of computational approaches. *Frontiers in Neurorobotics* 1(6):1–14.

Pasquier, P.; Eigenfeldt, A.; Bown, O.; and Dubnov, S. 2017. An introduction to musical metacreation. *Computers in Entertainment* 14(2):2:1–2:14.

Ritchie, G. 2006. The transformational creativity hypothesis. *New Generation Computing* 24(3):241–266.

Salehie, M., and Tahvildari, L. 2009. Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems* 4(2):14:1–14:42.

Sternberg, R. J. 1982. Teaching scientific thinking to gifted children. *Roeper Review* 4(4):4–6.

Veale, T. 2015. Computational approaches to language and creativity: Creativity ex machina. In Jones, R., ed., *Routledge Handbook of Language and Creativity*. Routledge. 353–366.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept? In *Proceedings of the Seventh International Conference on Computational Creativity*, 17–24.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.