Modes for Creative Human-Computer Collaboration: Alternating and Task-Divided Co-Creativity

Anna Kantosalo and Hannu Toivonen
Department of Computer Science and Helsinki Institute for Information Technology HIIT
University of Helsinki, Finland
anna.kantosalo@helsinki.fi, hannu.toivonen@cs.helsinki.fi

Abstract

The analysis of human-computer co-creative systems in current literature is focused on a human perspective, highlighting the benefits of co-creative systems for human users. This study paper examines different styles of human-computer co-creation from a more computational perspective, presenting new concepts for analysis of computational agents in human-computer co-creation. Our perspective is based on Wiggins' formalization of creativity as a search. We formalize for co-creative scenarios involving an alternating, iterative approach to co-creation, which we call *alternating* co-creativity and briefly discuss its non-alternating counterpart, *task-divided* co-creativity. With focus on alternating co-creativity, we analyze the co-creative process and discuss new modes and roles for the creative agents within it. Finally, we illustrate our theoretical findings in the context of current co-creative systems and discuss their relation to the roles and expectations presented in current literature.

Introduction

Human-computer co-creativity, a form of collaborative creativity between a human and a computational agent is a topic gaining more and more interest in various domains. Especially interaction designers have been interested in human-computer co-creativity, in order to develop better creativity support systems. In these systems, computational agents are often seen as mere tools (see e.g. Lubart (2005), Maher (2012), McCormack (2008)). As computational creativity researchers we are interested in how the computer can take the role of a more equal partner in the co-creative process.

To be able to facilitate the study of this partnership from a computational perspective, we need concepts and language to discuss the properties of computationally creative agents and frameworks to analyze them further. In this paper, we first look at human-computer co-creativity from a human-centered perspective common in current literature. We see what kind of roles have been commonly taken, or shared, by humans and computational agents and how creative responsibility has been shared in previous projects. We then assume a more computationally oriented perspective, and revisit Wiggins' framework of creativity as a search.

We propose a means for extending Wiggins' framework to human-computer co-creativity that allows for both system and agent level analysis of co-creativity. On the system level, we focus on what we call *alternating co-creativity*, an iterative setting, where a human and a computational agent take turns in constructing and modifying a single creative artifact, or concept. We also briefly consider an alternative scenario, in which the human and the computational agent perform specific creative sub-tasks. We call this *task-divided* co-creativity. On the agent level, we focus on *complete* agents, which themselves form complete systems under Wiggins' formalization and are thus capable of *alternating* co-creativity as opposed to *incomplete* agents, which are only capable of *task-divided* co-creativity.

Our formalization of *alternating* co-creativity focuses on a pairwise case involving only one human and one computational agent, although the setting generalizes to more than two participants. The collaboration typically starts from scratch and the aim of the participants is to create and converge into a result that satisfies both parties. With the framework we analyze a number of potentially challenging scenarios in *alternating* co-creativity to achieve a more balanced human-computer co-creative partnership. Finally, we illustrate the framework in the context of some current co-creative systems, highlighting different modes and roles in *alternating* and *task-divided* co-creativity.

Human-Computer Co-Creation from a Human-Centric Perspective

Current literature on human-computer co-creativity is focused on a human perspective and on how computational agents can support human creativity. This is a noble goal, but often seems to reduce the computational agent into the role of a tool as opposed to an individual creator.

The concept of computational agents as a tool is well illustrated by Lubart's (2005) classification of creative computational partners into four roles:

1. Computer as a Nanny: The computer manages user's work and time spent on creative tasks and takes on routine tasks such as saving and information presentation.
2. Computer as a Pen-Pal: The computer facilitates information flow between the artist and the audience, or other human co-authors.
3. Computer as a Coach: The computer can advice the user

of creativity-inducing techniques to stimulate the user's creative process.

4. Computer as a Colleague: The computer can be creative in itself, "or contribute new ideas in a dialogue with humans".

The same focus on computational agents as tools can also be seen in a more recent article by Maher (2012). She examined the question "Who's Being Creative?" in the context of co-creative ideation and described three roles for the computers: support, enhance and generate. Computers in the support role provide the human with tools and techniques for supporting creativity. The computer as an enhancer extends the creative abilities of the human user by providing knowledge or encouraging creative cognition. Finally, the computer as a generator will provide the user with creative ideas to interpret, evaluate and integrate as creative products.

There is a great overlap between the roles suggested by Maher and Lubart, although the exact equivalence seems to depend on the skill level of the human participant. If we assume a naive human creator, Maher's support role is similar to Lubart's Nanny or Pen-Pal roles. The enhance-role becomes parallel to Lubart's coach role, and the generator role is similar to the role of a colleague.

The two classifications differ most in the role of the human. Maher defines two roles for the human: to model and to generate. The first role describes a human who defines the computational models an processes of the computational agent, while in the second the human is facilitated or enhanced by a computer. Lubart does not explicitly define any roles for the human, but the human is seen as essential for evaluation and fine tuning of creative ideas, while Maher allows also for a more audience-like role, where the human only interprets the final artifact. This allows for an interpretation in which Maher's computational generator can be slightly more independent compared to Lubart's colleague.

Similarly to Lubart, McCormack (2008) implicitly represents the human in the role of a final evaluator in his article. He envisions a future where machines will enable "modes of creative thought and activity currently unattainable" while the human is still an essential part of the creative process. His vision describes creative systems fulfilling the role of an instrument; again the computational agent is seen as an interactive tool with creative potential for the human to master. Burleson (2005) talks about a more balanced relationship between the human and the computational partner, and considers that a hybrid human-computer system may enhance both human and computer capabilities.

Where the strength of the human is seen to lie in evaluation, the strength of the computer is seen in performing mundane tasks fast: Yannakakis et al. (2014) consider that in mixed-initiative game level co-creation, computational agents can improve human creativity by offering lateral thinking aids (fresh stimuli), diagrammatic reasoning aids (pictoral presentations for aiding the creative process) and searching massive search spaces quickly for novel and useful concepts.

Both Maher's and Lubart's classifications also clearly show how a creativity support system does not necessarily need to be creative in order to be able to offer valuable support to the creative process. It is easy to imagine systems fulfilling multiple roles of either classification without any system components designed specifically to contribute creatively. Similarly, McCormack's vision of new instruments and the tasks presented by Yannakakis et al. do not necessarily require autonomous creative capability from the system.

The role of the computer is also defined by the needs of the user of the co-creative system: Lubart (2005) refers to earlier work by Bonnardel and Marmche, who concluded that the user's level of expertise affects what kind of computer support is most helpful for the user. Nakakoji (2006) has similar considerations, as he classifies the role of computational creativity support systems to "dumbbells", "running shoes" and "skis" based on weather the user needs to develop her creative capability, create faster, or if she needs new ways to create that go beyond her own capabilities.

Finally, human-computer co-creativity can take place in multiple configurations: According to Maher (2012), both humans and computers may participate in co-creation as individuals, in groups of humans vs. teams of computational agents, or as a part of the human society vs. a computational multi-agent society. However she notes that most interaction in current systems seems to happen between an individual human and a computer, whereas interactions between societies of humans and agents are nonexistent.

Formalization of Alternating Co-creativity

We focus on cases where one human and one computational agent collaborate in co-creation, as this is currently the most common case presented in literature. We define *alternating co-creativity* as co-creativity in which the co-creative partners take turns in creating a new concept satisfying the requirements of both parties. As a sister term, we define *task-divided co-creativity* as co-creativity in which the co-creative partners take specific roles within the co-creative process, producing new concepts satisfying the requirements of one party. We focus on the first which we consider more interesting as it puts the human and the computational agent in a more equal position.

Under the surface, the goals of the participants in *alternating* co-creativity are much deeper than just generating an artifact. Only in trivial cases will both parties agree from the outset on what is relevant and interesting. Instead, in the interesting cases, to reach an agreement they will need to modify their views and opinions.

For the human participant, this is a chance to get new inspirations and reach artifacts she could not have reached otherwise, potentially expanding her capabilities. For the computational participant, the setting offers both motivation and resources for transformational creativity (see below for more): transformational creativity is needed in order to reach a result that satisfies both the human and the computational agent and input from the user can be used to guide the transformations.

We will build our description and analysis of the two modes, *alternating* and *task-divided*, on Wiggins' (2006) formulation of creativity as search.

The Creative Systems Framework

Wiggins (2006) gives a generic framework for describing creative systems as search; we give a brief overview of the concepts and notation here, with our interpretation, as well as some simplifying notation. For full details, we refer to Wiggins (2006).

A creative system operates in some space $\mathscr{U}$ of concepts or artifacts. For instance, for a poetry writing system, this universe $\mathscr{U}$ could consist of all possible sequences of words. (Wiggins' formulation of the universe can be understood more broadly, but we find it useful that $\mathscr{U}$ specifically denotes the space where the system can technically operate.) Our example poetry system can deal with sequences of words even if they are not poem-like, but it cannot handle melodies or pictures even if they had poetical properties.

A set $\mathscr{R}$ of rules defines the actual search space within the universe by specifying which artifacts are valid in the system's view. A poetical system imposes constraints on the structure and grammaticality of word sequences and the system aims to find sequences that are considered valid poems. An interpretation function $[\![\cdot]\!]$ applies the rules on concepts, yielding real numbers between $[0, 1]$. Assuming a threshold for admitting valid concepts, we denote the valid subspace of $\mathscr{U}$ by

$$R \equiv [\![\mathscr{R}]\!](\mathscr{U}). \tag{1}$$

(Wiggins denotes the same set by $\mathscr{C}$.)

Another set $\mathscr{E}$ of rules evaluates concepts in the universe for their quality or value. In the case of poetry, the quality could be related e.g. to the contents and meaning of the poetry. We denote the subspace of $\mathscr{U}$ evaluated favorably by

$$E \equiv [\![\mathscr{E}]\!](\mathscr{U}). \tag{2}$$

The goal of the system can now be stated simply as creating—or finding, using the search metaphor—concepts in $R \cap E$.

How the system searches the space is defined by a set $\mathscr{T}$ of traversal rules. Another interpretation function $\langle\!\langle\cdot\rangle\!\rangle$ applies the traversal rules $\mathscr{T}$ to move from a point (concept) c^i in $\mathscr{U}$ to a new point c^{i+1}:

$$c^{i+1} \equiv \langle\!\langle \mathscr{R}, \mathscr{T}, \mathscr{E} \rangle\!\rangle(c^i). \tag{3}$$

Since the aim is to satisfy $\mathscr{R}$ and $\mathscr{E}$, the actual traversal is naturally informed by them. In general, the input c^i and output c^{i+1} can be sets of concepts.

To ease discussion of concept sets reached from a particular concept c^i, we use $T^n(c^i)$ to denote the set of concepts reachable in at most n recursive applications of the traversal step of Equation 3:

$$T^n(c^i) \equiv \bigcup_{j=0}^{n} \langle\!\langle \mathscr{R}, \mathscr{T}, \mathscr{E} \rangle\!\rangle^j(c^i). \tag{4}$$

Let $c_\emptyset$ denote the empty concept and assume that it is always a member of $\mathscr{U}$. When a system has no existing concepts to start from, the space reachable to it is now denoted by $T^\infty(c_\emptyset)$. The set of valid and valued concepts that the system can generate can now be expressed simply as $T^\infty(c_\emptyset) \cap R \cap E$.

In our setting, the human and computational agent take turns in modifying a single concept. We use $t(\cdot)$ to denote a single traversal step, taken according to $\mathscr{T}$, $\mathscr{R}$ and $\mathscr{E}$, omitted from the notation for simplicity, and returning a single concept:

$$t(c^i) \equiv \max_{\mathscr{R},\mathscr{E}}(\langle\!\langle \mathscr{R}, \mathscr{T}, \mathscr{E} \rangle\!\rangle(c^i)),$$

where $\max_{\mathscr{R},\mathscr{E}}(\cdot)$ denotes selection of a single item according to how high it evaluates on $\mathscr{R}$ and $\mathscr{E}$.

Transformational creativity (Boden 1992) takes place when the system changes its own conception of which concepts are valid (by modifying $\mathscr{R}$ and thereby R) or its method of traversing the space (by modifying $\mathscr{T}$) (Wiggins 2006), or its standards (by modifying $\mathscr{E}$ and thereby E).

Alternating and Task-Divided Human-Computer Co-Creativity

Interpreted through Wiggins' framework, a creative system consisting of two collaborating parties, a human and a computational agent, aims in principle to create artifacts in the intersection $R \cap E$ of its R and E just like any other system.

We use Wiggins' creative systems framework, however, to describe each agent separately. This allows us to analyze the capabilities and roles of the agents, and to characterize various issues in co-creation. We use subindices h and c to denote the human and computer parts as follows:

$\mathscr{U}_h$	$\mathscr{U}_c$	Sets of all possible concepts that the the human and the computer can process
R_h	R_c	Sets of valid concepts
E_h	E_c	Sets of appreciated concepts
$T_h^n(c)$	$T_c^n(c)$	Sets of concepts reachable in n steps from c
$t_h(c)$	$t_c(c)$	The concept produced after c.

The above sets are defined by the respective rules $\mathscr{R}_h, \mathscr{E}_h, \mathscr{T}_h, \mathscr{R}_c, \mathscr{E}_c, \mathscr{T}_c$. Note also that the traversal (e.g. $T_c^n(c)$ and $t_c(c)$) depends in practice also on the history of already generated/seen concepts, not only the most recent one, c.

Alternating Co-creation In *alternating* co-creation we assume that each party takes turns in co-authoring a single joint concept. Using the above notation, alternating co-creation can be described as cycles of

$$c_c^i = t_c(c_h^{i-1}) \quad \text{and} \quad c_h^{i+1} = t_h(c_c^i) \tag{5}$$

where the subscripts h and c are used to denote the concept creator and superscript i the relative order of the created concepts c.

The universe where both parties can operate is $\mathscr{U}_h \cap \mathscr{U}_c$. The goal of alternating co-creation is to produce concepts that satisfy both parties, thus the valid and appreciated sets of artifacts of the system are also characterized by the respective intersections, $R_h \cap R_c$ and $E_h \cap E_c$, respectively.

In the interesting cases, the goal is not simply to find concepts in the intersection of the mutual initial sets of valid and appreciated concepts $R_h \cap R_c \cap E_h \cap E_c$. For instance, some of the pairwise intersections may be empty. E.g., if

$E_c \cap R_h = \emptyset$ then the computer does not appreciate any concepts considered valid by the human, and the task has no solution. Therefore, the parties will need to be able to transform their rule sets so that they can find a solution that has become acceptable to both, leading to transformational creativity. We will return to this in the later sections.

We define two further modes of alternating co-creativity: *symmetric* and *asymmetric*. When the computational agent encounters a situation where its rules do not allow it to operate further, there are two ways to continue co-creation: transform the rules to adapt to the new situation, or skip turns during the process. If an agent uses transformational creativity to solve conflicts instead of skipping turns, we say it is capable of *symmetric alternating* co-creativity. If it uses skipping instead, we say it is only capable of *asymmetric alternating* co-creativity.

Even in the case where transformations are not needed, alternating co-creation may help either party reach areas they could not have reached otherwise. For instance, if the new concept $c_c = t_c(c)$ returned by the computer was not reachable for the human, i.e., $c_c \notin T_h^n(c)$ for any reasonable n, then the human user gains access to new concepts.

Task-Divided Co-creation In *task-divided* co-creativity, the human and the computational agent do not take equal turns in creating a concept together, but instead the task of creating a concept is divided into specific subtasks within Wiggins' (2006) formulation. These tasks include defining the conceptual space ($\mathscr{R}$), defining the value of the system ($\mathscr{E}$), and generating new concepts within the system ($\mathscr{T}$).

Task-divided co-creativity can be performed by *incomplete* creative agents, restricted systems which are incapable of defining their own concepts (missing $\mathscr{R}$), evaluating their concepts (missing $\mathscr{E}$), or generating concepts (missing $\mathscr{T}$). For instance, some creative systems use genetic algorithms to search a space of possibly interesting concepts, but outsource the evaluation $\mathscr{E}$ (the fitness function) to the user. In contrast, we define a *complete* agent as one which has its own $\mathscr{R}$, $\mathscr{E}$ and $\mathscr{T}$ and is therefore able to take part in alternating co-creation.

Formally *task-divided* co-creativity can be defined as a search performed in $\mathscr{U}$ by an interpretation function utilizing either $\mathscr{R}_h$ or $\mathscr{R}_c$ for defining concepts, $\mathscr{T}_h$ or $\mathscr{T}_c$ for search within $\mathscr{U}$, and $\mathscr{E}_h$ or $\mathscr{E}_c$ for evaluation of the concepts depending on the division of tasks between the human and the computational agent. The interpretation function could then take for example the following form:

$$c^{i+1} = \langle\!\langle \mathscr{R}_c, \mathscr{T}_c, \mathscr{E}_h \rangle\!\rangle (c^i).$$

Obviously, real systems rarely fall into rigid categories such as alternating or task-divided, or complete or incomplete, just like their rule sets $\mathscr{R}$, $\mathscr{E}$ rarely produce crisp sets of valid concepts. We believe, however, that the concepts here and the following analysis of challenges are useful for a better understanding of different possible roles, issues and opportunities in human-computer co-creation.

Computational Challenges in Alternating Co-Creativity

With the formalization of *alternating* co-creativity, we can analyze some potential problem situations encountered when trying to achieve a mutually beneficial, symmetric co-creative session for the human and the computational agent. We focus on four main problems: *universal mismatch*, *conceptual mismatch*, *artistic disagreement* and *generative impotence*. These problems bear similarities to the situations addressed by Wiggins (2006) as aberration and uninspiration, as discussed below.

We define and characterize the problems using the turn-taking structure of alternating co-creation. In particular, we consider different cases where the output of one participant is problematic for the other participant when the latter is supposed to use it as its input. Solutions to the problems are suggested from a position striving to better fill the needs of the human participant, a.k.a the user.

Universal Mismatch

In a *universal mismatch*, the human agent or the computational agent produces a concept that is outside the universe of the agent next in line:

$$c_h^i \notin \mathscr{U}_c \quad \text{or} \quad c_c^j \notin \mathscr{U}_h.$$

Such a situation could happen, for example, in poetry co-creation: If in the computational agent's universe, concepts are ordered lists of words, but the human suggests a visual poem which requires the understanding of the shape of a poem, the agent is fundamentally unable to understand the concept and operate on it.

Unfortunately, a *universal mismatch* is a fundamental problem since, by our definition, the computational agent cannot reach outside its universe $\mathscr{U}_c$.

However, if we allow the computational agent to skip its turn, we may wait until the human proposes another concept that fits within the universe of the agent. This allows for some level of *asymmetric* co-creation without proper alternation between the parties (see above). In the extreme case there is no overlap between the universes $\mathscr{U}_h$ and $\mathscr{U}_c$ (except for the empty concept $c_\emptyset$). In such a case, not even asymmetric co-creation is possible. With this we can formulate a fundamental requirement for alternating co-creation:

$$\mathscr{U}_c \cap \mathscr{U}_h \neq \{c_\emptyset\}.$$

Since there is no way to correct a *universal mismatch* during co-creation, such issues should be tackled already in the design of the co-creative agent.

Conceptual Mismatch

In a *conceptual mismatch* the computational agent is unable to recognize the concept given by the human as a valid concept or *vice versa*:

$$c_h^i \notin R_c \quad \text{or} \quad c_c^j \notin R_h.$$

Compared to a universal mismatch there is still a possibility to represent the relevant characteristics in the universe of the computer. For example, if a computer strictly requires a specific poetic meter but the human has a different meter in mind, the computer can still process the poem (as a sequence of words) but it does not consider it as a proper poem.

The problem could again be solved trivially by having the human continue the creative process alone until we find a concept recognized as valid by the computational agent. If we want to achieve *symmetric* co-creation, we must consider transformational strategies instead.

This problem is somewhat similar to Wiggins' formalization of aberration where the single system comes up with a new concept outside its (current) conceptually valid space. Depending on the value of the concepts, Wiggins proposes some strategies for transformational creativity: If the system has found a new set of concepts which are all valued, we can change rules $\mathscr{R}$ to include the new concepts in the conceptual space. If only some of the concepts found are valued, Wiggins suggests in addition to modify $\mathscr{T}$ to avoid the unvalued concepts. If only unvalued concepts are found, he suggests to modify $\mathscr{T}$ in order to avoid unwanted concepts.

In the case of alternating co-creativity, we can solve the *conceptual mismatch* problem by similar means, by transforming $\mathscr{R}_c$ to include the new concept. Depending on the system, we may also need to make changes to $\mathscr{T}_c$ or $\mathscr{E}_c$, to allow for the search to continue from the new concepts, or to expand the set of valued concepts to cover the new ones (see *Artistic disagreement* and *Generative impotence* below).

If the human participant is unable to understand the computational agent's suggestion, we have two possibilities to adapt to the humans needs: We can transform $\mathscr{R}_c$ to exclude the "wrong" concepts, or modify $\mathscr{T}_c$ to avoid them. This scenario however is difficult to successfully attain for two reasons: The human may be unable to communicate the problem to the computer in a sufficient manner, especially as we assume no other communication means except for the artifact. Also conforming too much to the human's desires may limit the creativity of the computational agent and decrease the overall value of the system for co-creation.

Artistic Disagreement

An *artistic disagreement* takes place when the human and the computational agent disagree on the (aesthetic) value of a concept produced by the other:

$$c_h^i \notin E_c \quad \text{or} \quad c_c^j \notin E_h$$

Artistic disagreement may seem like a trivial problem as the evaluation of the previously produced concept is not computationally necessary for continuing the search. However, from the perspective of co-creation, it is necessary to define this problem, as it may lead to a situation where the system continuously produces concepts that are of no value to the user, or the system is forced to search areas of no artistic interest to itself.

Conceptually, *artistic disagreement* is similar to Wiggins' concept of uninspiration. An uninspired system is unable to find highly evaluated concepts. In "hopeless" uninspiration, we have $E = \emptyset$, in "conceptual" uninspiration we have $E \cap R = \emptyset$ and in "generative" uninspiration we have $E \cap T^{\infty}(c_{\emptyset}) = \emptyset$.

Similarly, an *artistic disagreement* may stem from multiple underlying scenarios:

– The human and the computational agent do not value anything in their shared universe:

$$E_c \cap \mathscr{U}_h \cap \mathscr{U}_c = \emptyset \quad \text{or} \quad E_h \cap \mathscr{U}_h \cap \mathscr{U}_c = \emptyset$$

– The human and the computational agent do not value anything in their shared conceptual space:

$$E_c \cap R_h \cap R_c = \emptyset \quad \text{or} \quad E_h \cap R_h \cap R_c = \emptyset$$

– The human and the computational agent do not value anything the other one can produce:

$$E_c \cap T_h^n(c^i) = \emptyset \quad \text{or} \quad E_h \cap T_c^n(c^i) = \emptyset$$

Wiggins considers that "hopelessly uninspired" and "conceptually uninspired" systems are fundamentally ill-defined. Similarly, we consider that if the human and the computational agent are unable to value anything in each other's universes or conceptual spaces, and they are incapable of transformation, the human and the computational agent are fundamentally unsuited to work together in alternating co-creation. This implies that the computational agent is not designed to fit the user's needs.

In the case of systems capable of transformative creativity we have, however, some options for continuing the creative search in an alternating manner: If the computer does not value any objects in the shared universe we need to change $\mathscr{E}_c$ to better fit the human valuation. If the computer does not value any objects in the shared conceptual space we can either change $\mathscr{E}_c$ as previously, or change $\mathscr{R}_c$ to increase the number of potentially valued concepts in the shared conceptual space. The case for handling specific concepts, unvalued by either the human or the computational agent is more nuanced.

If the computer is unable to value the concept provided by the human, the only option is to again change $\mathscr{E}_c$. However, if the computer produces concepts not valued by the user, we can either again accommodate the user's valuation by modifying $\mathscr{E}_c$, completely forbid search on uninteresting concepts by removing them from R_c by modifying $\mathscr{R}_c$, or direct the search towards more interesting concepts by modifying $\mathscr{T}_c$.

Since evaluation of the offered concept is not required in the formalization, *artistic disagreements* can also be solved by non-transformational means if we allow for the computer to simply trust the user's evaluations. In these situations the computer could simply continue the search despite the evaluative outcomes, but this could imply that the computer gives up, at least partially, its own $\mathscr{E}_c$ and becomes more a servant to the user's goals. The new concepts found in this manner may then be either relevant or irrelevant to the human. Therefore, if the human is similarly trusting the computer, we may soon end up searching areas that are interesting to neither party.

Generative Impotence

Generative impotence occurs if the human or the computational agent is incapable of continuing the creative search from the concept provided by the other:

$$T_c^n(c_h^i) = \emptyset \quad \text{or} \quad T_h^n(c_c^j) = \emptyset.$$

Due to the differences in human and computational creativity, we are much more likely to end up in a situation where the computer is unable to process the current concept.

Trivially the case could be solved either by allowing the computational agent to perform a random search in $\mathscr{U}_c$, or returning to an earlier state, but these solutions seem unfit for a co-creative scenario. Simple random searches are not deemed very creative, and returning to an earlier state may in the worst case lead the human and the computational agent into an endless loop. Again, if we allow asymmetric co-creation, the computational agent can wait until the human produces a new c_h^i which it can process.

In order to enable the co-creation to continue in a symmetric manner, we will need to change $\mathscr{T}_c$ so that the computer is able to continue its search for new concepts. Similarly, if the human is unable to continue creating from a concept provided by the computer, we can either continue the computational creation, or change the search strategy. However, in this case, it would be again extremely important for the human to be able to communicate to the computer in a relevant manner where the problem lies.

Computer Roles in Alternating and Task-Divided Co-Creativity

Formalizing co-creativity as alternating or task-divided search allows us to discuss the role of the human and the computational agent in co-creation from a computational viewpoint. We argue that *alternating* co-creativity poses more strict requirements to the computational agent than *task-divided* co-creativity. To be able to participate in alternating co-creativity, an agent has to be *complete*, whereas also *incomplete* agents can participate in task-divided co-creativity. This section discusses the roles of computational agents in alternating and task-divided co-creativity. We also give practical examples from literature to show how the formalization can be used to analyze existing systems.

Complete Creative Agents in Co-Creation

Computational creative agents, which are *complete* in the sense that they are capable of identifying ($\mathscr{R}_c$), generating ($\mathscr{T}_c$), and evaluating ($\mathscr{E}_c$) some concepts in a space ($\mathscr{U}_c$), can take more advanced roles compared to their *incomplete* counterparts. If they are capable of transformational creativity, i.e., of modifying their own behavior by changing ($\mathscr{R}_c$, $\mathscr{T}_c$, and $\mathscr{E}_c$) based on the human input, we can achieve *symmetric alternating* co-creativity at the system level. Complete agents incapable of transformational creativity can participate in *asymmetric alternating* co-creativity by skipping turns when needed. Naturally, complete agents are also capable of participating in *task-divided* co-creativity, if they suppress some of their capabilities.

Instances of *symmetric alternating* co-creativity are very rare in current literature. Many systems based on *complete* computationally creative agents have been transformed to interactive systems exhibiting creatively unbalanced scenarios: For example, in the Poetry Machine system (Kantosalo et al. 2014) the computational agent works in an environment where it is restricted to provide partial concepts (poetic fragments) only when the human specifically asks for them. On the other hand in the pun generating STANDUP system (Waller et al. 2009), the computational agent seems to be performing the whole creative act alone, based on some minimal human input, such as a word to be included in the pun. These systems are good examples of originally *complete* creative agents participating in *task-divided* co-creation, where the creative responsibility is unevenly distributed to the human and the computational agent.

Among systems described in literature, the game level design system Tanagra (Smith, Whitehead, and Mateas 2010) seems to fit the definition of an *alternating* co-creation system best: In Tanagra, the computational agent and the user take turns in working on the same game level. In addition to generation, Tanagra also participates in evaluating the playability of both human and computationally produced content throughout the creative session.

Pleasing and Provoking Agents The nature of alternating co-creativity and the role of the *complete* creative agent are largely dependent on how it chooses to react to human input. In *symmetric alternating* co-creation, the interaction is defined by how much the computational agent decides to adapt to the user's needs. The agent can either try to *please* the human, by conforming to the human's ideas about concepts and their evaluation or *provoke* the human, by being more willing to challenge the human-provided concepts.

An extreme case of an agent striving to *please* would modify its creative process to better comply with the human's needs and preferences, even to the extent where it effectively reduces its own creativity by limiting $\mathscr{R}_c$ or $\mathscr{T}_c$, or adjusting $\mathscr{E}_c$ to avoid concepts that seem to be unpleasing for the human. Current co-creative systems mainly employ pleasing agents. For example in Tanagra, the user's modifications are given priority over the computational agent's modifications so that the system can not change level components placed by the human. This effectively reduces the search space of Tanagra to accommodate the human.

Provoking computational agents can be though of having stronger opinions, defending their viewpoints and resisting changes based on human preferences. This may make the agent outright challenging towards the human user's suggestions. Unfortunately, such systems are so far nonexistent, and in fact such a stance seems to be opposed by literature. For example, the creators of Tanagra talk about ensuring "that Tanagra does not push its own agenda on the designer" (Smith, Whitehead, and Mateas 2010).

Both *pleasing* and *provoking* agents have use-cases within co-creative systems. For example, if a user is attempting to produce concepts that convey his or her specific style, a *pleasing* agent which adapts to the user's preferences is more desirable. However, if a user is searching for more varied ideas, a *provoking* agent is a more ideal creative partner.

Naturally, agents do not have to be just *pleasing* or *provoking*, but a more balanced position between these two extreme stances is recommended. An agent balancing between the two extremes would conform to the user's preferences whenever it would deem the transformation necessary and mutually beneficial. Therefore the agent should not outright accept or refuse transformational changes introduced by the human suggesting a new concept, but evaluate how valuable it would be to add new acceptable concepts, techniques, or value functions to its library. This manner of intentional, human-induced transformational creativity would potentially allow the computational agent to take more creative responsibility and be a better creative partner.

Incomplete Creative Agents in Co-Creation

Task-divided co-creation is unbalanced by nature, so it can take place between an *incomplete* as well as a *complete* creative agent and a human. So far, most examples of co-creative systems seem to be instances of task-divided co-creation, where the computational agent and the human clearly divide the creative responsibility over a concept to distinct subtasks, including generation and evaluation of concepts, and even the definition of the conceptual space.

The conceptual space where an agent operates is usually defined by the author of the program, but here we are more interested in how the human user participating in co-creation can effectively partake in defining the conceptual space in which the program does its generative and evaluative acts. In some systems, such as the pun generating STANDUP-system by Waller et al. (2009), the user can effectively set the conceptual space by controlling the level of word familiarity and joke class before the computationally driven generation of puns starts. In this case, the computational agent does not have a way to explore the search space beyond the user given constraints, nor does it have a chance to transform the conceptual space where it works. The user therefore acts effectively in the role of a *"concept definer"*.

The strong generative capability of computational agents is often seen as the largest advantage of human-computer co-creation. For example Yannakakis et al. (2014) promote searching massive spaces as an advantage of computational systems in mixed-initiative co-creation. The role of *"concept generator"* is the de facto role of the computational agent in many systems, including especially many systems utilizing genetic algorithms. For example, the Evolver system (DiPaola et al. 2013) is essentially restricted to generating new populations of artwork candidates for the human user to evaluate and select for the next round of generation.

Where generation is often held as the forte of the computational agent, evaluation then again is very much held as the domain of the human author. Both Lubart (2005) and Maher (2012) assume that even systems of the most autonomous sort (computer colleagues or generative agents) will have a human evaluating their creative outputs. Human as the *"concept evaluator"* is clearly seen also in the previously mentioned Evolver project (DiPaola et al. 2013), where the human hand picks the candidates for each evolutionary round. Of course, some systems seem to share the evaluation responsibility, but on distinct topics: For example the Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos 2014) will do evaluations of playability even for user generated content, but ultimately the human decides which concepts are good. In fact, it could be argued that at least the final evaluation of when to end the search for better concepts is in current co-creative systems done by the human.

Discussion and Conclusions

From a computational perspective, the human and computer roles presented in earlier literature do not seem to be precise enough to categorize and describe the responsibilities of the human and the computational agent within the co-creative setting. First, roles of creativity support systems, including Lubart's (2005) computer as a nanny or pen-pal and Maher's (2012) support role, seem irrelevant from an analysis based on the creative systems framework (Wiggins 2006) since the tasks included in these roles (e.g. facilitation of communication between humans) do not count as creative behavior. Second, the computer as a coach (Lubart 2005) or enhancer (Maher 2012) rely in the computer re-formalizing the human's work by introducing specific creativity techniques, or giving the human fresh stimulus to induce creativity—both tasks again may be done by the computational agent without any creative behavior. Finally the final two categories, computer as a colleague (Lubart 2005) and generator (Maher 2012) actually fit a number of computationally very varied scenarios described in this paper. Therefore the introduction of new terms such as *symmetric alternating*, *asymmetric alternating*, and *task-divided* co-creation for describing the creative process, as well as the introduction of two types of computational agents *complete* and *incomplete* should be useful for the analysis of co-creative systems.

With regard to *alternating* co-creation we defined two modes for the computer to take: *pleasing* or *provoking* the human. It seems that whether a system should take the role of a more adaptive or a more challenging colleague depends on the needs and skill level of the user. This has a direct connection to Nakakoji's (2006) work, which underlines the role of co-creative systems as enabling faster creativity, training creativity, or entirely new areas for creativity for the user, and is also supported by the work of Liapis et al. (2013) on designer modeling. Indeed, choosing between a pleasing and a provoking stance will require further work on user modeling. In the future, systems taking a more *provoking* stance may be of particular interest for co-creativity research, as Maher (2012) points out that "successful examples of [human] collective creativity encourage diversity but do not require that everyone understand others' perspectives or even necessarily to reach consensus".

With regard to *task-divided* co-creation we were able to define three distinctive roles which can be taken either by the human or the computational agent: *concept definer*, *concept generator*, and *concept evaluator*. All of these roles can be clearly justified from the point of view of co-creativity as search, as all of them immediately relate to the capabilities in Wiggins' (2006) creative systems framework. The evaluator and generator roles are also implicitly defined in literature. However, in the formal categorizations by Maher and Lubart the systems again have little differences.

In our formalization, we have focused on the responsibilities of the human and the computational agent mostly within an iterative co-creative scenario. However, it is important to note that human influence on the co-creative agent is not limited only to how the computational agent chooses to conform to human needs during a co-creative session, instead the design of co-creative systems is from early on influenced by user needs (Kantosalo et al. 2014), and they can be encoded in such fundamental aspects of the system that limit the universe of concepts the system can work on.

The co-creative session is also characterized by other factors besides the viewpoints and roles presented in this paper. One of the largest factors characterizing co-creation is interaction. We have omitted interaction entirely from this paper, but we want to note that some form of communication besides sharing the concepts could be valuable. Exchanging information such as descriptions of the creative process or evaluations of the concepts shared might provide significant improvements to the co-creative experiences between the human and the computational agent. Certainly, for the computational agent, such information would facilitate making educated decisions on how to carry out the creative transformations required to achieve *symmetric alternating* co-creation.

For possible communication between agents, we can learn from other frameworks of computationally creative agents, such as the FACE model (Colton, Pease, and Charnley 2011) or from how societies of computational agents work together e.g. in the creative workshop model suggested by Corneli et al. (2015). We could also learn from the perspective of social creativity by having the computational agent model the utility value of concepts to the human user, in order to direct the creative search into mutually more beneficial areas. In the future it would also be interesting to consider scenarios involving multiple computational agents and humans.

For now, the framework can be used to analyze current systems to pinpoint computationally interesting areas for research. Likewise, it can be used in the design of new co-creative systems, as it introduces new terminology for discussing both the goals of co-creation as well as the roles and stance taken by the system towards the human during the co-creative process.

Acknowledgments

This work has been supported by the European Commission under the FET grant 611733 (ConCreTe) and by the Academy of Finland under grant 276897 (CLiC). We would like to thank the anonymous reviewers for their constructive comments.

References

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

Burleson, W. 2005. Developing creativity, motivation, and self-actualization with learning systems. *International Journal of Human-Computer Studies* 63(45):436–451.

Colton, S.; Pease, A.; and Charnley, J. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the Second International Conference on Computational Creativity. April 27-29, 2011. Mexico City, Mexico*, 90–95.

Corneli, J.; Jordanous, A.; Shepperd, R.; Llano, M. T.; Misztal, J.; Colton, S.; and Guckelsberger, C. 2015. Computational poetry workshop: Making sense of work in progress. In *Proceedings of the Sixth International Conference on Computational Creativity. June 29-July 2, 2015, Park City, Utah, USA*, 268–274.

DiPaola, S.; McCaig, G.; Carlson, K.; Salevati, S.; and Sorenson, N. 2013. Adaptation of an autonomous creative evolutionary system for real-world design application based on creative cognition. In *Proceedings of the Fourth International Conference on Computational Creativity. June 12-14, 2013, Sydney, Australia*, 40–47.

Kantosalo, A.; Toivanen, J. M.; Xiao, P.; and Toivonen, H. 2014. From isolation to involvement: Adapting machine creativity software to support human-computer co-creation. In *Proceedings of the Fifth International Conference on Computational Creativity. June 10-13, 2014, Ljubljana, Slovenia*, 1–8.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Designer modeling for personalized game content creation tools. In *AIIDE Workshop on Artificial Intelligence & Game Aesthetics*. AAAI.

Lubart, T. 2005. How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies* 63(4):365–369.

Maher, M. L. 2012. Computational and collective creativity: whos being creative? In *Proceedings of the Third International Conference on Computational Creativity. May 30 - June 1, 2012. Dublin, Ireland*, 67–71.

McCormack, J. 2008. Facing the future: Evolutionary possibilities for human-machine creativity. In *The Art of Artificial Evolution*, Natural Computing Series. Springer Berlin Heidelberg. 417–451.

Nakakoji, K. 2006. Meanings of tools, support, and uses for creative design processes. In *International Design Research Symposium '06, Seoul, Korea*, 156–165.

Smith, G.; Whitehead, J.; and Mateas, M. 2010. Tanagra: A mixed-initiative level design tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games. June 19-21, 2010. Monterey, California, USA*, FDG '10, 209–216. New York, NY, USA: ACM.

Waller, A.; Black, R.; O'Mara, D. A.; Pain, H.; Ritchie, G.; and Manurung, R. 2009. Evaluating the STANDUP pun generating software with children with cerebral palsy. *ACM Transactions on Accessible Computing* 1(3):16:1–16:27.

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

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity. In *Proceedings of the 9th Conference on the Foundations of Digital Games. April 3-7. 2014. Ft. Lauderdale, Florida, USA*.