IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684,p-ISSN: 2320-334X, Volume 17, Issue 4 Ser. IV (Jul. – Aug. 2020), PP 53-66
www.iosrjournals.org

Estimation of Change Propagation Effort Using a Knowledge Perspective

Aamir Patel, Xiaoqi Zhang, Vince Thomson
(Mechanical Engineering, McGill University, Canada)

Abstract:
Background: *Many times, product complexity increases if customer requirements or product specifications are changed; moreover, the product development process is always more complex if change occurs, and this drives cost. In fact, complexity increases drastically if there are changes in any design activity as a change in one product function can trigger changes in other functions, i.e., change propagation, and then, many related design tasks are impacted. From a knowledge perspective, product development can be viewed as the supply and demand of knowledge between designers, driven by the relationships among product functions.*
Materials and Methods: *An approach for estimating the total effort needed to create the necessary change in product functions is introduced that is based on the need for new knowledge in order to design change impacted functions.This paper explores how to model change propagation paths, how to model the probability of change propagation, and how to use simulation to estimate the effort required to address design changes due to change propagation. Data from GE Hydro for the development of a hydroelectric generator is used as a demonstration to verify the proposed method.*
Results: *Simulation of a product development process using a knowledge perspective was able to determine the effort and span time for different changes for the design of a hydroelectric generator, and thus, estimate change propagation effort. Results showed that design effort increased exponentially with the amount of change propagation.*
Conclusion:*The proposed method successfully estimates effort due to change. The method also helps managers how to identify critical functions and their interfaces so as to provide insight into how to improve product development performance while reducing effort, and thus cost, at the same time.*

Key Word: *complexity; product development; change propagation; knowledge.*

Date of Submission: 29-08-2020 Date of Acceptance: 14-09-2020

I. Introduction

Product development (PD) is a very challenging activity with various iterative stages during which the product design team performs a crucial role by decomposing a product into different functions that are interdependent. This dependency among functions can involve many levels of complexity, which can make PD difficult to manage, especially when PD requires knowledge from multiple engineering domains. The ever increasing customer demand for more product functionality and the need to combine different types of knowledge to develop the functionality have resulted in more complex design processes[1]. One thing in which every company is interested is how to evaluate, predict and manage complexity efficiently to reduce total effort, which is the total hours that designers spend on all activities required for PD.

To evaluate the required effort for PD, significant work has been carried out with regard to various aspects of calculating complexity, such as information entropy theory[2], computational complexity theory[3], and axiomatic design[4]. However, a high level of inaccuracy for contemporary complexity measurement approaches still exits, which is caused by the fact that the actual design process in physical space is not consistent with the theoretical simulation in virtual space[5]. The evaluation of complexity can be improved by using a knowledge perspective where product functions embody the knowledge required to design a product[6], team members possess knowledge and use tools and networks to create product designs[7], and the PD process is impacted by the required dynamics of knowledge manipulation[8]. The complexity of the PD process is driven by the introduction of new technology needed to implant the knowledge for product improvement[9] and by the required intensity and diversity of knowledge[6]. To determine complexity, the knowledge embedded in functions and the relationships between the knowledge that is used to realize functions need to be considered[10,11].

The BZT (Bashir-Zhang-Thomson) complexity metric was developed using a knowledge perspective[12]. Itcan calculate the complexity of a product or project taking into account the required knowledge, and then, determine the needed effort from a graph of complexity versus effort created from historical data. An agent-based PD model/simulation was built using this metric to calculate the effort and span time of a development project[6]. However, the agent-based model considered rework to be caused by technical errors and integration issues only. Another source of rework that was not studied was engineering change, which impacts the time and cost to develop new versions of existing products.

Total effort increases drastically if there is an engineering change in any design activity during PD. This is because a single change in a function can induce a series of changes to connected functions, which is known as change propagation[13]. Every time change propagation occurs, an extra amount of effort is needed to accomplish the redesign of the connected, product functions. To manage engineering change efficiently, one must be able to comprehend and predict how change propagates through product functions during the PD design stage[13]. The questions that need to be explored are: how to model the propagation paths when a change happens, how to determine the probability of change propagation between a pair of dependent functions, and how to estimate the effort required to address the changes due to change propagation.

In order to address these three questions, a modelling/simulation approach using function and knowledge perspectives is presented. It calculates the total effort needed to create new product functions or modify existing functions due to change. Data from GE Hydro for the development of a hydroelectric generator is used to verify the proposed method that employs an agent-based PD process model.

The introduction has described the background to the problem of estimating PD effort caused by change propagation. Section II describes the use of both product function and knowledge perspectives to estimate design effort due to change propagation. Section II also presents the modelling of a design process and the required knowledge as well as the simulation methods used to obtain estimates of effort due to design change. The results of simulation of change propagation in the development of a GE Hydro hydroelectric generator is given in section III. Section IV presents conclusions and future work.

II. Material and Methods

2.1 Functional Analysis and the Knowledge Perspective

In PD, a product can be represented by a set of functions, which are then assigned to designers who are responsible for developing their detailed structure and behavior (figure 2.1). Under a knowledge perspective, PD can be viewed as a knowledge-intensive process where a product is the embodiment of knowledge, i.e., there is knowledge concerning functional requirements, knowledge of the product structure and behavior, and knowledge that the designers provide to realize the functions. Using this viewpoint, the essence of PD is the integration of multiple knowledge domains by designers with corresponding expertise to achieve the desired product functions.

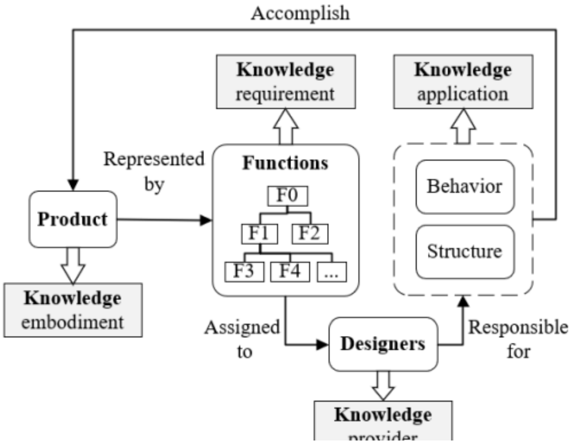


Figure 2.1 Function and knowledge perspectives of product development

From a knowledge perspective, product functions and designercapability should be mapped into different types of knowledge. Achieving the development of a product within a stipulated time relies on matching the required knowledge of product functions with designer capability in the corresponding knowledge domains. The key steps of knowledge-based PD are decomposing product requirements into functions viewed from a knowledge perspective, and then, creating product specifications using the appropriate knowledge.

2.1.1 Functional Decomposition

Kota and Ward[14] define a function as “the behavior which is required for a device to satisfy a given requirement”. Bytheway[15] defines a function as a transitive verb followed by an associated object, such as ‘store energy’ and ‘upload data’. A functional architecture of a product can be attained through functional decomposition techniques. Bashir and Thomson[16] describe this technique as: “each function can be decomposed into subfunctions and each subfunction can be further broken down into sub-subfunctions, and so on, where each of the lowest level functions meets one, or a combination, of the following conditions: it is considered simple; it is mapped to an existing component.” Representations of a hierarchy of functions are developed to spur creativity and prevent unintentional omission of product functionality; one of the most widely used functional hierarchy forms is the FAST (Function Analysis System Technique) diagram[15]. The FAST diagram of a hydroelectric generator by GE Hydro is given in figure 2.2. Functionality is not related to product embodiment or any PD method, which is oneof the advantages of using a functional perspective. Also, a FAST diagram allows product change to be tracked at any stage of PD[17].

2.1.2 Mapping Functions to Knowledge Space

As discussed in section 2.1.1, design is a knowledge-intensive process where every product function can be described in terms of different types of knowledge, viz., control, physics, electrical, mechanical, etc. Technical knowledge is defined as professional abilities used by designers that are acquired through professional learning or actual product development, which is divided into general knowledge and product knowledge[6]. General knowledge is the basic knowledge of a domain usually learned through training, and product knowledge is the knowledge obtained through developing products[6]. Thus, functions are developed through general knowledge and they are integrated through product knowledge (interfaces).

To map a function into a knowledge space, a scale is used to evaluate the required knowledge items used in designing a specific function. To this end, knowledge magnitude is used to indicate the importance of a certain knowledge during the development of a function. Knowledge items should be defined using the same scale, expressed by [0, r] ($r \in N^+$). For example, if the magnitude of a knowledge item is 0, it is not important for the creation of a function. Table 2.1 shows the 10 types of knowledge and their possible magnitudes needed to develop a hydroelectric generator.

Table 2.1 Knowledge scale for a hydroelectric generator[11]

# Knowledge	Minimum	General	Advanced
HVAC (heating, ventilating and air conditioning)	0	1	2
Air circulation	0	1	2
Water circulation	0	1	2
Heat transfer	0	1	2
Electric heat generation	0	1	2
Control	0	1	2
Mechanical engineering	0	1	2
Sensor technology	0	1	2
Physics	0	1	2
Electrical engineering	0	1	2

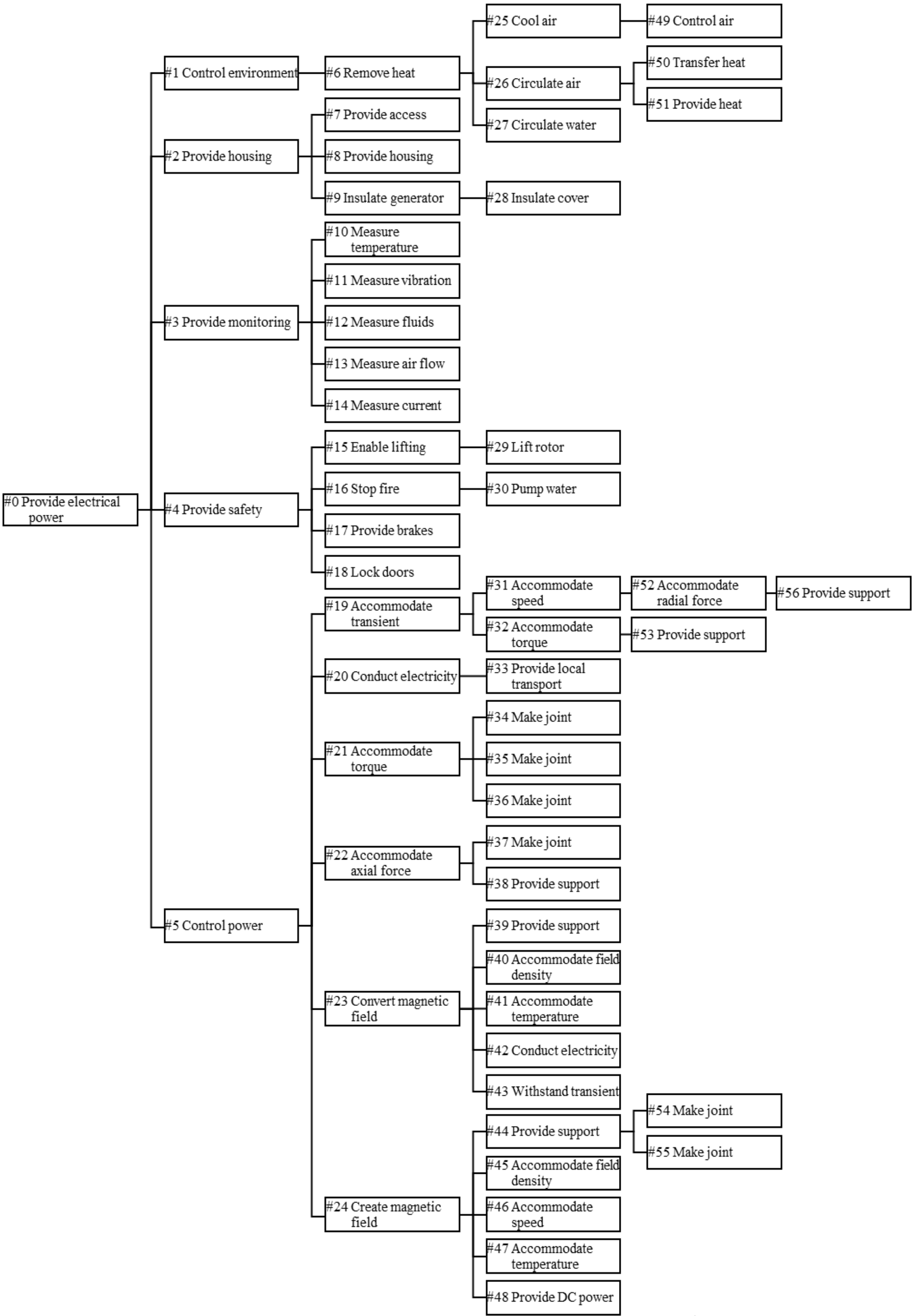


Figure 2.2 FAST diagram of a hydroelectric generator[16]

If the development of a function involves n knowledge items, they can form an n-dimensional space, called a knowledge vector $\boldsymbol{F}=(k_1, k_2,\ldots k_i\ldots, k_n)$[11], where $k_i$is the magnitude of the knowledge item i. Figure 2.3 shows the mapping of hydroelectric generator functions into knowledge vectors.

#	Function	Required knowledge									
		K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
0	F0	1	1	1	1	1	1	1	1	0	0
1	F1	1	0	0	1	1	1	1	1	0	0
2	F2	1	1	1	1	1	1	0	1	0	1
3	F3	0	0	0	1	0	1	1	0	0	0

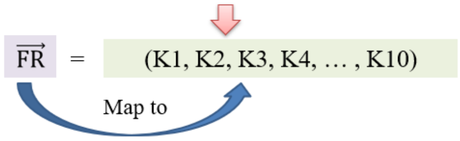


Figure 2.3 Mapping of functions into knowledge vectors

2.2 Methodology

2.2.1 Mapping Functions into Knowledge Space

PD is viewed as the supply and demand of knowledge between designers and product functions. This viewpoint requires modelling the interdependence and interactions among functions and designers.In an agent-based PD model, the product to be designed is modelled as a group of interdependent function agents requiring knowledge and effort to create or integrate them[6]. Designers are modelled as designer agents with appropriate knowledge aimed at creating product functions through development activities. To develop functions, designer agents perform three main development tasks: technical work, communication and consultation. This is shown in figure 2.4 for a functional change.

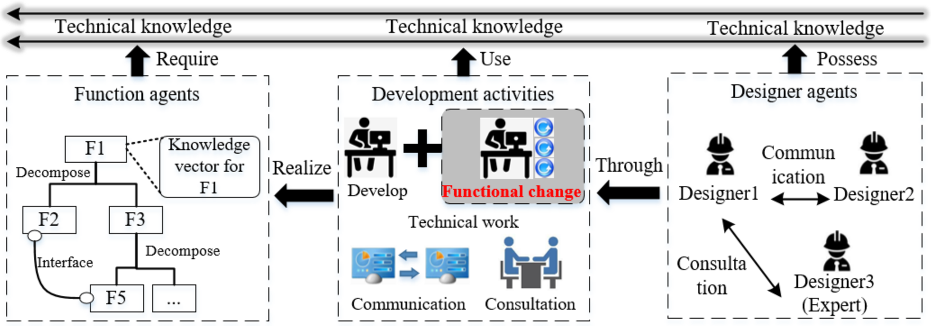


Figure 2.4 An agent-based PD model that considers functional change

The aim of technical work is to develop product functions, which is an iterative process. Before PD completion, a designer goes through a review task, where rework is requested when technical issues, errors or insufficiencies are detected. After the completion of all tasks, intermediate or senior designers begin the integration work, i.e., development of interfaces. Rework due to engineering change is encountered when a function is added or removed, a knowledge item is added or removed, or any interface between functions is modified.

In the agent-based, PD process model/simulation presented in this paper, functional change is considered in the technical work of development activities, as represented by the area highlighted in grey infigure 2.4. In order to predict the complexity caused by functional change and to calculate the additional effort, a change propagation model was constructed, and the probability of functional change propagation and added effort were calculated. The model is described in detail in the following sections.

2.2.2 Information Modelling of FunctionalChangePropagation

Four classes of information used to model products from a functional point of view are: the functions that represent the product to be developed, the FAST diagram that shows the relationships among functions, knowledge that is able to quantify functions, and interfaces that define the relationships between functions, as shown in figure 2.5. The first three classes have been discussed previously, and interfaces are discussed below.

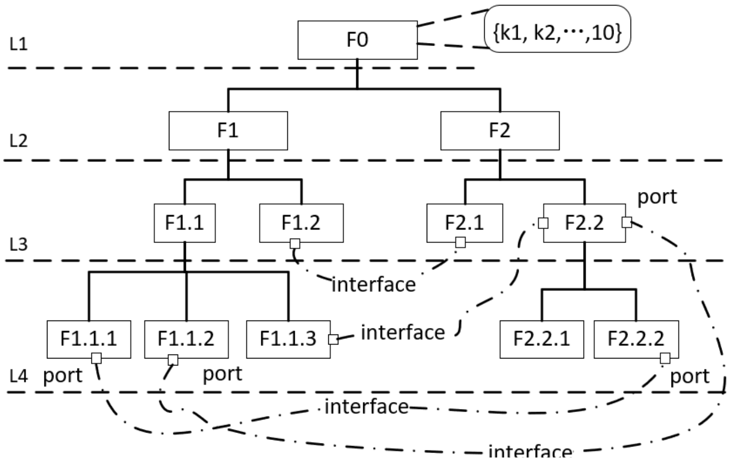
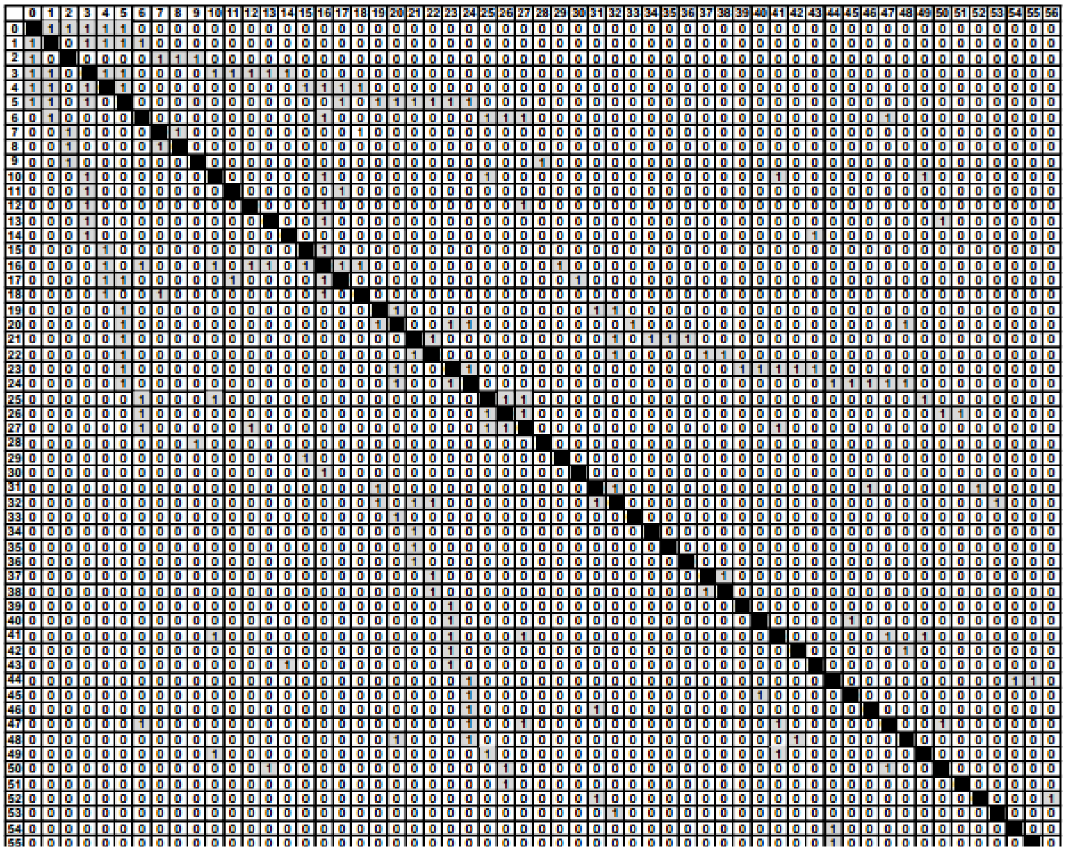


Figure 2.5 FAST diagram of a product

Interfaces refer to the connections between subsystems and components of a product where they allow communication between functions[18]. They can be described by a Design Structure Matrix (DSM), where'1' indicates that there is an interface and '0' indicates that there is no interface, i.e., the functions in a row and the corresponding column are independent. The DSM in figure 2.6 shows the interfaces for the 57 functions of the GE Hydro hydroelectric generator where interfaces are shown by a 1 and coloured grey.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
0	■	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	■	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	■	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	0	■	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	0	1	■	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	0	1	0	■	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	■	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	■	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	1	0	0	0	0	1	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	1	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0	0	■	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
11	0	0	0	1	0	0	0	0	0	0	0	■	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	1	0	0	0	0	0	0	0	0	■	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	1	0	0	0	0	0	0	0	0	0	■	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	■	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	1	0	1	0	0	0	1	0	1	1	0	1	■	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	■	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	■	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	■	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
21	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	■	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	■	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
26	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	■	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
27	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	■	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	■	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	1	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	1	1	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	■	0	0	1	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	■	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	■	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	■	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0	1
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	■	0	0	0
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	■	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	■	0

Figure 2.6 Interfaces between functions in a hydroelectric generator are indicated by a 1 and shaded in grey[11]

Figure 2.7 shows an interface that represents the interaction between two functions. Pimmler and Eppinger[19] categorized interfaces with five interaction types or characteristics between two elements

1 material interrelation identifies needs for material exchange
2 energy interrelation identifies needs for energy transfer
3 spatial interrelation identifies needs for adjacency or orientation
4 information interrelation identifies needs for exchange of information, and
5 control interrelation identifies needs for transmission or reception of control signals.

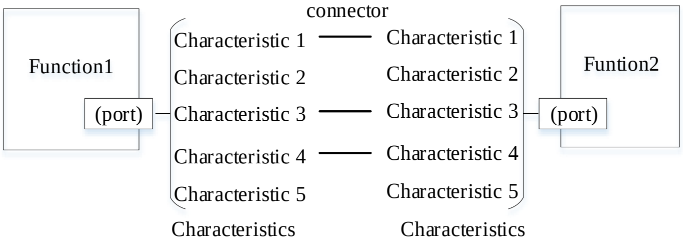


Figure 2.7 Interface model with characteristics 1, 3, 4 being shared between two functions[20]

Interface characteristics are different for every design. For example, a hydroelectric generator converts the power of falling water into electricity when the generator 'controls power', 'conducts electricity' and 'provides safety' (figure 2.2). Some interface characteristics for energy are: 'convert energy' and 'conduct electricity' and for material: 'cool air', 'circulate air' and 'remove heat'. Moreover, some interface features are more important than others. Importance is shown as a three-point scale: undesired, desired and required in table 2.2.

Table 2.2 Quantified interface characteristics[20]

#	Characteristics	Scales and Need		
		Undesired	**Desired**	**Required**
C1	Spatial	0	1	2
C2	Power	0	1	2
C3	Material	0	1	2
C4	Information	0	1	2
C5	Control	0	1	2

With interface information, one can trace change propagation between functions by following rules of propagation. Wang et al.[20]give the followingfour propagation rules:

1 "A change can propagate through functional relationships when
  a) a child function influences a parent function
  b) a parent function influences a child function
  c) a peer function influences another peer function.
2 A change can propagate due to dependency, i.e., there is an interface between a pair of functions."

As shown in figure 2.8, when there is a change in *function 2.2*, the following functions can be changed according to the rules: 1a) - parent functions (*function 2* and *function 0*), 1b) - child functions (*function 2.2.1* and *function 2.2.2*), 1c) - peer functions (*function 2.1*) and 2 - dependent functions (*function 1.1.3* and *function 1.1.2*). There can be a direct impact, e.g., generated by *function 2.2* to function 2.1, and an indirect impact, e.g., generated by *function 2.2* to *function 0* via *function 2*. Thus, possible impact can occur due to a functional relationship or due to a dependency relationship, and it can be direct or indirect.

2.3 Calculating the Probability ofFunctionalChange Propagation

Propagation is governed by the knowledge sameness of different function. Knowledge sameness is due to the extent of common knowledge. The more two functions have similar knowledge, the higher is the change propagation probability. There is no change propagation with no common knowledge.

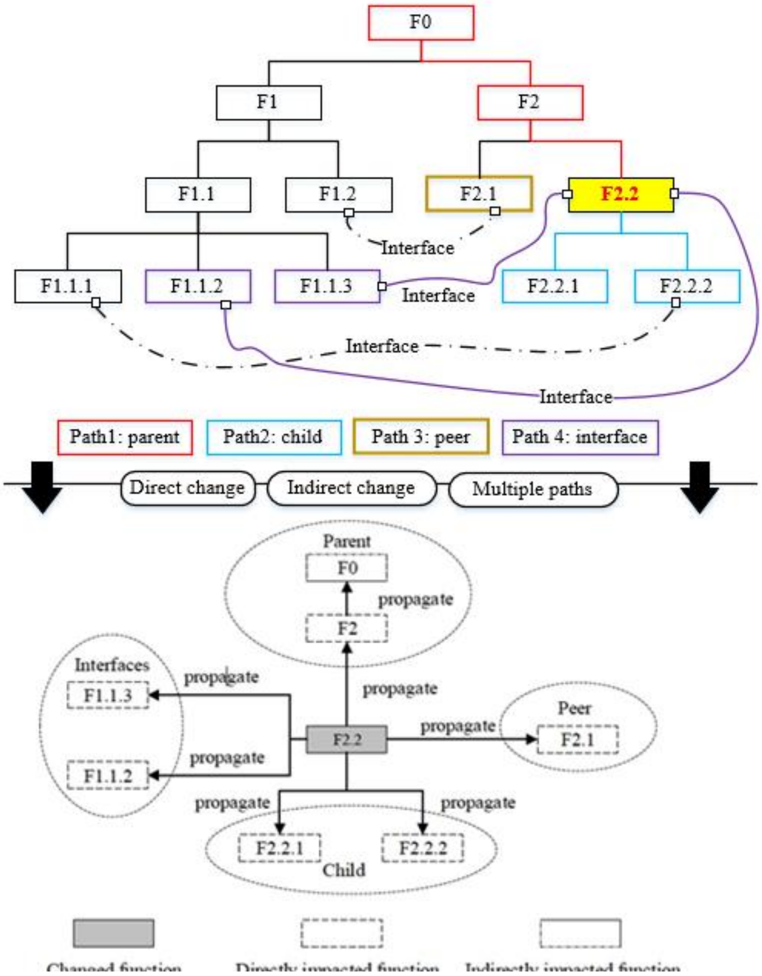


Figure 2.8 Four possible paths for change propagation

Knowledge sameness is used to find propagation of change probability. However, we start with knowledge difference (KD) where KD_{ij} for functions i and j is computed by equation (1)[6]. $r$is the maximum of the scale [0, r] (for the hydroelectric generator r =2), and θ_{ij} is the angle between knowledge vectors $\vec{F_i}$ and $\vec{F_j}$. KD is shown in figure 2.9. We use $r^{\sin\theta_{ij}}$ to denote KD because it is [1, r] when θ_{ij} is [0°, 90°], andthe biggerθ_{ij},the bigger is KD. In the extreme cases, if $\vec{F_i}$ and $\vec{F_j}$ require entirely different knowledge, KD_{ij} is 2 and the likelihood of change propagation is 0, whereas, if $\vec{F_i}$ and $\vec{F_j}$ have the same knowledge, KD_{ij} is 1, and the likelihood of change propagation is 1.

$$KD_{ij} = r^{\sin\theta_{ij}} = r^{\sqrt{1-\left(\frac{\vec{F_i}\cdot\vec{F_j}}{\|\vec{F_i}\|\cdot\|\vec{F_j}\|}\right)^2}} \tag{1}$$

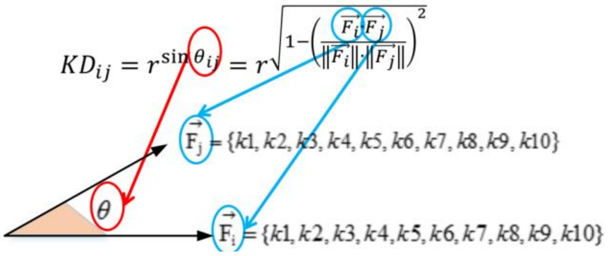


Figure 2.9 The difference of knowledge between two knowledge vectors[6]

Now, the sameness of knowledge is $\left(1-\frac{KD_{ij}-1}{r-1}\right)$ and change propagation probability is given by equation (2), and CP_{ij}^{Dir} is caused by direct propagation[20]. The larger the change probability is, the larger is the probability of having an impacted function.

$$CP_{ij}^{Dir} = \left(1-\frac{KD_{ij}-1}{r-1}\right)\times 100\% \quad (2)$$

Graphs of change propagation can be considered to be logic graphs in line with the combined likelihood algorithm[18]. Then, vertical connections in a graph are "∩" (indirect change) and horizontal connections are "∪" (possible multiple paths). Equations (3) and (4) define the change propagation likelihood[20] where equation (3) computes the probability of indirect change propagation, and equation (4) computes the probability of all change propagation.

$$CP_{ac}^{In} = CP_{ab}^{Dir} \cap CP_{bc}^{Dir} = CP_{ab}^{Dir} \times CP_{bc}^{Dir} \quad (3)$$

$$CP_{ij}^{Com} = CP_{ij}^{D} \cup CP_{ij}^{I} = CP_{ij}^{D} + CP_{ij}^{I} - \left(CP_{ij}^{D} \times CP_{ij}^{I}\right) = 1-(1-CP_{ij}^{D})\times(1-CP_{ij}^{I}) \quad (4)$$

The BZT complexity metric estimates effort as a function of product complexity, which has two components: technical complexity due to the amount of required knowledge and integration complexity due to required interfaces[12]. As product complexity increases, so does design effort. Thus, after identifying potential propagation paths arising from the change of function i, $\overrightarrow{F_i}$, change propagation complexity (CPC) caused by a change in $\overrightarrow{F_i}$is calculated using equation (5), which represents the sum of the overall complexity C of those functions impacted by change propagation and its change probability is CP_{ij}^{Com}. In equation (5), TC_i refers to the technical complexity of functional development, and IC_i is the complexity due to integratingfunctions (interfaces), where $L_i$is the level of $F_i$in the function decomposition structure, e.g., the FAST diagram in figure 2.2 has 6 levels.

$$CPC_{F_i-changed} = \sum_{j=1}^{n} C_j \times CP_{ij}^{Com} = \sum_{j=1}^{n}\left(\left(\sum_{i,j} IC_{i,j} + \sum_{i} TC_i * L_i\right)\times CP_{ij}^{Com}\right) \quad (5)$$

2.4 Estimating the Extra Effort due to FunctionalChange Propagation

From the above analysis, a six-step process for estimating the effort for functional change with required inputs and expected outputs is the following:

1) Indicate the function that is changed in the FAST diagram. From a knowledge perspective there are 3 possibilities: a function is added or removed, a required knowledge is added or removed, or an interface between functions is modified, added or removed.
2) Consider the 4 propagation paths given in section 2.2.2 and select functions where any knowledge is the same as that of the changed function. To do this, compare each knowledge of each pair of functions one at a time.
3) Indicate possible propagation using a functional decomposition diagram and a functional dependency diagram.
4) Use knowledge sameness to compute the change probability for any function. The more there is knowledge sameness, the more that change is possible. The probabilities are computed with equations (1), (2) and (3). The total probability for combined change propagation is computed with equation (4).
5) Calculate the complexity of the changed product using equation (5).
6) Determine the extra effort due to change propagation by taking the difference in effort of a baseline model/simulation of the PD process, which represents all functions before any change, and of a model/simulation with the changed function. To create the model with the changed function, insert the probabilities computed in step (4) and the new product complexity computed in step (5) into the agent-based PD process model.

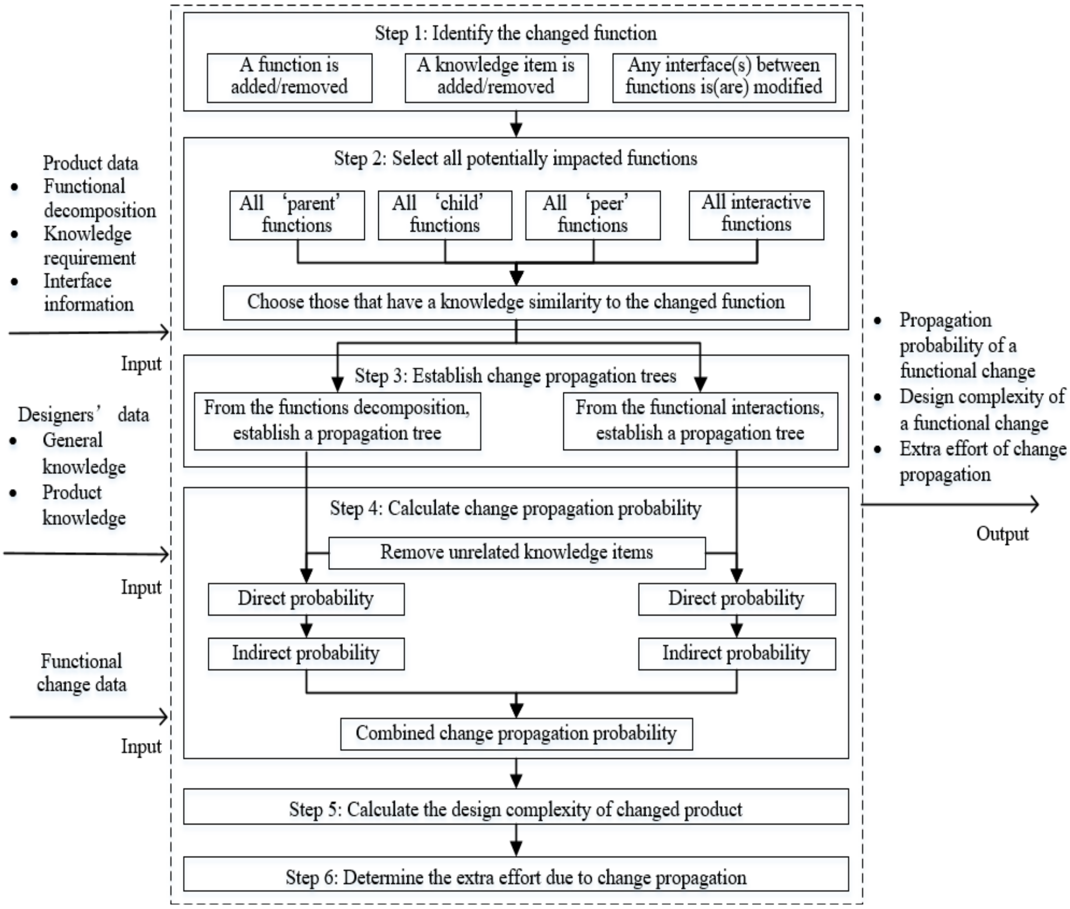


Figure 2.10 Process for estimating change propagation effort

III. Result

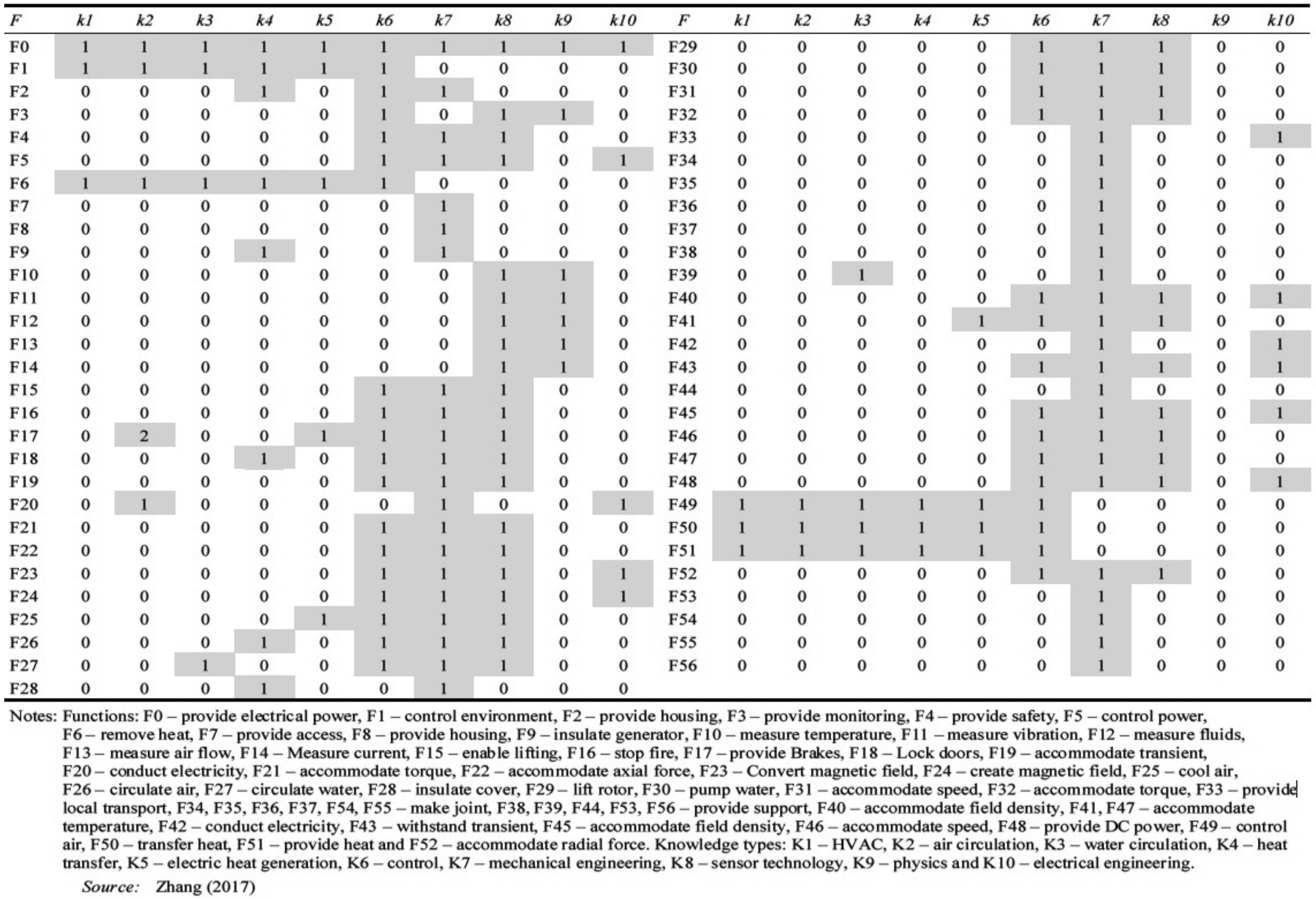
The procedure for determining change propagation is demonstrated with the design of a GE Hydro hydroelectric generator. The generator is an excellent instance with its 57 functions, various interfaces and its 10 knowledge items. The functional decomposition structure is shown in figure 2.2. The interfaces are shown using the DSM in figure 2.6. The knowledge items are tabulated in table 3.1.

To study the effect of change propagation in a hydroelectric generator, various scenarios were modelled and simulated. To generate scenarios for studying change propagation, a particular function, function #31, was assumed to change. Then, using the technique for computing the change propagation probability, all impacted functions were identified. The impacted functions were found to be functions 0, 5, 19, 21, 22, 31, 32, 46, 52, and 56. For these functions, various test situations were generated. For example, knowledge items were changed, extra interfaces were added, functions were modified, etc. The simulation of a PD process model was used to calculate the effort and time span for each case. Descriptions of the 15 scenarios are listed in table 3.2.

The PD process computer model used Anylogic software, which allowed the use of both discrete event and agent-based modelling paradigms. Simulation of the PD process model calculated effort and span time for the execution of the PD process. The original model[6]was tuned to work only with specific input data for the GE Hydro hydroelectric generator. To enable this model to work with other PD processes, a MATLAB code was developed that extracted and formatted all relevant input data from a master spreadsheet in a form that was readable by the PD process model. During simulation, each scenario was run five times to obtain averages due to the stochastic nature of the model. The simulation results are shown in figure 3.1.

Scenario #1 is the baseline scenario for this study. This scenario includes all original 57 functions and the 10 knowledge items of the GE hydroelectric generator. The result gives the total effort required to completely design a GE generator. Note the larger amount of effort for scenario 1 that shows the effort for the complete design compared to scenarios 2-15 that only show the effort for a change to the original design.For scenarios 2 and 3, the model was run for only impacted functions with some knowledge items changed. In scenario 4, two new random functions were added to the list of impacted functions. The model was then run with one and two new random interfaces with different knowledge items. These scenarios were numbered 5-15. The main aim of these scenarios was to cover as many of the possible instances of change propagation, i.e., addition/removal of functions, addition/removal of knowledge items, and modification of interfaces. The simulation results for the sample scenarios gave how much extra effort was needed for the possible required design changes, which fulfilled the objective of the research.

Table 3.1 Knowledge requirements for designing a hydroelectric generator[11]



F	*k1*	*k2*	*k3*	*k4*	*k5*	*k6*	*k7*	*k8*	*k9*	*k10*	*F*	*k1*	*k2*	*k3*	*k4*	*k5*	*k6*	*k7*	*k8*	*k9*	*k10*
F0	1	1	1	1	1	1	1	1	1	1	F29	0	0	0	0	0	1	1	1	0	0
F1	1	1	1	1	1	1	0	0	0	0	F30	0	0	0	0	0	1	1	1	0	0
F2	0	0	0	1	0	1	1	0	0	0	F31	0	0	0	0	0	1	1	1	0	0
F3	0	0	0	0	0	1	0	1	1	0	F32	0	0	0	0	0	1	1	1	0	0
F4	0	0	0	0	0	1	1	1	0	0	F33	0	0	0	0	0	0	1	0	0	1
F5	0	0	0	0	0	1	1	1	0	1	F34	0	0	0	0	0	0	1	0	0	0
F6	1	1	1	1	1	1	0	0	0	0	F35	0	0	0	0	0	0	1	0	0	0
F7	0	0	0	0	0	0	1	0	0	0	F36	0	0	0	0	0	0	1	0	0	0
F8	0	0	0	0	0	0	1	0	0	0	F37	0	0	0	0	0	0	1	0	0	0
F9	0	0	0	1	0	0	1	0	0	0	F38	0	0	0	0	0	0	1	0	0	0
F10	0	0	0	0	0	0	0	1	1	0	F39	0	0	1	0	0	0	1	0	0	0
F11	0	0	0	0	0	0	0	1	1	0	F40	0	0	0	0	0	1	1	1	0	1
F12	0	0	0	0	0	0	0	1	1	0	F41	0	0	0	0	1	1	1	1	0	0
F13	0	0	0	0	0	0	0	1	1	0	F42	0	0	0	0	0	0	1	0	0	1
F14	0	0	0	0	0	0	0	1	1	0	F43	0	0	0	0	0	1	1	1	0	1
F15	0	0	0	0	0	1	1	1	0	0	F44	0	0	0	0	0	0	1	0	0	0
F16	0	0	0	0	0	1	1	1	0	0	F45	0	0	0	0	0	1	1	1	0	1
F17	0	2	0	0	1	1	1	1	0	0	F46	0	0	0	0	0	1	1	1	0	0
F18	0	0	0	1	0	1	1	1	0	0	F47	0	0	0	0	0	1	1	1	0	0
F19	0	0	0	0	0	1	1	1	0	0	F48	0	0	0	0	0	1	1	1	0	1
F20	0	1	0	0	0	0	1	0	0	1	F49	1	1	1	1	1	1	0	0	0	0
F21	0	0	0	0	0	1	1	1	0	0	F50	1	1	1	1	1	1	0	0	0	0
F22	0	0	0	0	0	1	1	1	0	0	F51	1	1	1	1	1	1	0	0	0	0
F23	0	0	0	0	0	1	1	1	0	1	F52	0	0	0	0	0	1	1	1	0	0
F24	0	0	0	0	0	1	1	1	0	1	F53	0	0	0	0	0	0	1	0	0	0
F25	0	0	0	0	1	1	1	1	0	0	F54	0	0	0	0	0	0	1	0	0	0
F26	0	0	0	1	0	1	1	1	0	0	F55	0	0	0	0	0	0	1	0	0	0
F27	0	0	1	0	0	1	1	1	0	0	F56	0	0	0	0	0	0	1	0	0	0
F28	0	0	0	1	0	0	1	0	0	0											

Notes: Functions: F0 – provide electrical power, F1 – control environment, F2 – provide housing, F3 – provide monitoring, F4 – provide safety, F5 – control power, F6 – remove heat, F7 – provide access, F8 – provide housing, F9 – insulate generator, F10 – measure temperature, F11 – measure vibration, F12 – measure fluids, F13 – measure air flow, F14 – Measure current, F15 – enable lifting, F16 – stop fire, F17 – provide Brakes, F18 – Lock doors, F19 – accommodate transient, F20 – conduct electricity, F21 – accommodate torque, F22 – accommodate axial force, F23 – Convert magnetic field, F24 – create magnetic field, F25 – cool air, F26 – circulate air, F27 – circulate water, F28 – insulate cover, F29 – lift rotor, F30 – pump water, F31 – accommodate speed, F32 – accommodate torque, F33 – provide local transport, F34, F35, F36, F37, F54, F55 – make joint, F38, F39, F44, F53, F56 – provide support, F40 – accommodate field density, F41, F47 – accommodate temperature, F42 – conduct electricity, F43 – withstand transient, F45 – accommodate field density, F46 – accommodate speed, F48 – provide DC power, F49 – control air, F50 – transfer heat, F51 – provide heat and F52 – accommodate radial force. Knowledge types: K1 – HVAC, K2 – air circulation, K3 – water circulation, K4 – heat transfer, K5 – electric heat generation, K6 – control, K7 – mechanical engineering, K8 – sensor technology, K9 – physics and K10 – electrical engineering.

Source: Zhang (2017)

Simulation of the PD process model determined the effort and span time for a set of product functions as shown in figure 3.1. In addition to effort, span time was observed in order to verify that each simulation was run correctly.

Table 3.2 Scenarios for simulating the GE hydroelectric project model for change propagation

#	Included functions	Included interfaces	Included knowledge items
1	All Original data	All	All
2	F31 + Impacted functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56	No change All predefined interfaces	All
3	F31 + Impacted functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56	No change All predefined interfaces	Only mechanical
4	Added 2 new functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56, F23, F37	No extra interface Basic parent-child interfaces	All
5	Added 2 new functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56, F23, F37	1 extra interface F22-F23	All
6	Added 2 new functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56, F23, F37	2 extra interfaces F22-F23 & F37-F19	All
7	Added 2 new functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56, F23, F37	No extra interface Basic parent-child interfaces	Only mechanical
8	Added 2 new functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56, F23, F37	1 extra interface F22-F23	Only mechanical
9	Added 2 new functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56, F23, F37	2 extra interfaces F22-F23 & F37-F19	Only mechanical
10	F31 + Impacted functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56	No change All predefined interfaces	Mechanical + Control
11	Added 2 New Functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56, F23, F37	No extra interface Basic parent-child interfaces	Mechanical + Control
12	Added 2 new functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56, F23, F37	1 extra interface F22-F23	Mechanical + Control
13	Added 2 new functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56, F23, F37	2 extra interfaces F22-F23 & F37-F19	Mechanical + control
14	F31 + Impacted functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56	1 extra interface F46-F21	Mechanical + Control
15	F31 + Impacted functions F0, F5, F19, F21, F22, F31, F32, F46, F52, F56	2 extra interfaces F46-F21 & F56-F32	Mechanical + Control

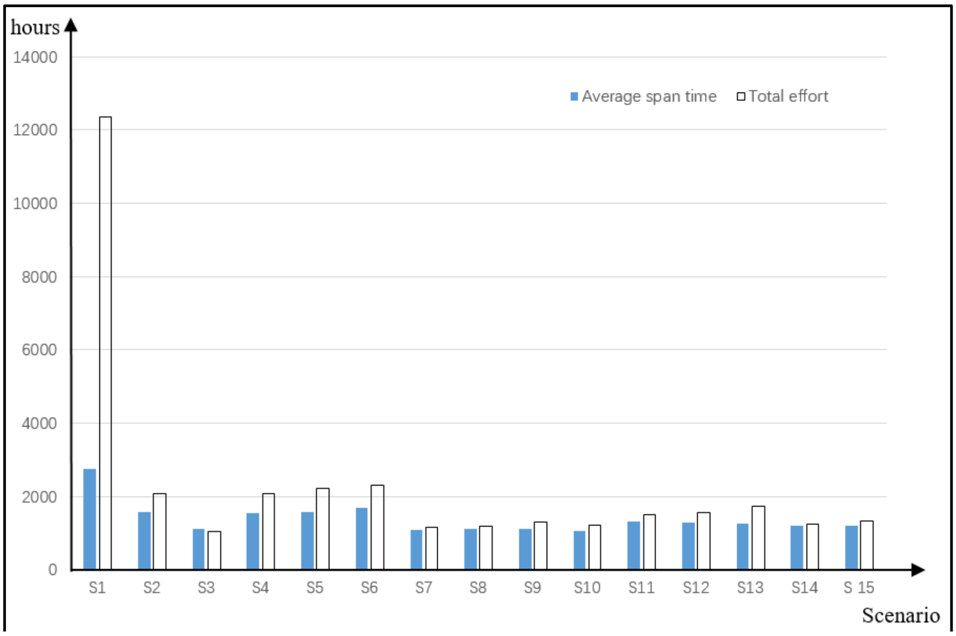


Figure 3.1 Simulation results for 15 scenarios using GE hydroelectric project data

IV. Conclusion

The novelty of the method described in this paper lies in its views of a product from both a functional and a knowledge perspective, and the use of agent-based simulation to estimate the amount of design effort due to change propagation. The use of a functional view to represent a product allows a simple way to assign the required knowledge for a product and any change to that knowledge. The method uses a knowledge perspective where the development of product functions needs knowledge and engineers deliver the knowledge. Effort to perform design change is linked to complexity where both product and design task complexity are increased by the intensity and diversity of the knowledge required to change functions, and where increased complexity causes increased effort. The method uses a modified BZT complexity metric and a simulation of the PD process to determine the design effort to accomplish a change in a product function.

Data from GE Hydro for the development of a hydroelectric generator was used to validate the proposed method. A baseline scenario was created, and 14 change scenarios were generated to test various propagation situations. The effort to execute the change for each of the 14 scenarios was calculated.

Managers can use models of their own PD processes. Analysis of results can identify critical functions and/or interfaces during change propagation in order to show which function(s) have the most effect on the total effort of a project. Managers can also use results to optimize the design process. Overall, simulation results can provide managers insight into how to improve PD performance while reducing effort (cost) at the same time.

The method to estimate change propagation effort was validated using the data from one case, the data from GE Hydro for a hydroelectric generator. Data from other cases are needed to further validate the method's correctness. Liaising with companies to validate the method is a future work. Also, the model in this study uses a probabilistic, statistical method to calculate results. Estimations of error need to be made to determine the degree of uncertainty in the simulation results.

References

[1]. Kleinsmann M, Buijs J, Valkenburg R. Understanding the complexity of knowledge integration in collaborative new product development teams: a case study. Journal of Engineering and Technology Management. 2010,27(1):20-32.

[2]. Efthymiou K, Pagoropoulos A, Papakostas N, Mourtzis D, Chryssolouris G. Manufacturing systems complexity review: challenges and outlook. Procedia CIRP. 2012,3(1):644-649.

[3]. Bossaerts P, Murawski C. Computational complexity and human decision-making. Trends in Cognitive Sciences. 2017,21(12):917-929.

[4]. Kulak O, Cebi S, Kahraman C. Review: applications of axiomatic design principles: a literature review. Expert Systems with Applications. 2010,37(9):6705-6717.

[5]. Tao F, Sui F, Liu A, Qi Q, Zhang M, Song B, Guo Z, Lu S, Nee AY. Digital twin-driven product design framework. International Journal of Production Research. 2019,57(12):3935-3953.

[6]. Zhang X, Thomson V. Modelling the development of complex products using a knowledge perspective. Research in Engineering Design, Special Issue: Design and Development Processes. 2019,30(2):203-226.

[7]. McGrath JE, Argote L. Group processes in organizational contexts. Blackwell Handbook of Social Psychology: Group processes. Blackwell Publishers Ltd., Oxford, UK, 2001,603–627.

[8]. Carlile PR. Transferring, translating, and transforming: an integrative framework for managing knowledge across boundaries. Organization Science. 2004,15(5):555-568.

[9]. Granstrand O, Bohlin E, Oskarsson C, Sjöberg N. External technology acquisition in large multi-technology corporations. R&D Management. 1992,22(2):111–134.
[10]. Markham SK, Lee H. Marriage and family therapy in NPD teams: effects of wellness on knowledge sharing and product performance. Journal of Product Innovation Management. 2014,31(6):1291–1311.
[11]. Zhang X. Measures and Mitigation of Complexity during Product Development. PhD thesis, McGill University, Canada, 2017.
[12]. Zhang X, Thomson V. A knowledge-based metric of product complexity. Computers and Industrial Engineering. 2018,115:80–87.
[13]. Ahmad N, Wynn DC, Clarkson PJ. Change impact on a product and its redesign process: a tool for knowledge capture and reuse. Research in Engineering Design. 2013,24(3):219–244.
[14]. Kota S, Ward AC. Functions, structures, and constraints in conceptual design. Proceedings of the Second International Conference on Design Theory and Methodology. Chicago, Illinois. 1990,239–250.
[15]. Bytheway CW. FAST Creativity and Innovation: Rapidly Improving Processes, Product Development and Solving Complex Problems. 2007, J. Ross Publishing, Plantation, FL, USA.
[16]. Bashir HA, Thomson V. Estimating design effort for GE Hydro projects. Computers and Industrial Engineering. 2004,46(2):195–204.
[17]. Oduncuoglu A, Thomson V. Evaluating the risk of change propagation. Proceedings of the 18th International Conference on Engineering Design (Impacting Society through Engineering Design), Lyngby/Copenhagen, Denmark, 2001.
[18]. Hamraz B, Hisarciklilar O, Rahmani K, Wynn .C, Thomson V, Clarkson PJ. Change prediction using interface data. Concurrent Engineering.2013,21(2):141-154.
[19]. Pimmler T, Eppinger S. Integration analysis of product decompositions. ASME International Design Engineering Technical Conferences (Design Theory and Methodology Conference), Minneapolis, MN, US, 1994.
[20]. Wang H, Thomson V, Zhang X. Automation of change propagation analysis using functional analysis and a knowledge perspective.International Journal of Product Lifecycle Management. 2019,12(2):107–130.

Aamir Patel, et. al. “Estimation of Change Propagation Effort Using a Knowledge Perspective.” *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)*, 17(4), 2020, pp. 53-66.