Artificially artistic

Putting the Art back into Artificial Intelligence

Ásgeir Jónasson, Hrafn Jökull Geirsson, Jökull Jóhannsson

Abstract

Creativity is an integral part of humanity and one that defines us as a species. Since research in Artificial Intelligence has always been in part aimed at understanding the human mind and trying to replicate some of its functions, it makes sense to try and understand this quality that defines us and separates us from other species.

Making computers creative is not only a matter of understanding creativity in humans but also how to represent creativity in a computer system, how to make computers learn from past experience in art creation and most importantly, trying to avoid hard coding what constitutes quality art to give the computer the necessary freedom to be creative, all the while allowing it to learn from human artists and perhaps, cooperate with them.

With all authors having dabbled in music creations and other fields of the arts, some more than others, it made sense for us to research this field of Artificial Intelligence. In this paper we take a look at how creativity works in humans and how it might be done in computers, see what has been done in the field of artificial art in the past, brainstorm how we might apply some of the techniques we have learned so far (and others) to creative systems and explore what fields of the arts a computer might master and which ones might even be lucrative.

The age old question is a simple one. "Where do ideas come from?"

GENERAL CREATIVITY

Creativity is not an easy thing to define, and there is no single scientific definition that describes it in a way most people could agree on. Generally it could be associated with the creation of anything which is in any way new, although most people will agree that to be creative, one needs to recognize the created novelty as new (i.e. one needs to be aware that one is being creative). Some people associate creativity with doing something unexpected or something that breaks the rules, but as modern theorist Teresa Amabile once said: "If a machine can do only what it was programmed to do, how can its behavior ever be called creative?"

Creativity has been associated with a wide variety of things, including but not limited to language, humor, problem solving, art, business and cooking.

Margaret Boden famously coined the terms "P-creativity" ("psychological creativity") and "H-creativity" ("Historical creativity"). It should be noted that just because someone else has come up with the same idea at some previous point in time does not mean the entity that comes up with the idea on its own later is not being creative, it is simply not being "H-creative". If an entity is not being "P-creative" it can obviously not be "H-creative".

The degree of the usefulness or enjoyability of the novelty should be taken into account when judging creativity. It should not be enough to come up with something new, the novelty should also have at least some value. Otherwise it is by definition worthless.

In cognitive science creativity is usually thought of as what happens when a new analogy is invented. When your mind connects things together for the first time. People usually get ideas when they are not trying to, but while they do try, they usually fail. and the idea comes at some point in time when they have stopped focusing on it. Our memories are linked in strange, and often times chaotic ways. A smell can bring back a memory of a place, a color can bring back the memory of a dream. Our minds seem to constantly connect together what we sense, creating new connections, new ideas, new experiences.

Humans are easily bored. They find beauty and hold interest only in the kind of art that presents just the right amount of new things to their senses. Generally, a person will enjoy the piece of art, only when it is relatively new, and the person is yet to have fully discovered all of its secrets. As soon as the person understands all the secrets of the piece, the piece becomes predictable and boring. Any artist must bring something new into his or her works of art, or at least something the average appreciator of said art, has yet to familiarize oneself with.

A.I. Creativity

Opinions on what constitutes true machine creativity differ greatly depending on who you ask and this is also true of the authors of this paper. This is one of the first challenges to overcome when constructing a creative machine; how to define when the machine has become truly creative.

One way to construct a creative machine is to tell it about all the necessary requirements for quality art, which of course, depend on the field. The machine could follow these rules and come up with something pleasing to some of our senses, be it hearing, vision or others, but the outcome would be completely predictable, giving the machine little freedom, a necessary component of true creativity.

Then again, neither can random creation of art be thought of as true art, since the randomness would also have to be hard coded. The machine would probably never come up with something pleasing to the senses and even if it did, it might never create something similar or of equal quality again. One way to overcome this is to give the machine feedback on its random creations and hope that it will gather enough information to know what qualifies as quality art. Most forms of art however do not favor complete randomness, and describing to the computer what you like about a picture that is just a random distribution of pixels on a computerized canvas would be tedious or even impossible and it might take the computer a long while to gather enough information from these tests to infer something useful from them. However, there is a method that might allow for random creation of the first attempted artworks but we will get to that later.

Emotions

Some say that no computer can truly be creative without simulating human experience and emotion. We say, a computer cannot know that which we do not tell it.

Some say that a computer cannot tell a good idea from a bad one, but just what is a good idea. A good idea is an idea which makes somebody feel good. How can a computer know what it is that makes us feel if we do not tell it about it?

Since emotions are such a big part of art creation and perception, we would like a system to take in facts about as many aspects of human perception and the different effects they can have, including all these kinds of data: colors, situations, sounds, scents, taste, sights, and touches. The system is then fed thousands of artworks of any type (we will get into specific fields of the arts later on), descriptions of what defines that particular artwork, the colors, what's the situation etc along with the emotions these pieces of art stir inside people. Using inference, the system can gather a knowledge base of emotional effects of all these different types of perceptions.

This poses at least two big problems. How do we represent the emotional data in a form that the computer can understand and analyze? How does the computer go about inferring a new piece of art from what is has already observed? Also, art is always sensitive to context and the atmosphere of a particular environment so for the knowledge base to take into account the perceiver's context and environment would be very useful. Having these components the system might have the following train of thought:

This environment I'm in calls for music the makes people nostalgic >Music that makes people nostalgic is this subset of songs from my database >I notice what these songs have in common >I will make a song which incorporates these qualities.

The main character has no job > this makes the character sad and insecure >I am writing a book that makes people feel safe and happy >people usually feel the same way as the main character > in the end the character must not have no job >the main character must get a job before the end of the story.

There is a search heuristic that can be used to tackle problems similar to those presented above: how to represent a certain domain, such as emotions, and how to create new pieces of art from what the system has already seen. It's called genetic algorithms and we will come back to that at the end of this paper.

A.I. IN VARIOUS FIELDS OF THE ARTS

THE PAST AND PRESENT

Musical composition

Defining music is hard, but it's a beautiful form of expression. Feelings like love, hatred, sadness, happiness can be expressed through a song. As AI became more and more advanced people started wondering whether a machine could actually master musical composition. Some argue that a machine isn't capable of expressing feelings like humans do thus making the music it creates worthless. But some people are convinced machines can indeed make music that is enjoyable. But what would we gain by implementing such a system has it been done before?

The first successful attempt to program an agent that can write music was made by Lejaren Hiller and Leonard Issacson both professors in the University of Illinois at Urbana-Champaign. Their work was done on a supercomputer named ILLIAC and their main result was named the "Illiac Suite" which was a string quartet that was composed with a problem solving approach named "generate and test".

Their solution worked by generating notes randomly by means of Markov chains. Markov chains is a mathematical system that transits from one state to another in a chain like manner. The notes were then tested by means of heuristic compositional rules of classical harmony and counterpoint and the ones that satisfied the rules were kept. If none of the generated notes satisfied the rules a backtracking command was issued and the procedure was started again.

This solution is not perfect because if they rely too heavily on Markovian processes they are not informed enough to make high quality music. The solution seems to have incorporated elements from genetic algorithms, subject we will cover a little later with regard to musical composition.

We know it's possible to make agents that can write music. Those agents would probably not be on the level of someone like Mozart but you could make one that makes good music. Our thoughts on the matter are expressed in the genetic algorithms segment.

Dancing

Systems that have tried to express themselves with the art of dancing have been implemented in the past. One of them is Dance Evolution, which is a program that can make 3d models dance. The program is able to make the models dance specific and varied dances on a range of sounds. The way the program basically works is that it takes in input which is in the form of music. The program then builds dance functions based on the input. Finally it outputs the dance by making the 3d model dance.

We find this project very interesting because of the focus on making virtual dancer dance to songs, rather than teaching them any specific dances. This makes the dancers seem more human and less mechanical.

Painting

Systems that can paint have also been implemented and an example is Harold Cohen's work on AARON which is an agent that can paint pictures. The first goal of this project was to have AARON make some marks that a human observer would believe were done by a human.

By 1985 AARON had rules for the behavior of the outside world, and by describing the statue of liberty in enough detail, AARON was able to draw following picture.

Figure I Statue of liberty

Benefits of implementing these systems

What could we possibly hope to gain from such agents. You could possible make some money by selling the music the musical agent comes up with and the knowledge that would come out of this could be used as a reference in other AI creative projects but most of all, it could be enjoyable to have an agent making music for you.

So even if some people might argue that the music, dances and the picture those agents created are not art, we might ask the question: then what are they?

We think those agents created art, maybe not at the highest level but art all the same.

Story-telling

In the field of AI, storytelling is perhaps somewhat of a misnomer. The goal of storytelling systems is usually not to tell a story, but rather to create storylines. The stories consist mainly of a series of statements about the events which take place, and the vocabulary is usually very limited. So far, storytelling systems have not much concerned themselves with the feelings of the readers. Mostly, effort has been made trying to create a coherent story, fitting into a particular genre or world model, and making sure the characters achieve their goals. An example of a typical computer generated story can be found in appendix A.

Some methods used in story telling include placing a character in a certain situation, giving him or her a goal to achieve through certain actions. A system might also model each character separately, each one making decisions based on the current situation, along with that characters personality traits.

In the field of drama, a new type of interactive fiction has been created. Facade is an interactive story, experienced by the user from the first peron perspective, allowing the user to interact with virtual agents and influence their feelings and behavior through conversation.

Figure II: Facade

The user plays a long time friend of the AI characters Grace and Trip. What starts out as a normal visit quickly turns dramatic as Grace and Trip start fighting and their marriage seems to be falling apart in front of the user's eyes. It is up to the user to decide whose side to take and the story can be replayed and different endings discovered.

Another interesting storytelling system is the text-based interactive storytelling system Say-anything which allows users to start a narrative in any domain they wish. The user and computer take turns writing sentences for a story. The user picks his sentences any way he wants, but the computer searches a collection of millions of personal stories extracted from Internet weblogs in order to generate a coherent response. If the user does not like the response he is given, he may click the computer generated response and be provided with some alternatives he can pick from.

A key aspect of a good storytelling system is its understanding of the emotional effect each situation has on the reader and the story's characters. The latter is important for character development and the structure of the story. A story telling system, like others we mention later on, should therefore have access to a database of artworks, in this case stories, with accurate and concise descriptions of the effects each segment of the story has on the reader, the characters and the overall flow of the story. This feature would require sophisticated language parsing mechanism for the system to be able to identify why these segment have this particular affect.

GAMES

Interactive storytelling approaches are of a great interest to the creators of Interactive games. The story structure of games is usually of a linear nature, and requires that great amounts of time be spent on formulating stories and quests. MIST (Multiplayer Interactive StoryTelling) is a system for interactive storytelling which is focused particularly on multiplayer virtual worlds. In MIST NPCs (non-playable characters) can perform tasks to satisfy their characters motivations, and interact among themselves and a drama manager keeps the story going towards an acceptable conclusion.

Figure III MIST Server

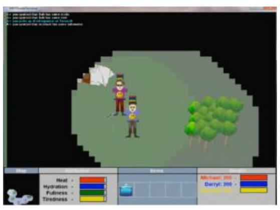

Figure III MIST Client

The idea is to provide human controlled characters with greater freedom in the way they interact with the virtual world than a strictly plot based approach can offer.

Drama: It, I (MIT Media Lab)

A 30 minute computer theater play titled “It/I” [1] featured a live actor and an autonomous, vision controlled computer actor. It was the first play ever produced involving a character automatically controlled by a computer that was truly interactive. The cast consisted of two main characters. The computer was in full control of its character “It”, which taunted the human actor “I”. The human actor could interact with the computer actor through the kinds of actions that could be recognized through automatic image processing, using his body to form shapes. The computer actor could interact with the human actor through images and videos projected onto screens. The scenes were pre-scripted, but allowed for some flexibility, the gestures controlled the flow of the play.

Figure IV Scene from IT, I

For example, in a scene the computer actor “It” displays a countdown timer threatening to set of a bomb, unless the actor perform an action specified by the computer. Each time the actor performs the action, the countdown is restarted. Eventually the human actor must give up, but the amount of times the human actor obeys the computer actor is not prescripted, but is based on the judgement of the human actor during the

1 http://www.youtube.com/watch?v=qUWY0u7BqHY&feature=related

time of the play. After the performance was over, the audience was invited to go on stage and re-enact a scene from the play.

Interactive installations

Interactive installations are pieces of art that respond to the users activity. AI has occasionally been used in interactive installations, such as CrossTalk, which featured multiple agents in an interactive marketplace who would tell anyone who approached the installation about the features of a car, but while no-one was around. they would engage in smalltalk and sales rehearsals among themselves.

Figure V CrossTalk

Another example is SmallTalk 3 in which two computer programs chat with each other about a subject the user chooses. The computers usually end up misunderstanding each other, producing various kinds of chatter, eventually they start repeating themselves, and the conversation is brought to an end.

Other art forms, speculations and Genetic Algorithms

Cinematography and photography

Instead of looking only at the more traditional forms of art artistic A.I. systems have been faced with in the past, we want to speculate about the possibilities in the less obvious applicable fields.

One of the most lucrative fields of the arts is the making of movies, especially in western civilization. We have already mentioned examples of systems implementing the biggest aspects of movie making, the storytelling and acting. Another key aspect is the cinematography. An A.I. cinematographer must possess certain qualities, some of which A.I. researchers have dealt with for decades. First of all, the A.I. cinematographer would need to be equipped with exceptionally sophisticated video tracking, object recognition and motion estimation.

Videotracking would be used to keep track of certain objects, for example the main character in a scene. The Cinematographer also needs to be able to recognize objects, for example all the characters in a script, but also other objects that might be of interest to the audience. The cinematographer must therefore be able to make assumptions about what are the most important objects in each scene, based on the storyline. Motion estimation is also important, for example in fast scenes where the trajectory of a moving object must be known.

Most importantly, the A.I. cinematographer must be able to take into account the context of each scene and the emotions of each character. This poses new problems for the object recognition, not only must it recognize each person but also what emotional state that person is in. And to makes things even harder, it must use this information to decide where to keep the focus of the camera, the angle from which to shoot, and the proportion of the screen the object of interest will occupy and where it is positioned in the frame. This information could only be derived in one way, in our opinion, from a large knowledge base based on many observations of quality human cinematography.

On the positive side, this calls for a modular architecture which makes things a bit easier. Each part of the process can be done in special module which can be developed separately from other ones. For example, the module that takes care of recognizing emotions might use similar strategies as the storyteller does for that particular problem. The modules might also be reusable for a system that takes photographs.

The problem of filming a scene could be abstracted to a search problem where each possible frame is a state. In each frame the amount of emotions captures could be calculated, how well the focus of the camera captures the essentials of the scene and in case of a high speed take, how close the camera is to keeping the moving object in the right place of the frame. These calculations might be too hard to perform in real time, unless the amount of possible frames (a matter of how many pixels each movement in space represents) is reduced significantly. Animated films might therefore make this problem a lot simpler since the cinematography can be defined at any given time and additionally, the system might be told which emotions the characters are feeling at any giving time, based on the script from the storyteller.

Architecture

Architecture is rather different from the fields of art we have covered so far. First of all, architecture sometimes favors function over aesthetics which puts constraints on the creativity.

Whether on macro or micro level, functionality, the technology available, social norms and context all need to be taken into consideration in the design process, along with cost estimation and construction scheduling. Having said that, automating the design could be extremely lucrative.

Just as with cinematography and music, we think it would be optimal to feed a system which designs buildings (or cities/landscapes) with as many images of buildings, cities, landscapes, etc as possible, along with critique on the architecture, both its pros and cons. This poses a problem. What information should the system derive from these images?

First of all, architecture, both the aesthetic and functional part of it, can be represented with geometry and physics. The system would need to be able to calculate the geometry of each object in an image and preferably know which materials were used in its construction. This could be done with supervised learning, perhaps with classification. This system would need to be fed this information with the first (few thousands even) images and then be able to infer from that information facts about other images. Regarding functionality, the system would have to know enough about physics and materials science to know when its design is serving its purpose.

The system should also learn what it costs and how much time it takes to build something in a given context, given certain materials. It would be very nice if the system could be given a picture of the environment in which the building will be built, and know immediately how much time it will take to build there and what it will cost.

Regarding aesthetics, the system would need to know something about what influenced the construction of a given building in its database, e.g. roman architecture, to be able to design something based on the same principals if asked to do so. It should also know what makes a building economical and which aspects of the building the ambiance is a function of.

Now, having all of the features mentioned above, the system would be able to take in parameters such as maximum cost, description of the environment, functionality of the building and the ambiance the buyer is hoping for. Having its huge knowledge base of inferences about designs it should then be able to come up with a design which satisfies each of these constraints.

One might ask if this is true creativity and the answer is probably no. However, how the system designs each building (or landscape) could come as a surprise to the user, since the system would have total freedom in its chose of materials and the geometry, as long as all constraints are met, and that is no different from human architecture, e.g. you cannot build a reversed pyramid, however beautiful it might be.

Now we come back to a point we made early on. This particular system would design buildings using information it has inferred from images it has seen in the past and their accompanying data.

Is it ever different with humans? When we are trying to be creative, aren't we always influenced by things we have seen in the past, even subconsciously? When painting, can we ever truly rid ourselves of the influences made on us by the likes of Van Gogh? When writing music, isn't Mozart always sneaking around somewhere in the shadows of our brain? The only thing we can hope for when creating artistic systems, is to try and give the system as much freedom as possible but we cannot deny it the most fundamental part of any creative process; past experience.

Now we come to a method incorporating past observations and randomness in a special way, using genetic algorithms.

GENETIC ALGORITHMS

The following is only a speculation about how genetic algorithms might be used for art creation and for simplicities sake, let's only look at how a system might go about writing music using genetic algorithms.

Genetic algorithms are based on one of the most fundamental areas of biology; evolution. Instead of being concerned with individuals, like evolution, genetic algorithms work with solutions to specific problems, these solutions are their individuals, for example represented as a string of letters or binary. Also, similar to the way evolution affects a population of a certain species, genetic algorithms deal with a population of candidate solution strings.

The biggest problems facing genetic algorithms creators are how to represent the solution domain and how to evaluate each solution. Regarding evaluation, what we would like the most, is for the musical composition system to have a huge database of songs, along with their critique and as much information about what kind of emotions each song brings out in the listener as possible and also what the key elements are in each songs. Using supervised learning, the system would be fed this information to begin with, but over time, it should be able to infer this information about other songs from their structure. This would make the system sustainable and automatic, making it able to add new songs to its database and evaluate them as they come in. Furthermore, this would be the basis for the evaluation of candidate solutions, in our case, candidate songs.

The problem of finding out which features of a song defines it, is closely related to the problem of representing the solution domain, since these distinctive features must be represented in the solution string, or at least be inferable from it.

A song is, in essence, a collection of sound waves, each of which has many different properties such as amplitude, frequency and wavelength. It would therefore be optimal to view each track of a song, not as a sequence of notes, but as a wave which can be broken down into many little bits for evaluation. With this method, we get rid of such problems as representing different instruments, time signatures, length of notes and whether an instrument is played trough an effect of any kind. It's all implicitly represented in the sound wave.

If each song in the system's database is represented in this manner, the system can infer a lot of interesting information from what it knows about the sound waves of these songs and the information it has about each song, such as emotional effect, influence etc. The system might, for example, find out that songs with rather flat waves are generally calming and that songs with high frequency make people excited. How the waves change throughout the song might also have a lot of impact on the listeners emotional response or how much he or she likes the song. Come to think of it, this system might shed a whole new light on how neurology and sound are linked, but that's out of our scope.

Now getting back to genetic algorithms. We have defined how we can represent a solution string; simply as a wave. The tricky part will be the evaluation function. We have mentioned that we would like the system to have a huge database of songs from which it can infer information about what constitutes great musicianship and the impact of each song on the listener. The problem is finding out, which part of the sound waves are mostly responsibly for the effect the song has and its quality. Unfortunately, we cannot know for sure how this might be implemented, but with a database big enough, along with thorough information about each song, the system should have enough data to make sound inferences, and the bigger, more accurate, and more consistent the database, the better the inferences will be.

We do however have general idea how this might work. The system might for example notice that songs with large chunks of the sound wave consisting of high frequency notes, with frequency radically changing, are more often than not marked as "heavy metal" songs. If the user asks for a song influenced by heavy metal, this might be one of the things the evaluation function looks for. Other easily noticeable traits of heavy metal songs might be soft beginnings (intros) followed by sudden drop in frequency and increase in tempo. Again, something to look for in candidate solutions.

In the end, the evaluation function should try and find out how many of the desirable traits (defined by the user) a candidate solution comes close to having. If the song that's created should have many tracks, each track could be evaluated as a whole, or their composition could be evaluated.

Now we come to one of the problems mentioned early on. How do we allow for random creation of art? That question brings us to three features of genetic algorithms we haven't mentioned so far: population creation, breeding and mutation.

First of all, the population is simply a large set of candidate solution strings. Traditionally they are generated randomly but of course fitting into the specific domain. This fits perfectly with our desire to have a random element in the systems creativity. The initial population of sound waves can be created, hopefully completely, randomly. Now, just like in nature, not everyone is born equal (controversial subject when it comes to humans, but let's not go there) and each candidate solution is evaluated with the evaluation function. Similar to the way things work in nature, the best solutions are most likely to breed, which brings us to the second subject, breeding.

After the generation of the population, breeding starts. One way to do implement this, and it's quite common, is to classify each candidate solution using the outcome of the evaluation function and mostly breed the ones in the highest class. Occasionally though, you might want to bring in to the mix a solution from lower classes so you don't miss out on qualities that may have lost due to much inbreeding (similar to what happens in nature). Songs with a hint of a guitar solo, or a desirable intro, might have been excluded in previous generations because other sound waves with other desirable properties scored higher. Also, you might never breed the solution you are looking for from the original population, if certain features are not found there, so you might want to mutate some of the solutions occasionally. This might simply be reducing the frequency of a certain part of a track or increasing amplitude for a few seconds.

By favoring some solutions in each breeding process, each new generation hopefully brings you closer to a desired outcome, defined by the user's criteria. The population might for example slowly move towards a song that makes you happy or melancholy.

Now we are left with speculations about the most difficult part of genetic algorithm creation, the actual breeding of two solutions. For example, how would the offspring of Hotel California, by the Eagles, and Invaders must die, by Prodigy, sound? That alone would be quite interesting.

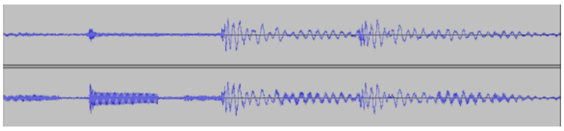

Figure VI Hotel California

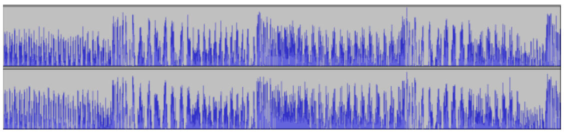

Figure VII Invaders must die

One approach might be to try and keep all the features from both songs that we know are bringing us closer to a good solution and play around a little bit with the rest. We might even cheat a little bit and check whether the offspring gets a high score in the evaluation function but that is not really important because it would not be up for breeding again anyway if it doesn't get a high score since each new generation is evaluated and classified. We might even consider keeping the amplitude of some parts of one parent and the frequency of the other one, at least for some part of a song. This is something that is easy to fiddle around with and would probably be the most interesting part of the implementation. Other mathematical solutions might be averaging all the features between the parents or somehow bringing together the functions that define each part of the parents' waves.

Now to bring all of this together, let's imagine a scenario. A user needs a song that will make the listener melancholy, and incorporates some of the elements of the 70's progressive rock scene. Melancholy music is often rather slow, with soft instruments and the 70's progressive songs were experimental and often the theme changed drastically throughout the song. These might be features that the system has seen in thousands of songs and it knows a lot about the sound waves representing these songs and all that information is used for the evaluation function.

A large population of random sound waves is created and each of them is evaluated. Some of them represent a rather slow song and are picked for breeding, along with others representing sound waves with radical changes throughout the song. After a few generations, features that define songs that match the users criteria are predominate in the population.

It's easy to see that human intervention is at a minimum in the creation of the song. The initial population is created randomly, how the system should go about creating an evaluation function is coded by hand but the parameters the function uses are inferred from thousands of songs. And the best thing is, no one is overlooking the breeding process so the outcome will definitely come as a surprise to the user, a key aspect of every great work of art.

INTEGRATED PROJECT PROPOSAL: A.I. THE MOVIE

To summarize what we have discusses in this paper, we propose a project, integrating all of the different systems we've discussed. Many features of these systems were mutual between all of them, such as inference about the emotional effect of different pieces of art. How these systems might share this information, how they could interact and cooperate might be particularly interesting. The project we propose is a short movie.

The first task of creating a movie is of course the writing of the script, which will be handled by the storytelling system. This system will have to accept some parameters telling it what kind of story to write and something about the characters' goals, but it should then be able to write a story fitting the given criteria, based on stories it has observed in its database.

Since the story telling system figures out the emotional effect each segment of the story has on the characters, and similarly the audience, it should have no problem with passing this information on to the composer, perhaps along with information about the scenario or the year the story takes place. The composer can then start its genetic algorithm with the parameters from the story telling system, hopefully resulting in a melody that captures the emotional effect of the scene.

The environment that each segment takes place in and the emotional effect it should have on the audience can be passed on to the architecture system. In the case of an animated film, which would probably be easiest, the system would not need to be concerned about building cost or scheduling, but that might be possible. Instead it would focus on the emotional affect the landscape or the room the characters find themselves in have on the reader. A gray, barren forest at nightfall is particularly bleak for example.

That leaves the cinematographer. The architecture system, having vast geometrical knowledge, could put the characters in sensible places in the scene, and come to think of it, we would probably need a system like Humanobs to define how the characters position themselves as a group and the gestures they make, since a script will never describe this in detail. With information about the geometrics of the scene, descriptions of the emotions portrayed by the characters throughout the scene and the actions that take places, the cinematographer should be fully capable of shooting the scene with the features described in the "Cinematography and photography" section of this paper.

This interaction between the different systems would be extremely interesting to implement, offering possibilities for modularity and making sure that each system is developed with the greater picture in mind. Having a system capable of performing the tasks described above has great potential in the computer games industry, where the automation of scoring, writing, and cinematography or character interaction can save a lot of money.

Most importantly though, it would be a lot of fun.

References

DREAM-LOGIC, THE INTERNET AND ARTIFICIAL THOUGHT - David Gelernter 2010
http://www.edge.org/3rd_culture/gelernter10.1/gelernter10.1_index.html

Say Anything: A Demonstration of Open Domain Interactive Digital Storytelling
Reid Swanson and Andrew S. Gordon

CrossTalk: An Interactive Installation with Animated Presentation Agents
Thomas Rist, Stephan Baldes, Patrick Gebhard, Michael Kipp, Martin Klesen, Peter Rist, Markus Schmitt

Computational Approaches to Storytelling and Creativity
Pablo Gervás

MIST: An Interactive Storytelling System with Variable Character Behavior
Richard Paul, Darryl Charles, Michael McNeill and David McSherry

Simple Algorithmic Principles of Discovery, Subjective Beauty, Selective Attention, Curiosity & Creativity
Jürgen Schmidhuber

"It/I": A Theater Play Featuring an Autonomous Computer Graphics Character
Claudio S. Pinhanez, Aron F. Bobick

Facade: a one act interactive drama
http://www.interactivestory.net/ Tekið af netinu þann 30.3.2011

http://en.wikipedia.org/wiki/Computational_creativity Tekið af netinu þann 30.3.2011
Greg A. Dubbin and Kenneth O. Stanley, *Learning to Dance through Interactive*

Evolution, Lecture Notes in Computer Science, 2010
http://www.springerlink.com/content/r64078762305823v/

Harold Cohen, *the further exploits of AARON, Painter.* SEHR, volume 4 , issue 2: Constructions of the Mind. July 22, 1995.
http://www.stanford.edu/group/SHR/4-2/text/cohen.html

http://www.mediaartnet.org/works/illiac-suite/

Ramón Lopez De Mantaras, *Making Music with AI: Some examples*
http://www.iiia.csic.es/files/pdfs/1265.pdf

Appendix A

The Vengeful Princess

Once upon a time there was a Lady of the Court named Jennifer. Jennifer loved a knight named Grunfeld. Grunfeld loved Jennifer. Jennifer wanted revenge on a lady of the court named Darlene because she had the berries which she picked in the woods and Jennifer wanted to have the berries. Jennifer wanted to scare Darlene.Jennifer wanted a dragon to move towards Darlene so that Darlene believed it would eat her. Jennifer wanted to appear to be a dragon so that a dragon would move towards Darlene. Jennifer drank a magic potion. Jennifer transformed into a dragon. A dragon moved towards Darlene. A dragon was near Darlene. Grunfeld wanted to impress the king. Grunfled wanted to move towards the woods so that he could fight a dragon. Grunfeld moved towards the woods. Grunfeld was near the woods. Grunfeld fought a dragon. The dragon died. The dragon was Jennifer. Jennifer wanted to live. Jennifer tried to drink a magic potion but failed. Grunfeld was filled with grief. Jennifer was buried in the woods. Grunfeld became a hermit.

MORAL: Deception is a weapon difficult to aim.