Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 53 (2016) 231 – 236

www.elsevier.com/locate/procedia

10th International Conference on Axiomatic Design — ICAD 2016

The axiomatic design of Chessmate: a chess-playing robot

Freyja Yeatman Ómarsdóttir[a], Róbert Bjarnar Ólafsson[a], Joseph Timothy Foley[a,*]

[a]*Reykjavik University, Menntavegur 1, Reykjavík 101, Iceland*

* Corresponding author. Tel.: +354–599–6569; fax: +354–599–6201. *E-mail address:* foley@ru.is

Abstract

Successfully completing a project on time is often a difficult task especially when the project is not well defined. This paper demonstrates the application of Axiomatic Design principles to shape and direct a multi-disciplinary project from initial conception to the final tested product. This product is Chessmate: a small robot which plays chess on a physical board. This robot is intended as a telepresence mechanism or for players who are physically challenged. Verifiable requirements were developed at the beginning of the project based upon this top level goal. These Functional Requirements ensured that the team focused on the necessary capabilities of the end product even while working on electrical, mechanical, and software elements in parallel. Construction of a design matrix identified sources of coupling that would require additional effort to avoid delays. Coupling was reduced in software by careful Application Programming Interface (API) and abstraction development. Testing parameters were explicitly stated by the requirements enabling regular validation in both software and hardware. The result was a complete chess-playing system from start to finish in 12 weeks.

Peer-review under responsibility of the scientific committee of The 10th International Conference on Axiomatic Design

Keywords: Axiomatic Design; chess; robotics;

1. Introduction

Chessmate is a software controlled mini robotic arm which moves small chess pieces around a chessboard. The target audience was players who wished to play against each other remotely or who have physical disabilities. In both of these cases, interfacing with a standard chess board is not usually an option.

This project developed over 12 weeks in the Reykjavík University Mechatronics 1 course. In this course, Axiomatic Design Theory (ADT) was first introduced as a mechanism for structuring multi-disciplinary complex projects. The ADT principles [1] were applied at the start of the project and provided guidance for the most challenging elements in the design.

Nomenclature

ADT	Axiomatic Design Theory
FR	Functional Requirement
DP	Design Parameter
C	Constraint

1.1. Chess Robots

The concept of a robot that can play a game such as chess has been around for a long time. Sadly, the earliest well-known robot "The Turk" was actually a hoax where an operator inside a cabinet controlled a manikin which moved the pieces [2]. A simple search turns up many similar concepts for automated chess players. The two most common mechanism for moving pieces is either a robotic arm that lifts the pieces or an X-Y controlled magnetic slider underneath the board [3].

Using a similar mechanism, a project called "Wireless Arduino Powered Chess" consists of twin chess boards which physically mimic each other's moves and could theoretically be separated by thousands of miles [4] Chhangani describes a similarly named Arduino-based telepresence chess robot which uses an X-Y magnetic pickup below the chessboard for the remote player [5]. In both designs, piece movement is identified by magnetic reed switches beneath each board.

A graduation project from the Kuwait University College of Engineering and Petroleum [6] as well as a number of home projects (e.g. [7] posted on `instructables.com` use a 3 or 4 axis robotic arm. The mechanism for piece detection is also often reed switches.

The focus of these existing projects is to provide an AI (or remote player) physical mimicry during games. The human

Peer-review under responsibility of the scientific committee of The 10th International Conference on Axiomatic Design
doi:10.1016/j.procir.2016.07.002

player must move the piece on the physical board to indicate a move in these cases. In comparison, the aim of Chessmate is to allow players with physical barriers to moving the pieces (e.g. remote distance or limited hand-eye coordination) to still enjoy the game.

Chess robots are inherently mechatronic systems with many parts. Developing one requires interacting software, electronics, and mechanical components. Multi-disciplinary products like this are particularly vulnerable to becoming unreasonably complicated and complex [8] without a guiding methodology. Modern designers have a plethora of methodologies to choose from including Axiomatic Design [9], TRIZ [10], Design Thinking [11], and Innovative Design Thinking [12]. Each methodology has its own means and methods with a particular focus. Axiomatic Design Theory (ADT) was chosen to be included in the Mechatronics curriculum due to its successful deployment at Reykjavik University in many disciplines including additive manufacturing [13], rocket parachute deployment [14], educational spectrometers [15], and ultrasonic carburetors [16].

1.2. Axiomatic Design Theory

The Axiomatic Design Theory (ADT) methodology [1] is a structured way to formulate a design due to its focus on how different requirements interact. This is particularly relevant on time-constrained (or otherwise resource-constrained) projects where such coupling factors can result in significant delays and overruns [8,17]. This is due to these interactions creating often unpleasant surprises later in development. Best practice indicates that Functional Requirements (FR) are focused on transformative or action verbs and can be verified [9,18]. By following this rule, the testing process flows directly from these explicit criteria. This generalized description of "what is needed" reminds the entire team of the goals and how they will be validated without constraining how to solve them. In the same way, best practice for Design Parameters (DP) indicates the focus on a noun that can be quantified or instantiated. This restriction allowed the DPs to effectively be a consistent universal language for the designers to explain how a particular element was to be solved. The concept of using ADT as an intermediate language for different disciplines which otherwise have trouble communicating is discussed in [19].

At the beginning of the design, often the voice of the customer is translated into a set of Customer Needs. In the case of Chessmate, this domain was not investigated deeply due to the developers being the main customers. The primary Customer Need (CN0) was simply "Play chess with someone who cannot touch the pieces on my board."; no further decomposition was performed. Rather, the focus was placed on developing comprehensive FR and DP lists, then evaluating the coupling between them. This coupling is symbolized in a design matrix, which is a Cartesian product of all FR and DP combinations [20,21]. Where there is an interaction between an FR and DP, this is denoted by a non-zero coefficient, or in the case of the value being unknown, simply a placeholder variable X. Minor levels of coupling, often considered higher-order effects, are annotated with x to show their lessened effect. A diagonal matrix is "uncoupled" and satisfies the Independence Axiom: "to maintain the independence of the functional requirements (FRs)" [9]. Such a design can be easily optimized by adjusting a particular FR or DPs without affecting others. A diagonal matrix indicates a "decoupled" or "path-dependent" solution, which can still be optimized, but the ordering of parameter choice selection becomes important. All other design matrices are "coupled" and may have a usable local solution but usually resist modification and optimization [9]. Needless to say, the focus is on minimizing coupling wherever it may appear.

ADT's second axiom is "minimize the information content of the design." Simply put, ensure that the design has the highest probability of meeting the stated FRs. When systems are not able to meet FRs all of the time, this is denoted in ADT as "complexity" and is deeply explored in [8]. As will become apparent in the next section, this axiom became integral to the design of the interaction between the robot and its chess pieces. Finally, any factors to be considered that are not functional are categorized as "Constraints." These are often resource-focused and affect all of the design decisions; they need to be revisited often especially when choosing between otherwise equivalent implementations.

2. Design

The design of Chessmate essentially had 4 elements to it, a software component, an electrical component and two physical components: the epitome of mechatronics. As previously mentioned, it was critical at the beginning to develop a comprehensive set of requirements before proceeding. Discussion within the team including some general concepts developed the CN0 into primary requirement and the mechanism for satisfying it:

FR_0: Synchronize chessboards in two locations without touching pieces

DP_0: Arduino-controlled robotic arm moves pieces on board based upon user input into a serial terminal (remote or local).

With the top-level complete, it was time to employ ADT's process for decomposition called "zig-zagging." The process simply states that each pair of domains completes a full "mapping-decomposition" process before moving to the next level.

2.1. Top-level Decomposition

For the Chessmate project, two levels of Functional Requirements (FRs) and their corresponding Design Parameters (DPs) were developed. The first iteration of top-level FRs and DPs can be seen in Table 1.

Upon later review, it was discovered that these FRs and DPs were heavily coupled, redundant, and incorrectly formulated as described by Thompson [22]. Applying AD best practices allowed the designers to heavily refine the FRs and DPs to their core as shown in Table 3. Similar to Bragason et al. [14], some of the originally conceived FRs were actually constraints (Table 2) due to their effect on the system as a whole.

After this analysis had been completed the next step was to specify the robot and any related geometrical constraints. The MeArm platform from MeArm Robotics in the UK was chosen due to its affordability (9000 ISK), availability (2 weeks), and suitability for interfacing with an Arduino micro-controller (existing library and connectors). Once the MeArm arrived, it was