IDoESE 2007 Study Plan

Title: Software Engineering Processes Under the Influence of Aesthetics and Art Projects
Author: Salah Uddin Ahmed and Anna Trifonova, Department of Computer and Information Science, Norwegian University of Science and Technology
Contact: salah@idi.ntnu.no, trifonova@idi.ntnu.no, Tel: 0047 73594469

Abstract

The main focus of the study is to investigate the interdisciplinary domain of art and software engineering for the purpose of understanding the different issues, ideas, and concepts that are worth investigating for software engineering discipline in order to leverage technology to the artists as well as enriching the discipline with the experience gathered from art discipline. The research that this study will focus on can be split into: RQ1) How do software engineering and art intersect, i.e., how do software engineering and art can influence and involve each other? RQ2).How we can characterize the development process for software supported artworks and projects? RQ3) How can we benefit our discipline by incorporating the concepts, the knowledge and the experiences that we gain from the art world? RQ4) How can we leverage technology to artists through Software Engineering i.e., providing better tools, processes and roles? To investigate these, we have elaborated the research questions with sub-questions.

1. Introduction

With the rapid development of technology, software is being used in almost every sector of life. As many other fields, art has been influenced by Information Technology. The interconnection between art and computer science has a long history that dates back to the early seventies (Sedelow 70). The intersection of the two fields has interested many artists, researchers and theorists in the recent years. The literature is rife with examples of artists applying mathematics, technology, and most recently, computing (i.e. artificial life, genetic algorithms, artificial intelligence etc.) to the creation of art (Fishwick 06, Wilson 03). Research institutes (e.g. MIT Media Lab[1], Creativity and Cognition studios[2]), art festivals (e.g. Transmediale[3], Read_Me[4], ARS Electronica[5], Make Art[6], FILE[7], Trondheim MatchMaking[8]) and articles appearing in the journals on art, science and technology (e.g. MIT press Leonardo[9], Convergence[10]) or found in ACM digital library or IEEE Xplore and many artists' works and activities that are accessible through web sites (see Dart 07, DeCordova 07, Rhizome 07), blogs and homepages give us the indication of recent interest of

[1] MIT Media Lab http://www.media.mit.edu/

[2] Creativity and Cognition studios http://www.creativityandcognition.com/

[3] Transmediale - Festival for art and digital culture http://www.transmediale.de/

[4] Read_Me - Media art festival with a focus on software art http://readme.runme.org/

[5] ARS Electronica - Festival for Art, Technology and Society http://www.aec.at/

[6] Make Art - Art festival with focus on free and open source software in digital arts http://makeart.goto10.org/

[7] FILE - Electronic Language International Festival http://www.file.org.br/

[8] Trondheim Matchmaking - Annual festival for electronic arts and new technology http://matchmaking.teks.no/

[9] MIT Leonardo - International Journal on Arts, Sciences, and Technology http://www.leonardo.info/

[10] Convergence - International Journal of Research into New Media Technologies http://convergence.beds.ac.uk/

many researchers and theorists in the intersection of software and art. While in the literature the influence of technology including software and information technology on art is clearly visible, the opposite, i.e. the influence of art theories and artistic practices in computing technology is not as obvious (Fishwick 06). We assume that the intersection between software and art is beneficial to both the field of software engineering and art. Our research is an interdisciplinary research in this intersection of software and art.

Though lot of interest is shown towards software from the artists and theorists comparatively little research effort exists from the software engineering discipline. It is acknowledged by the research community that the science community should be proactive to address research involving art (Meyer 98). Even though interdisciplinary research is increasingly used in many other fields, yet software engineers are remarkably reluctant to look outside of own discipline for inspiration and answers both in terms of research and practice (Mehandjiev 06). Glass, Ramesh and Vessey pointed out that only 1.9% of the Software Engineering papers use theories and models from other disciplines (Glass 04). Computer Science papers used other disciplines in 10.77% of the cases, while Information Systems papers used other disciplines in 67.9% of the cases.

Art and software have been in contact in the recent years in numerous ways: such as artists participating in multimedia software projects or in the creation of games and software engineers participating in the creation of interactive art installations or software dependent art projects. With the increased use of technology the field is growing in number, size, aspects and context each year.Through this PhD research we would like to investigate the intersection of art and software engineering and contribute to the knowledge base. The intersection is quite broad and it involves many issues. We have identified several questions for the research, which we assume are too many to be completed during the PhD period. We have found difficulties in focusing on a certain direction at this early stage. Meanwhile it will be also good to mention the tension between our attempt to focus more on software engineering aspects and on the other hand, not to overlook the art aspects. We would like to get feedback on the research questions in terms of suitability for an interdisciplinary research conducted by the software engineering discipline and the feasibility in terms of completion time and complexity.

2. Relevant Prior Works

In his book ‘Information Arts’ Stephen Wilson has a brief description of the intersection of art, science and technology (Wilson 03). He has presented how and where artists and technologists work together and how the artists are using different technologies as media of art. It is a challenge for the software engineers to bring the technology within the reach of the artists. Colin Machin described some of the challenges that software engineers face in an art project (Machin 02). He mentions for example, the technology that serves the artwork may be simple as any other controllers for industrial machines, the software engineers who provide the firmware strives to make the technology more accessible to the artists.

The application of art theories and processes to computing is referred as aesthetic computing. Donald Knuth showed himself to be a strong advocate of aesthetics in programming (Knuth 03). According to Paul Fishwick (Fishwick 06) the idea of aesthetics processing was first explored in 1974 by Frieder Nake. Later, Gelernter (Gelernter 98a; 98b) has provided significant justification for aesthetics in computing. The design of web pages and operating system interfaces are some of fields of computing that use aesthetics. Paul Fishwick

recommends that significantly greater diversity and depth of aesthetics needs to be applied to all other areas of computing, from notations, to formal structures. National Academy study on computing (CSTB 03) recommends considering the effects of the arts within the computing field. Software engineering being a major field of computing also requires the same amount of focus of applying aesthetics in the discipline.

Collaboration is wanted from both artists and technologists in many cases; for example, industries and research institutes want it to nurture innovation and creativity (Harris 99, Wilson 03), artistic software developing companies want it in order to get feedback on their products (D2soft 07) and it is an important issue in the software supported artworks. But the collaboration is not always smooth and some of the challenges appear due to the fact that the two cultures are very different (Meyer 98). Researchers from both software and art domain have worked to smoothen this collaboration. Earlier research in this arena includes case studies of practice based collaborative art research (Salter 05), a "reflective approach" towards practice based research by Legget (Legget 06), collaboration viewed from the educational perspective through interdisciplinary courses such as animation, virtual reality (Ebert 00), (Zimmerman 01) and many more. In (Salter 05), authors have identified issues essential to collaborative practices such as shared language, construction of boundary objects, and accommodation of differing epistemic cultures. Robert Glass mentions how creativity can be incorporated in the software development process (Glass 06). Artist in residency has also been implemented by many other industries and institutes (Wilson 03), but it should be noted that these kinds of programs need a special industrial/research setting that is not always possible.

We have not found many research works so far that address this artist-scientist collaboration from the view point of project management and software methodologies. Linda Candy describes some features of the collaboration and investigates how co-creativity takes place in (Candy 02). Machin presents two models of collaboration in (Machin 02). The only paper that we have found about software process so far is (Marchese 06) where Marchese has shown how agile process was used in the multimedia art installation project Trigger successfully. Adaptive software Development (ASD) was used in that project. It is interesting for software engineering community to find out how this collaboration can be improved and adapted with the software engineering processes without hampering the artistic processes (Machine 02).

As part of this interdisciplinary research, we collaborate with Trondheim Academy of Fine Arts[11], Midgaard Media Lab[12], and Trondheim Electronic Arts Centre (TEKS)[13] and have participated in different art festivals such as Trondheim MatchMaking '06 and Transmediale[3] '07. The main objective is to keep us updated about the latest works and build a network with other researchers and artists. Through this network, we will look for relevant projects that we want to select for conducting case study in our research. It should be mentioned that getting an art project that match all the criteria to be chosen for a case study is not easy as finding an industrial project, as art projects implementation are less in number and they have limitations in terms of software development issues. Following is a short description of the three projects that we have observed as a pre-study in order to gain better understanding for conducting a case study and the future research.

[11] http://www.kit.ntnu.no/

[12] http://www.ntnu.no/midgard/

[13] http://www.teks.no/

1. Flyndre: is a sculpture with an interactive sound system that plays music generated by a software and changes the music dynamically based on the parameters (such as water level, temperature, time, etc.) received from environment through some sensors. The sound generating software was written by the sound composer. It was a single script file, and the composer having no software engineering knowledge later struggled to upgrade, modify the code. Later on Software Engineering students working in the project helped to improve the architecture of the software and made it available as open source for the artist so that he can ensure utilization and improvement of the software by the interested community of developers and users.

2. Sculpture with Interactive 3D sound: is an installation project that aims to decorate a junior high school with an interactive sculpture where people can be able to interact with the sculpture by sending messages, images or sound files from mobile phones, laptops and hand held devices through Bluetooth technology. The received files will be converted into sound and played by the sculpture. For the development of the hardware and the software sound sub-systems a team of students from Hist[14] is created and is working in collaboration with the artist (i.e. the sculpture designer). The members of the project use open source technology (e.g. PureData[15], Apache[16], Python[17]) due to availability of free software that allows development of low cost software, community support for maintenance and upgrade. The developed software will be made accessible as open source through the project web site at the end of the project.

3 .Open Digital Canvas: is a project which aims to embellish a white wall at NTNU with a number of main boards with LEDs on them, creating a big matrix of light pixels.

Observation of the projects has provided us some important issues; we discovered some similarities in the projects' requirements, development methods, etc. such as flexible requirements, low budget, afterward maintenance and upgrading problems. Artists desire to use latest technologies, trend to use open source technologies, attempt to attract interested users and developers to contribute afterwards are also visible from the projects.

3. Research Objectives and questions

As mentioned earlier the objective of this research is to contribute by establishing a knowledge base at the intersection art and software engineering. This knowledge base will later help us develop empirically based theories, models, and tools to support creativity and innovation in software technology development. Given the problem description and objective of the study, we would like to look into the following underlying questions in detail.

1. How do software engineering and art intersect, i.e., how do software engineering and art can influence and involve each other?
   This question can be broken down into the following sub questions:
   a) What are the areas where software engineers and artists involve each other? When, how and where they work together?
   b) What tools and technologies are used in this intersection of two domains?

[14] http://www.hist.no

[15] http://puredata.org/

[16] http://www.apache.org/

[17] http://www.python.org/

c) What are the artists' needs, usage for software?
d) How the collaboration between artists and software engineers takes place? What are the features and characteristics that make it successful?

As seen from the questions above, in this part, we will cover many issues that we have found in the state of the art (i.e. collaboration, methodologies, different art forms etc.) and rather be focused on a broad concept of relationship between software and art to discover as much information as possible in order to create the knowledge base.

2. How we can characterize the development process for software supported artworks and projects?

   This question can be broken down into smaller, more specific questions like:

   a) How the software supported art projects differ from other projects? What are characteristics and features that make this difference?
   b) What are the issues and the challenges that software developers have to tackle to implement that kind of projects?
   c) What are the features and the criteria that make the collaboration between software engineers and artists in an interdisciplinary project successful in a particular context?
   d) Which tool, methodology is more suitable than the others and how and in what context they perform better than the others in addressing the art projects?

In our ongoing literature survey (Trifonova 07), we have observed that art projects differ from other industrial projects. The most significant differences that we have observed are the flexibility of requirements and artists' need for intermediate tools and artists' nature of pushing technologists beyond the boundary.

When the art projects differ from ordinary industrial projects, it is assumable that the issues and challenges that software engineers face in art projects might differ as well. One challenge that software engineers face is to make the software more accessible to the artists. Machin posed some questions and future work such as how we can seek ways out to enable artist's ideas without inhibiting the artistic process (Machin 02). From our observation on the projects (i.e. sculpture with 3D interactive sound and Flyndre), we see that maintenance and upgrade of software is often a big challenge in the art projects. We want to review literature to find out what other challenges exist there. We would also investigate this question while we conduct case study of interdisciplinary projects in the future.

We have found some models of collaboration in the literature. There are also a number of researches that focus on the different features of collaboration. What we want to do here is to collect the different features and characteristics of collaboration and study the examples of some completed projects. We would like to see from the study what features of collaboration made a particular case successful. In other words, what distinguishes the successful collaborations from the less successful ones? Which model of collaboration works better in which context?

In the fourth question we want to study what tools, techniques and processes of software engineering are suitable for a particular art project in a particular context. By tools we mean tools used for the software development, such as software requirements tools, software engineering management tool etc. Methods can be like Heuristic methods, formal methods, or prototyping methods. We would like to investigate which of them are more suitable for art projects.

3. How the software engineering discipline can benefit from utilising experience, theories, or concepts from the art domain?
   To answer this question we need to get understanding of art theories, concepts and processes. Thus here we will use the knowledge gained from results of question 1 where we explore the intersection of art and software from both software engineering and artists' point of view. This question can be further broken into following smaller questions:
   a) What are the areas, tools, artefacts, methodologies in software engineering where aesthetics can be applied?
   b) What are the art theories and or concepts that can be applied to software engineering?
   c) What are the expected results of the application of aesthetics into software engineering? Is it better artefacts? Theories? With better representation and interaction or better processes of developing software?
   d) Can we use theories and experiences from art to nurture creativity and innovation in software engineering?

Question b) is about finding out the concepts of arts and the theories. Many researchers working in the interdisciplinary domain have tried to figure out the art theories that can be applied in science (Fishwick 06). Our objective will be to find out the theories and concepts from the literature and analyze their applicability to different fields of software engineering.

We also want to investigate what outcome we might get by applying the art theories. We would like to look into the other fields of computing where they have already applied aesthetics, art theories, and what results they have achieved. Is it possible that we can get the same results in software engineering too? Planned study to answer this question is based on qualitative analysis of literature. A complementary survey to support our hypothesis, (for example, applying aesthetics would result in better representation of the artefacts) can be conducted in order to justify the relevance of our hypotheses with the opinions of the practitioners and experts. For example, we can conduct survey and interview software engineers.

The fourth sub-question (3 d.) fruits from the results of the earlier questions. There is evidence that creativity and innovation can be nurtured through flexibility of process. Many researchers think that software development is much similar to an artwork. If such, what theories of art can be tied with creativity in software?

4. How can we proffer technology to the artists through software engineering i.e., providing better tools, processes and roles?
   The question can be broken down into following questions:
   a) What are the artists' needs, usages, and requirements for software?
   b) What kind of intermediate tools the artists need?
   c) Why do artists move towards open source software? Or what are the issues and features of OSS which makes it more desirable for artists?
   d) What is the role of open source in art?

The third sub-question is derived from our observation that artists are very interested in open source software. Our attempt will be to address artists' attitudes and needs towards software through which we can understand how software and technology can be made more accessible to the artists.

It will be good to define at this point what we mean by artists as there are people who can be both a programmer and an artist at the same time. Even though some artists might have significant knowledge about technology, but for our research and in general, for large art

works like interactive artworks, sculptures, installation arts, when we say artists we do not expect them to have sound knowledge of technology. This is the scenario that is common in most cases and also this is where the role of software engineers are more pronounced.

4. Empirical Study Design and Arrangements

For the method of our research, we use the Information system research framework developed by Hevner (Hevner 04). We will mainly follow literature review and case study strategies for this research. The data collection methods will be based on interviews, observation, questionnaires and documents (Oates 06). The research in the PhD work will be based on four types of studies.

Study A: *Literature study*

- **Goal:** *To understand the theories and concepts of art and how computing fields have used aesthetics.*
- **Result:** one or more journal article (essay type) to publish the findings from the literature on how software engineering can be applied through the catalysis of aesthetics.
- The study will cover all the questions that come under the research question 3.

Study B: *Systematic review*

- **Goal:** *To understand the intersection of software engineering and art. More specifically to gain an overview of the current shape of collaboration between software engineering and art: when do they involve each other?*
- **Result:** A journal article publishing the systematic review. A conference paper will be also published describing the survey strategy.

For the systematic review we have been searching the following electronic databases: IEEE Xplore, ACM Digital Library, Google Scholar, ScienceDirect – Elsevier, Springer, and Convergence. The systematic review is ongoing and the list of database is being continuously updated. Systematic review is being done according to the principles defined by Kitchenham in (Kitchenham 04).

Study C: *Observation*

- **Goal:** *To understand the different issues related to art projects and the nature of collaboration.*
- **Result:** One or more article to conferences.
- Our observation focus on collecting data on the following issues:
  - artists and software developers understanding of each other and their domains
  - common language, mode of communication and interaction
  - issues related to art projects such as aesthetics, budget, maintenance, upgrade, etc.

We have already started observing one art installation project in Trondheim, Norway. We will observe one or more projects in the future.

Study D: *Case-studies*

- **Goal:** *To collect and analyze data from art projects to better understand the model of collaboration between artists and software developers and study the software development process to analyze how successful it is in terms of addressing different issues of the project and compared to other processes in some other case studies. .*
- **Result:** Two or more conference article, one article describing the collaboration and lessons learned from it, the second one about the software process followed in the project.

- **Context:** the case for the study, which has not been selected yet, will be chosen based on the following context:
  a) Size: medium to big art installation/multimedia installation project. It will have minimum one or two artists with six to ten software developers.
  b) Duration: the project will be at minimum a six months long project.
  c) Software: the art work will be interactive and heavily dependent on software for its functionalities and aesthetics.

It should be noted that there is no big distinction between in study C and D from the design point of view. We mentioned it separately just to state that C is a pre study and is ongoing right now whereas D will be more detailed and focused. It should be mentioned again that study D depends on finding a suitable project defined in the context above. In the following section, we describe how the research questions mentioned in section 4 will be tackled regarding the studies presented above.

RQ1. We will mainly use literature study and systematic review (study A and B) for this part. **RQ2.** Question 2 will be investigated by systematic review, observation and case study, i.e., study B, C, and D. Some case study examples of art projects from the literature will be reviewed. Questions 2c will be mainly based on reviewing literature on collaboration in software art projects. Interviewing software engineers, artists and observing projects are also applicable to collect the features and criteria of collaboration. It should be noted that to get a right project to run case study will present different challenges than in industrial projects. **RQ3.** Question 3 (a, b, c and d) will be investigated mainly through literature study, study A. **RQ4**. Question 4 will be mainly investigated through literature study and systematic review.

Our literature review will finish at the end of October 2007. By that time, we plan to be able to choose one or at most two of our main research questions. The main investigation will start at the beginning of 2008. By that time we will have selected one or more concrete projects to study and one specific research method (qualitative in depth or quantitative in breadth) that suits the project. Depending on the nature of the projects, the number of developers, the nature of the code developed and the research method chosen, we will be able to design data collection and data analysis.

5. Conclusion

The planned research will contribute to the knowledge base of interdisciplinary research between the fields of software engineering and art. One of the main goals of this research is to create the knowledge base which does not exist at the moment. The study has been planned mostly based on literature study and systematic reviews. At the end of the research we expect to have some frameworks or conceptualized view of different issues and themes at the intersection of software and art. As the interdisciplinary domain of software engineering and art is relatively new and there is no established knowledge base, it is understandable that controlled study to validate certain claim or theory is not planned in our research. Rather at the early stage of research we planned to collect important and relevant data by systematic review and case study observation which will later form the basement for theories and axioms. One important threat can be biasness to certain focus while observing projects and literature study. As in the case of case study, we do not have much control to see the effects of confounding factors, and due to the nature of case study it is hard to generalize the result of the study. Qualitative analysis therefore is preferred to analyze the data collected from the studies and careful design and implementation of case study is considered to be followed in every steps.

6. *References*

(Candy 02) Linda Candy and Ernest Edmonds, Modeling co-creativity in art and technology, Proceedings of the 4th conference on Creativity & cognition 2002.

(CSTB 03) Beyond Productivity: Information Technology, Innovation, and Creativity. Washington, DC: National Academies Press, Computer Science and Telecommunications Board.

(D2soft 07) Internet announcement, last accessed on 12th February, 2007, URL: http://www.d2software.com/img/pdf/D2_explore_nuke_fin72806.pdf

(Dart 07) List of artists, digital art and software for artists. Accessed form the link http://dart.fine-art.com/ on 7th February, 2006.

(DeCordova 07) Profile of software artists, collected from the Official site of DeCordova Museum and Sculpture Park, link http://www.decordova.org/Decordova/exhibit/2006/softwareart.html

(Ebert 00) David S. Ebert and Dan Bailey, A collaborative and interdisciplinary computer animation course, ACM SIGGRAPH journal, 2000. Vol-34

(Fishwick 06) Paul Fishwick, editor, Aesthetic Computing, The MIT Press, 2006.

(Gelernter 98a) David Gelernter. Machine Beauty: Elegance and the Heart of Technology. New York: Basic Books, 1998.

(Gelernter 98b) David Gelernter. The Aesthetics of Computing, Detroit: Phoenix Press, 1998.

(Glass 04) R. Glass, V. Ramesh and I. Vessey. An Analysis of Research in Computing Disciplines, Communications of the ACM, 47(6), June 2004.

(Glass 06) Robert L. Glass, Software Creativity, developer.* Books, 2006, **ISBN-13:** 978-0977213313

(Harris 99) Craig Harris, 1999. Art and Innovation, the Xerox PARC Artist-in-Residence Program. MIT press, ISBN-13: 978-0-262-08275-4

(Hevner 04) A. R. Hevner, S. T. March, J. Park, S. Ram, (2004) "Design Science in Information Systems Research", MIS Quarterly Vol. 28, No. 1, pp. 75-105, 2004.

(Kitchenham 04) B. Kitchenham, Procedures for Performing Systematic Reviews. *Keele University Technical Report TR/SE-0401*. ISSN:1353-7776, available online at http://www.elsevier.com/framework_products/promis_misc/inf-systrev.pdf, last visited 13/04/2007.

(Knuth 03) Donald Knuth. Things a Computer Scientist Rarely Talks About, Stanford,CA: CSLI Publications.

(Legget 06) Mike Leggett, Interdisciplinary collaboratio0n and practice based research, The International Journal of research into new media technologies, 2006-12.

(Machin 02) C. H. C. Machin, Digital artworks: bridging the technology Gap, Eurographics UK, 2002.

(Marchese 06) Francis T. Marchese, The Making of Trigger and the Agile Engineering of Artist-Scientist Collaboration, Proceedings of the Information Visualization, 2006.

(Mehandjiev 06) Nikolay Mehandjiev and Pearl Brereton and John Hosking, Second international workshop on interdisciplinary software engineering research (WISER), Proceedings of the 28th international conference on Software engineering, ICSE06.

(Meyer 98) Jon Meyer and L. Staples and S. Minneman and M. Naimark and A. Glassner, Artists and technologists working together (panel), Proceedings of the 11th annual ACM symposium on User interface software and technology, UIST '98.

(Oates 06) Briony J Oats, Researching Information Systems and computing, SAGE Publications Ltd

(Rhizome 07) List of artists and their biography. Accessed from internet on 5th febraury.2007. URL http://rhizome.org/art/by_artist.php

(Salter 05) Christopher L. Salter, Xin Wei, A case study in practice based collaborative Art Research, ACM conference on Creativity and cognition – 2005.

(Sedelow 70) Sedelow, S. Y. 1970. The Computer in the Humanities and Fine Arts. *ACM Computing Survey* 2, 2 (Jun. 1970).

(Trifonova 07) Trifonova, A., Ahmed, S. U., Jaccheri, L. (2007) SArt: Towards Innovation at the intersection of Software engineering and art. Submitted for The *16th International Conference on Information Systems Development*. Galway, Ireland, August 29-31, 2007

(Wilson 03) Stephen Wilson, Information Arts: Intersections of Art, Science, and Technology The MIT Press, 2003, ISBN-13: 978-0262731584.

(Zimmerman 01) Guy W. Zimmerman and Dena E. Eber, When worlds collide! an interdisciplinary course in virtual-reality art, SIGCSE '01: Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education, 2001