CIRP Annals - Manufacturing Technology 59 (2010) 191–194

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

journal homepage: http://ees.elsevier.com/cirp/default.asp

Method for supporting conflict resolution for efficient PSS development

Y. Shimomura (2)[a,*], T. Hara [b]

[a] Department of System Design, Tokyo Metropolitan University, Asahigaoka 6-6, Hino-shi, Tokyo 191-0065, Japan
[b] Department of Precision Engineering, The University of Tokyo, Tokyo 113-8656, Japan

ARTICLE INFO

Keywords:
Design
CAD
Service

ABSTRACT

A method is proposed to enable service designers to identify existing conflicts in design solutions and to develop basic strategies to solve them. For this purpose, two different approaches for detecting conflicts are introduced: one entails the use of lexical expressions of functions, and the other involves the ranges of design parameters. In the remainder of this paper, the authors describe a detailed process to identify conflicts in design solutions and to develop strategies for resolving them by applying the proposed methods in an actual service case. The proposed method is expected to lead to an improvement in the efficiency of PSS development.

1. Introduction

Recent studies in the field of service design (e.g., [1–4]) have focused on the effective integration of products and services (product/service-system: PSS). However, design is viewed as a form of problem solving, with the intent of satisfying customer demands. Namely, to date, an insufficient number of studies have been conducted on a method to support this design process. In service design, ad hoc design processes, employing trial and error methods, are still dominant, and the quality of a design solution largely depends on a designer's experience and intuition. The authors consider the conflicts present in a design solution to be a key to the creation of quality design. The quality and efficiency of design are largely dependent on how rapidly designers are able to discover and resolve such conflicts.

In this paper, a methodology for supporting design process is proposed. It will enable designers to identify existing conflicts in design solutions and develop the basic strategies required for resolving them computationally.

2. Conflict resolution

TRIZ is a well-known methodology that provides for the detection and resolution of conflicts in the product design field. TRIZ consists of various methods, and a knowledge base grounded on former inventions that enable designers to effectively solve problems. For example, the technical contradiction table [5] suggests basic principles for the resolution of conflicts among elements in a design solution. The principles are based upon the analysis of millions of patents. The use of TRIZ methodologies makes it possible to resolve the conflicts in a product design without the use of ad hoc trial and error. There are a number of interesting studies on the applications of TRIZ to service design. One such study examines the use of TRIZ tools for resolving conflicts in service design [6]. In this study, a set of TRIZ tools comprises the entire conflict resolution process. The proposed conflict resolution process is very useful, in that a simple flowchart of the process with the TRIZ tools is provided. However, the study employs an ambiguous approach to service modeling, treating it in a manner largely dependent on each designer. In this sense, success and failure of the entire conflict resolution process are strongly influenced by those of the service modeling effort.

Another such study suggests the modification of 40 inventive principles of TRIZ for service design [7]. In this study, new principles are proposed by matching the existing principles, and the effectiveness of the two contradictory methods is evaluated. For example, the principle of "Combination–Separation" is described as "Combine parts of an object or the phases of a process to form a uniform object or process. Separate a uniform object or a uniform process to form independent parts or phases". However, in this study, the lack of a defined approach to service modeling makes it difficult to resolve conflicts. In addition, the effectiveness of the new principles is not verified, whereas the principles originally proposed in TRIZ are based upon the analysis of existing patents.

We believe that conflict resolution must be based on a well-defined method of service modeling. Unlike the prior studies reviewed in this section, this paper provides a consistent methodology from service modeling to conflict resolution, and this methodology also enables designers to identify existing conflicts in design solutions comprehensively beyond just a pragmatic application of TRIZ method.

3. Service engineering

Service engineering (SE) [1] aims to provide an engineering methodology for the representation, design, and evaluation of a service. Various sub-models of a service are proposed in SE. However, only one sub-model, referred to as a view model, is detailed herein. In this study, a view model is treated as a target of conflict detection. View models describe a specific manner of

* Corresponding author.

0007-8506/$ – see front matter © 2010 CIRP.
doi:10.1016/j.cirp.2010.03.122

changing a receiver's state. The model describes a functional structure that achieves a change in a receiver's state parameter (RSP) and expresses a portion of the realization structure of a service. This is done via the relationship between the RSP and the functional structure, described through the form of the functional relations among the RSP, content parameter, and channel parameter [1].

The functions of channels and content are expressed by function names, as lexical expressions, and by function parameters (FPs), as target parameters of functions. Each function is related to one or more other functions. The FPs that are directly related to RSPs are recognized as content parameters, and those indirectly influencing RSPs as channel parameters. The lowest functions, which have been sufficiently deployed, are related to entities. An entity is something that exists in the real world; it may refer to a product, person, organization, or piece of software, and will have one or more attribute parameters (APs).

4. An approach to conflict resolution

4.1. Conflict detection using lexical expressions of functions

4.1.1. Basic strategy 1

In this method, conflicts in view models are detected using the lexical expressions of functions. In general, the lexical expression of a function consists of an object and a predicate. Conflict detection is performed through analysis of these elements, from the lexical point of view. For example, the functions "increase staff" and "decrease staff" contain antonymous predicates, but refer to a common object. It is impossible to increase and decrease staff size simultaneously; thus, the conflict between these two functions is apparent. Similarly, the functions "increase staff" and "decrease employees" incorporate antonymous predicates, and refer to lexically disparate, though still synonymous, objects. The conflict between these two functions remains apparent.

4.1.2. Building a Lexical Relation Database

In order to detect conflicts using the above method, frequently used terms, representing synonymous and antonymous relations, must be recorded and made available for comparison with search terms. Thus a Lexical Relation Database that incorporates this information must be established. However, as it is almost impossible to initially accumulate all relevant terms, due to the extremely large number of terms representing such relations, the database must be scalable and provide designers or system administrators with the ability to add new terms. It is largely up to the individual database designer to determine what constitutes an adequate number of terms. However, it is typically sufficient to retain a number equal to that of a commercial thesaurus. The above requirements are developed using a Resource Description Framework (RDF) [8] and a Web Ontology Language (OWL) [9], both of which are employed extensively in research pertaining to the Semantic Web. Their property [8] makes it possible to describe the terms and relationships among them. In addition, two supplemental properties are defined here by the authors: "IsSynonymOf" and "IsAntonymOf", for the identification of synonymous and antonymous relationships, respectively. "IsSynonymOf" is defined to be a transitive property [9]. For example, when the terms "staff" and "employee" present an instance of IsSynonymOf, and the terms "employee" and "worker" also present an instance of IsSynonymOf, the terms "staff" and "worker" are subsequently determined to be an instance of IsSynonymOf.

4.1.3. Process of conflict detection

The lexical expressions of the functions in the design solution (or portions of it) are first compared in a pair wise manner. For each pair of functions, the following process is executed.

1. Determine whether the two functions refer to a common object.

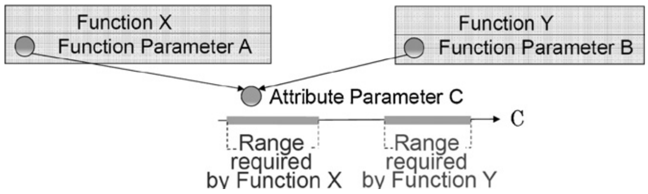

Fig. 1. Inconsistency between parameter ranges.

2. Determine whether the two functions refer to synonymous objects by querying the Lexical Relation Database.
3. Determine whether the two functions have the predicate in an antonymous relation by querying the Lexical Relation Database.
4. Alert a designer to conflict if a conflict is detected.

4.2. Conflict detection using the ranges of design parameters

4.2.1. Basic strategy 2

In this method, the conflicts among functions in view models are detected by using the ranges of design parameters. As an example, consider a portion of a view model where a given parameter influences multiple upper parameters of the constituent functions, as depicted in Fig. 1. The range information of parameter C is used to detect a conflict between functions *X* and *Y*. If there is no overlap between these two ranges, the designer is alerted to the conflict. In such a scenario, the lack of overlap suggests that the design solution requires an infinitely large volume of information to implement. This is based on "The Information Axiom" of Suh's Axiomatic Design [10]. Such an occurrence implies that the service design is infeasible and must be reconsidered.

4.2.2. Range calculation with set-based theory

For each parameter influencing multiple functions, the range required to develop the upper functions must be calculated. Thus, set-based theory [11] is utilized, which formulates parametric operations via a range. In the set-based theory, the variation of each parameter in a system is expressed as a set, which, in turn, is expressed as an interval. The influence of variation in the value of a given parameter, on any other parameter, is calculated by the interval operation rule. The set-based theory consists of four operations, however, only RangeOperation and DomainOperation are employed in the proposed method. RangeOperation calculates the range of an unknown dependent variable from the ranges of multiple independent variables, and DomainOperation calculates the range of an unknown independent variable, from a dependent variable and multiple independent variables. It should be noted that the set-based theory has been employed in prior studies of product design methodology (refer to [12]).

4.2.3. Conflict detection process using set-based theory

In this section, the details of the conflict detection process are provided, along with an example; a portion of a view model (Fig. 2), in which a single parameter influences multiple upper parameters of the functions. For each structure, the conflict detection method is executed as follows.

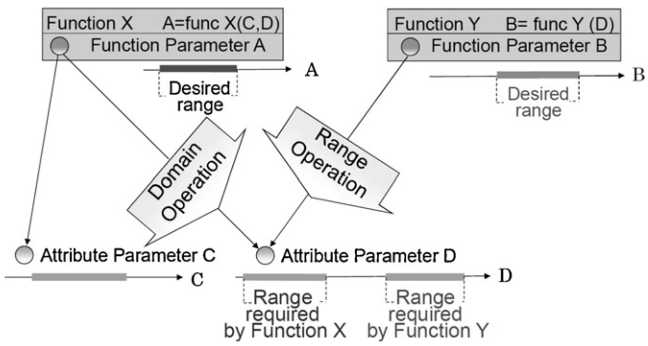

Fig. 2. Detection of inconsistency between parameter.