

CS CAPSTONE REQUIREMENTS DOCUMENT

DECEMBER 1, 2018

AUTODESK PREDICTIVE ANALYTICS

PREPARED FOR

AUTODESK, INC

ANDREW FAIX

PREPARED BY

GROUP 45

TEAM 45

MCINTYRE SANTA CRUZ

BLAKE HUDSON

SEAN CRAMSEY

Abstract

Autodesk Inventor is commercial software that lends itself to numerous industries. This is due to its ability to digitally model actual products that will eventually be put into production. To accurately reflect what a user of Inventor has created, there exist many options when it comes to command inputs that will hopefully result in a desired outcome. Due to wide array of uses, the user base is incredibly diverse and each user generally has a unique set of tasks that they perform frequently. Although this may be the case, the user interface is standard for all users. This means that a commands location on the screen and hotkeys are the same for every user no matter how frequently they are employed. Autodesk has therefore set out to make using Inventor easier for its existing user base by developing a software add-in that will learn from a specific users inputs and predict what command the user is most likely to use next. This prediction will be presented to the user and will eventually assist in reducing time spent completing a task. (<https://tobi.oetiker.ch/lshort/lshort.pdf>)

CONTENTS

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Glossary	2
1.4	References	3
2	Overall Description	3
2.1	Product Perspective	3
2.2	Product Functions	4
2.3	User Characteristics	4
2.4	Limitations	4
2.5	Assumptions and Dependencies	4
2.6	Apportioning of Requirements	4
3	Specific Requirements	5
3.1	Functional Requirements	5
3.2	Performance Requirements	6
3.3	Design Constraints	7

1 INTRODUCTION

Chapter one delivers what the document contains. Along with definitions and abbreviations for the language used in the document. The topics that will be discussed is the scope of the software developed

1.1 Purpose

The purpose of this document is to define the requirements needed for the development of an Autodesk Inventor Command add-in. Once defined, the requirements will be explored and properly outlined.

1.2 Scope

The Autodesk Inventor Command add-in is an addition to the already existing framework of the Inventor application. The add-in being discussed will gather inputs from a user and will make a visual suggestion of the next command a user will most likely enter. The add-in will be limited to functioning within the sketch mode found within Inventor. Upon completion of the application, users will complete tasks with fewer inputs required, and in less time.

1.3 Glossary

- Add-in - A special type of Inventor program. Add-ins are automatically started whenever Inventor is run. This results in add-ins having the capability to insert itself into Inventors user interface, and to interact and respond to events within Inventor
- AICA - Acronym for Autodesk Inventor Command add-in
- Assembly - In Inventor, assembly is a collection of parts [3].
- Autodesk - An American multinational software corporation that develops software for the architecture, engineering, construction, manufacturing, media, and entertainment industries.
- Autodesk Inventor - Software developed by Autodesk. It is a computer-aided design(CAD) application for 3D mechanical design, simulation, visualization, and documentation.
- Drawing - In Inventor, drawing is referred to as a blueprint of the assembly [3].
- Machine Learning - A method to analyze data that results in a predictive model. This is a form of artificial intelligence that can learn to identify patterns based on available data and to ultimately make decisions based on its own analysis.
- Part - In Inventor, a part is a small 3D or 2D component [3].
- Presentation - In Inventor, a presentation expands an assembly to view each individual part. It is used to see the product with each individual part [3].
- RNN - Acronym for recurrent neural network. It is a class of artificial neural networks. The type of neural network utilizes temporal dynamic behavior for a time sequence. This allows internal states to process sequences of inputs [2].
- Sketch Mode - Feature found within Autodesk Inventor. Allows a user to digitally sketch their product.
- XML - Extensible Markup Language. Primarily used to store and transport data. It is essentially a formatting style that has been standardized and can therefore be deconstructed quickly by the appropriate software.

1.4 References

[1] 3D CAD Software for Product Development. Inventor — Mechanical Design 3D CAD Software — Autodesk, www.autodesk.com/products/inventor/overview.

[2] Recurrent Neural Network. Wikipedia, Wikimedia Foundation, 25 Oct. 2018, en.wikipedia.org/wiki/Recurrent_neural_network.

[3] File Types and Templates in Inventor. Autodesk Support Learning, knowledge.autodesk.com/support/inventor-products/getting-started/caas/CloudHelp/cloudhelp/2014/ENU/Inventor/files/GUID-94B779C0-6B2B-499A-A4F9-2E4BAB49712F-htm.html.

2 OVERALL DESCRIPTION

The purpose of this section is to define how the Autodesk Inventor Command add-in relates to the overall system of Autodesk Inventor. It will also be a piece used to fully explain how the software functions and makes use of foreign libraries in order to accomplish its predictive goal. Finally the software will be placed in context, and possible User interactions will be fully explored.

2.1 Product Perspective

The form this software has taken is that of an Autodesk Inventor add-in. This add-in is run from within Inventor, and essentially becomes an extension of the existing system. The nature of an add-in being run from within Inventor means that any event which occurs from within Inventor can be responded to by an add-in. In the case of the AICA, all commands entered during the use of sketches in Inventor will be recorded into an XML database. The add-in itself will contain a machine learning algorithm that will learn from these stored user inputs. A visual suggestion will appear within Inventor, in the form of a pop-up, regarding the users next command needed that has been generated by a machine learning algorithm.

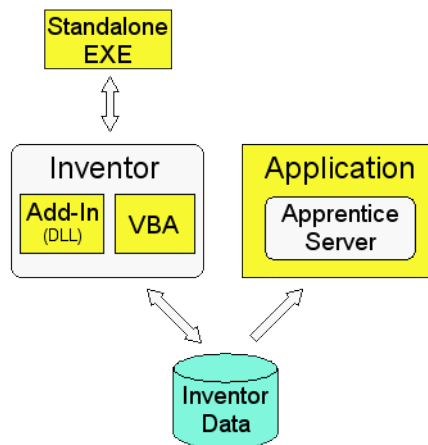


Fig. 1. Inventor API

2.2 Product Functions

The product will provide users with suggested commands which are generated based upon the users previous usage. This system will develop its functionality without direct interactions with the user. Over time this product should develop to be tailored to the users practices, allowing for the user to enhance their productivity by minimizing the time spent on repeated tasks. The suggested command itself will make an appearance within the user interface.

2.3 User Characteristics

In general, the user base of Autodesk Inventor consists of customers who have used the software for years and are familiar with how to complete the tasks they have set for themselves. This means that the target user for this product is someone who is well versed in what commands are available to them and exactly what each command does. Inventor as a software has numerous different uses which results in a wide variety of users, along with the reason for choosing Inventor. The one constant will be their high level of skill in what they commonly use Inventor for.

2.4 Limitations

The Autodesk Inventor application is a large piece of software, requiring a significant portion of power to run. With the inclusion of a program that will constantly be taking inputs and processing them in order to make a suggestion, which could lead to processes taking a noticeably long time. This would defeat the purpose of the add-in if the suggestion takes too long to be presented to the user. Another limitation is the library used to develop the machine learning algorithm must work alongside the Autodesk Inventor API.

2.5 Assumptions and Dependencies

Our assumption is that a user will use the sketch mode for a significant amount of time, and make use of a significant amount of commands in order to justify the development of this add-in.

A dependency will be the requirement that a user has the Autodesk Inventor software installed.

2.6 Apportioning of Requirements

The following graphic outlines our proposed agenda for the projects duration. While it is expected that all tasks will be completed within their allotted times, overflow periods have been included into this timeline.

PROJECT TIMELINE

PROJECT TITLE	Autodesk Command Analytics	COMPANY NAME	Autodesk
PROJECT MANAGER	Andrew Falx	DATE	10/30/18

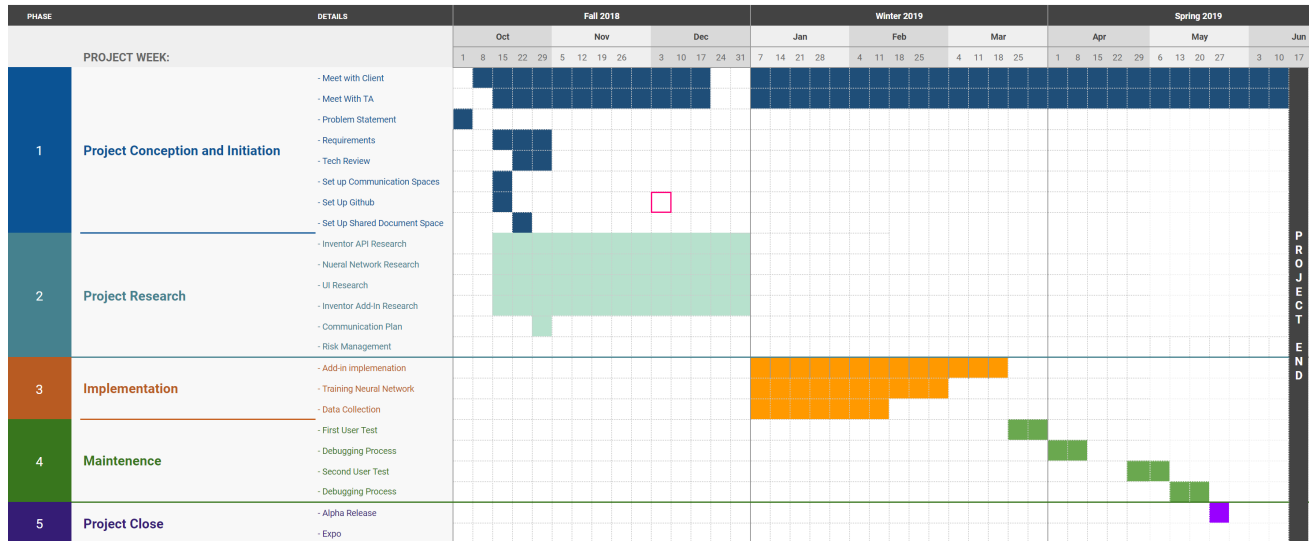


Fig. 2. GANTT Chart

3 SPECIFIC REQUIREMENTS

This section will contain a description of all functional and nonfunctional requirements for the Autodesk Inventor Command add-in. Details on the features of the add-in in question will be fully examined.

3.1 Functional Requirements

This section defines functional requirements applicable to the systems requirements.

ID: FR1

TITLE: Download Autodesk Inventor

DESC: The application is accessible from the Autodesk website [1].

RAT: The user must have access to the base software package in order for the add-in to be of any benefit.

DEPEND: None

ID: FR2

TITLE: Download add-in

DESC: Download the add-in from the Autodesk store.

RAT: The add-in must be available and be able to easily link with the Inventor application.

DEPEND: FR1

ID: FR3

TITLE: Open the Autodesk Inventor executable

DESC: A user should start the application. Depending on new users, a tutorial will appear.

RAT: This is where interaction with the Inventor environment occurs.

DEPEND: FR1

ID: FR4

TITLE: Create new design

DESC: A user should click the new button in the top left corner of the interface

RAT: Allows the user to create a new project.

DEPEND: FR1, FR2

ID: FR5

TITLE: Project Management Menu

DESC: A project management menu will give a selection to create a part, assembly, drawing, or presentation in 2D or 3D.

RAT: In order for the user to select what type of project they are working on.

DEPEND: FR3

ID: FR6

TITLE: Utilize commands

DESC: The user will begin designing their object using Inventor commands

RAT: In order for the user to create designs and use the downloaded add-in

DEPEND: FR5

ID: FR7

TITLE: Predictive Analytics

DESC: Our add-in will begin to generate predicted commands based on a sequence of the users commands.

RAT: The add-in will need input in order to generate predicted commands.

DEPEND: FR6, FR2

ID: FR8

TITLE: Stretch Goal for FR7

DESC: A command will pop up in near the cursor with the predicted command.

RAT: This will aid the user to become more efficient with Inventor commands.

DEPEND: FR7, FR6, FR2

3.2 Performance Requirements

ID: QR1

TITLE: Predictive Speed

DESC: The system will take no longer then 3 seconds to visually display to the user the predicted command

RAT: In order for the user to in fact increase productivity by lessening work time

DEPEND: None

ID: QR2

TITLE: Visual Understanding

DESC: The system will display the predicted command in an easy to understand method.

RAT: In order for the user to clearly understand what the system is recommending

DEPEND: None

ID: QR3

TITLE: Data Testability

DESC: The system will be easy to grade on accuracy due to the nature of a user choosing to follow the systems recommendation or to choose another option.

RAT: The system will know when it has predicted correctly or not

DEPEND: None

ID: QR4

TITLE: Runtime Minimization

DESC: The effect upon the main applications normal runtime must be minimal.

RAT: If the main application is slowed by the addition of this add-in then its core purpose is rendered null.

DEPEND: None

3.3 Design Constraints

The add-in solution is a local tool. That is to say that all data available will be locally collected and stored. The algorithm used to make command prediction will therefore only be able to learn from one specific source. The underlying specification for the add-in is to work faster than the user selecting the command from the current UI toolbar. This will put time constraints on how fast the algorithm will retrieve a predicted command. Thus, the final algorithm should have an efficient calculated running time.