Article

Recording and reporting a DSR using C-K Design Theory

Sivaramalingam Kirushanth [1,2,*], Boniface Kabaso [2]

[1] Department of Physical Science, Vavuniya Campus of The University of Jaffna, Vavuniya, Sri Lanka; SivaramalingamK@vau.jfn.ac.lk

[2] Department of IT, Cape Peninsula University of Technology, Cape Town, South Africa; KabasoB@cput.ac.za

* Correspondence: SivaramalingamK@vau.jfn.ac.lk;

Abstract: Concept Knowledge (C-K) theory has been used in engineering and science-based research for more than a decade. Design of an Information Technology (IT) artefact is mostly pragmatic in nature. Design Science Research (DSR) methodology applied and studied in many Information Systems (IS) research. Many sub design decisions involved through the design of an IT artefact from a concept (idea) to a working prototype. A DSR artefact is based on a combination of decisions made during several sub-design stages. Artefacts are built based on the selection of elements in each sub-design space. Recording the design decisions on each sub-design space would be beneficial for future researchers. By knowing the design decisions on each sub-design space, researchers would be able to try different combinations of the design. C-K theory provides the ability to capture the design processes' several sub-design spaces. In this paper, we discuss the DSR research methodology by looking at the stages proposed in the literature, and the application of C-K theory in an IT-based DSR. This paper also proposed a C-K theory-based protocol called *Concept Tree* for tracking and reporting artefact design steps. The application of C-K theory in DSR is exhibited using the implementation of the *Concept Tree* for a prototype design IT artefact.

Keywords: C-K Theory; Design Science Research; Design decisions; IT artefacts; WIM system design

1. Introduction

Basic and Applied are the two main classifications of research in literature. The basic research is more towards deriving a generalized solution than finding a solution for a real-world problem, the quest for fundamental understanding is high, but the consideration of use is low in basic research. The applied research tries solving problems in context by providing innovative solutions which are better than the existing solutions. The quest for understanding the fundamentals is low, and the consideration of use is high in applied research [1]. Use-inspired basic research uses pure research findings and theories in practices to find an appropriate solution. The engineering research is more towards applied. In literature, designing an IT artefact is classified under engineering research [2].

In literature, DSR is classified in as methodology [3] and strategy [4] . The literature on research methodologies shows that research involved in solving a real-world problem with the design of an artefact is design research, also known as design science research. Accreditation Board of Engineering and Technology (ABET) defines engineering design as the process of devising a system, component, or a process to meet the desired needs. They also classify engineering design as an iterative decision-making process, in which the resources are optimally converted to meet the desired objective. The basic knowledge of sciences, mathematics, and engineering are applied during the conversion of such resources [5].

DSR is a domain-independent research strategy [4], predominantly focused on developing knowledge on generic actions, processes and systems. DSR aims at the improvements based on the thorough understanding of problems or opportunities. There is no specific methods or fixed rules to follow during the DSR. The operationalization of the DSR strategy can be done in many ways [4].

DSR generates new and innovative artefacts [6]. Hevner et al. (2004) in their seminal paper on DSR, used the four types of DSR artefact suggested by March and Smith (1995) such as constructs, models, methods, and instantiations. Research articles in IT and IS using DSR method/strategy focus discussing a specific type of DSR artefact. The system design is the most common DSR artefact type discussed in many research papers in literature. In addition to System designs, Ontologies, Taxonomies, and Frameworks are very specific models frequently proposed in DSR [9]. Unlike engineering, IT, and IS research articles fail to discuss the prototype (instantiations) design process in detail. Capturing and reporting the design and development of a prototype IT artefact is lacking in the literature. We could view the process involved in the design and development of a prototype IT artefact as engineering design. Engineering design of a device or a system could be an upgrade (evolutionary change), or an innovation (new artefact). In the upgrade, the evolution of a product with a significant improvement occurs over a period. Usually, the upgrade is done when there is no competition. Since upgrading does not need a different idea, the creative capacity of the designer is limited. In contrast, innovation takes place when there is a need for a new product. The rapid growth in science and technology and the competition between companies draw the necessity for new innovative products.

A unified design theory called C-K theory was introduced by Hatchuel in 1996. One of the main contributions of C-K theory consists in clarifying tasks and teamworking in upstream or innovative design phases [10]. C-K theory captures the Knowledge flow from an Idea development (conceptualization) to the implementation of a prototype (instantiation). In this paper, we discuss the use of C-K design theory in DSR. Application of C-K theory is presented using an example of prototyping research. Further, we propose a new reporting protocol called *Concept Tree* to list the most important design decisions made during the prototype development research. In this research, we designed and developed a new concept tree for the design of the prototype WIM system.

2. The Prototype WIM System

Identifying overloaded vehicles in any road segment would be beneficial for the transport industries. Telematics data are becoming richer and more accurate due to the technological improvements in sensors and connectivity. The Internet of Vehicles (IoV) is becoming popular as a result of this. Use of Machine Learning (ML) and Vehicular Telematics (VT) data could be used to explore more about the road, vehicle, and driver [11]. An omnipresent WIM system capable of monitoring the payload of the vehicle in any road segment could help the transport industry in many ways. In [12], we proposed an approach for a new WIM system using the VT data and ML. We discussed the prototype design and development of a new WIM system using the proposed approach. The prototype system (see Figure 1) has the three main subsystems data collection (VT module), data (stream) handling backend (VT data ingestion module), and the ML module. A scalable cloud-native ML backend was developed. Android smartphone linked to the OBD-II Bluetooth dongle was used as the VT data collection device.

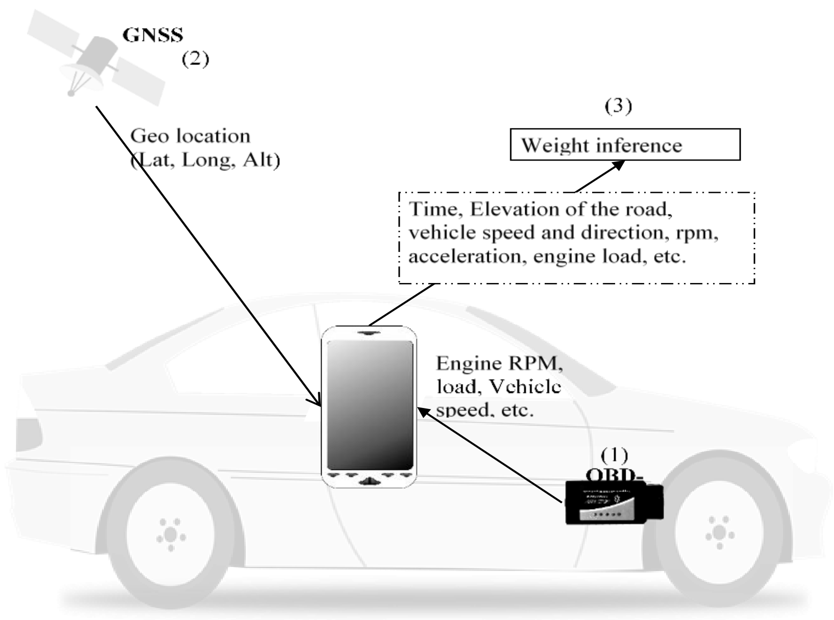

Figure 1: Developed WIM System Prototype

In this paper we discuss the prototype design, discussing the research methodology, and introduce C-K theory, our proposed simple DSR reporting tree called Concept Tree, and finally discuss the prototype design using the proposed Concept Tree.

3. Design Science Research Methodology

DSR is the design and investigation of artefact in context [13]. The DSR changes the state-of-the world through the finding of new artefacts. In our DSR, we design/develop an artefact with the aim to test the concept of a new WIM solution and advance our knowledge about the characteristics of these artefacts and the processes to design and develop them. Figure 2 shows the self-descriptive diagram explaining the process steps in DSR and the outputs of each step and the flow of knowledge by Vaishnavi and Kuechler [14]. They discussed five main DSR process steps. The awareness of the problem starts with the existing knowledge form the experience and the literature. New ideas and new concepts become suggestions to solve the problems identified. The ideas are then developed to produce an artefact. This development phase is an iterative process, since the initial concept would be mostly abstract, and the specification may change during the design development process. The final artefact is then tested in context. The conclusion is the acceptability of the design suggestion based on its test performance.

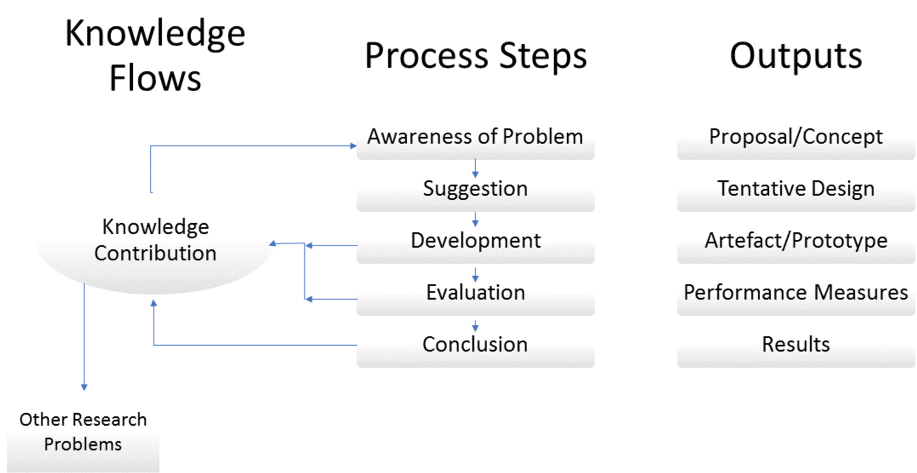

Figure 2: DSR Process Model adopted from [14]

A research process is the application of scientific methods to a complex task of discovering solutions to the problems [15]. Khandani (2005) listed five steps for solving design problems,1) Define the problem, 2) Gather pertinent information, 3) Generate multiple solutions, 4) Analyze and select a solution, 5) Test and implement the solution. Several other steps are also proposed in the literature. Offermann et al. (2009) subdivided the DSR process into three main phases, Problem Identification, Solution Design, and Evaluation. Table 1 lists some design science research processes discussed by different researchers and our research processes in each phase.

Table 1: DSR Processes adapted from [17]

	Takeda, Veerkamp and Yoshikawa, 1990 [18]	Nunamaker, Chen and Purdin, 1990 [15]	March and Smith, 1995 [19]	Vaishnavi and Kuechler, 2004 [14]	Peffers et al., 2008 [20]	Offermann et al., 2009 [17]	Our Approach
Problem	• Enumeration of problems	• Construct a Conceptual Framework (3)		• Awareness of Problem	• Problem identification and motivation • Define the objectives for a solution	• Identify the problem (2) • Literature research (1) • Expert interviews • Pre-evaluate relevance	1,2,3
Solution	• Suggestion Development	• Develop a System Architecture (1) • Analyze & Design the System • Build the System	• Build (3)	• Suggestion • Development	• Design and development	• Design artefact (2) • Literature research	1,2,3
Evaluation	• Evaluation to confirm the solution • The decision on a solution to be adopted	• Observe & Evaluate the System	• Evaluate	• Evaluation • Conclusion	• Demonstration (1) • Evaluation	• Refine hypothesis • Expert survey • Laboratory experiment(2) • Case study / action research • Summarize result	1,2 in a context

Our DSR is a combination of processes specified by Nunamaker et al. (1990), March & Smith (1995), Peffers et al. (2008), and Offermann et al. (2009). Which is as follows:

A. Problem identification
    1. Literature research
    2. Identify problem
    3. Construct a conceptual framework
B. Solution design
    1. Develop a system architecture
    2. Design artefact

3. Build

C. Evaluation

1. Laboratory experiment
2. Evaluate

In the problem identification phase, we have done a systematic literature review (SLR) [11] to identify the gaps in the use of Intelligent Transport System (ITS) and identified the problem in transportation, then a conceptual framework was proposed in [12]. In the solution design phase, we developed the proposed system architecture, design the solution artefact as a prototype, built it. Finally, in the evaluation phase, we evaluated and demonstrated the artefact in a context-specific context. In this paper, we discuss the reporting of design solution phase.

Unlike other research, a design-oriented research such as DSR often begins with an abstract, vague idea (concept) from the researchers' mind. Structuring an analysis is reported much easier than formulating a well-structured definition of a design problem may evolve through a series of steps or processes (phases) as we develop a complete understanding of the design problem [16].

Takeda et al. [18], used three factors which are prerequisites to describe a design process, 1) required specification (Ds), 2) design solutions (S), and 3) knowledge (K) . They described the deductive design process logically as:

$$S \cup K \vdash Ds, \qquad (1)$$

Where, the design solutions, Ds, are derived from the specification, S, and the knowledge K. In the same paper, Takeda et al. discussed the abductive design process as:

$$Ds \cup K \vdash S, \qquad (2)$$

Where the design solutions and the knowledge about the design objects can be used to derive the design specifications, Abductive design process is considered an incremental process, in which the refinement of the design object takes place on each step of the abductive design process.

The deduction is used in obtaining the properties of the current solution with respect to the existing knowledge. Given the current design solution, Ds, design knowledge, Ko, and the required properties, P, as the specification are by:

$$Ds \cup Ko \vdash P, \qquad (3)$$

At each step of the design process, the deduction is applied to obtain all the properties of the current solution with respect to the currently available knowledge. This is to know the properties the current solution has, and check whether the current solution satisfies the given specification and knowledge [21].

The design research is incremental and flexible where the requirements and the views may change over time. Whereat each step, the solution (concept) may change based on our experience (knowledge) on specifications within a context. This changing nature of the design process adds extra knowledge. There must be a way of capturing these knowledge expansions in design science research. Several engineering research report in literature carried to convey the knowledge expansion process during the design using C-K theory.

4. C-K Design Theory

In 1996 Hatchuel drafted and in the early 2000s with Weli introduced a unified Design theory called Concept-Knowledge (C-K) theory. "The name 'C-K theory' mirrors the assumption that Design can be modelled as the interplay between two interdependent spaces having different structures and logics: the space of concepts (C) and the space of knowledge (K). The structures of these two spaces determine the core propositions of C-K theory" [22].

The knowledge space, K, includes all established propositions which are true. The K space holds available knowledge such as scientific and engineering models and facts, physical laws.

Concept space contains the vague concepts (ideas) which are un-decidable (neither true nor false) about some partially unknown set of objects x called C-set. Concept space, C, corresponds to the incomplete description of objects. The partial description of objects in C space captures the notion of design briefs or broad specifications. In essence, concept space C holds two sets, 1) pragmatic notion of brief or broad specifications we find in innovative design, 2) unusual set of objects x.

Concepts are propositions of the form : There exists some object x, for which a group of properties p1,p2,..pk hold in K. All elements building the propositions in C come from K but not belong to K [22].

The design process can be described then by the interaction and co-evaluation of these two spaces C and K through the application of four types of operators; C→K, K→C, K→K, and C→C (described below). Design partitions the sub-concepts of C space by adding or deleting properties arose from K space. There are two kinds of partitioning, 1) Restrictive, 2) Expansive. In restrictive partitioning, the designer adds the usual properties of the object. In expansive partitioning, they add new properties (sub-concepts). The partitioning of a concept may result in an expansion of K space. This could happen due to the learning of new knowledge to pursue the creative expansion of C space or the experience from one concept design phase [23]. The feasibility of the available objects in C space cannot be determined with the available knowledge. The design process tends to expand the C and K spaces simultaneously. On one side of the expansion, new creative concepts, on the other side, new learning knowledge allowing the realism of concepts. Design ends when the properties introduced into the concept can be validated in K space; that is, it can be confirmed in K that such an object may exist [22,24]. It is shown that the C-K theory is sufficient to describe the generation of new objects and new knowledge, which are distinctive features of design [25]. Figure 3 shows the C-K diagram presented in [22]. The concept C0 from existing knowledge K1 is used to test the new concept C1. A new knowledge K2 is added after C1, C2 will be formed and tested using K2. The design phase continues until the end of the building and testing the artefact.

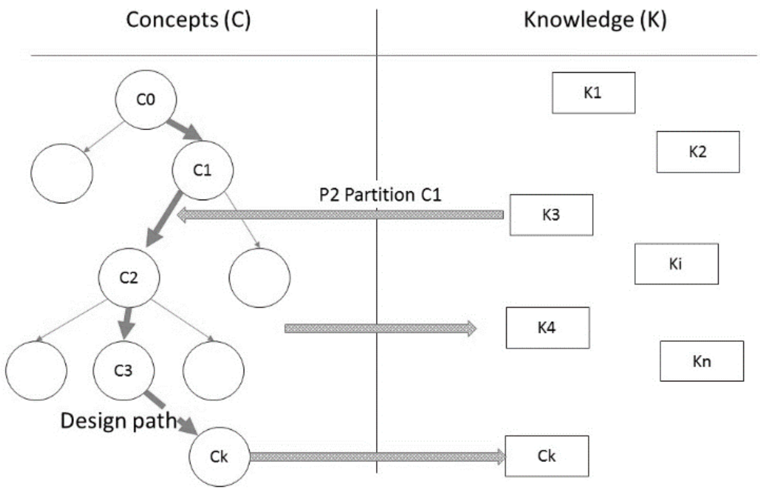

Figure 3: C-K Diagram adopted from [22]

4.1. The C-K Design Square

The design process always tries to find a better solution for the given specification in context. This helps the two spaces, C, and K, to expand during the course of the design. As mentioned earlier, in design, each space helps the other to expand. According to Hatchuel et al. [10], the design process is nothing more than the operations that allow these two spaces to expand. There are necessarily four different kinds of operators classified under two, internal, and external. Where, C→K, K→C are classified external, and C→C, K→K are classified internal. Figure 4 shows the C-K design square of four operators by [10]. In the design process, a concept may generate another concept, or it may transform into knowledge. Design process always seeks to expand the concept space (ΔC), with existing knowledge (Ko) through disjunctive (K→C) operators. Also expands Knowledge space (ΔK) with existing concepts (C) using conjunctive operators (C→K) [26].

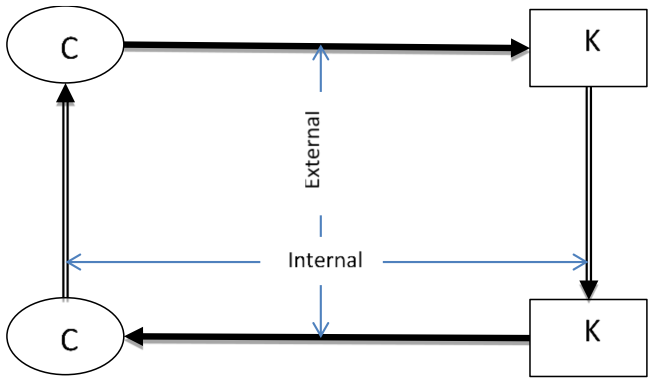

Figure 4: The C-K Design Square [10]

4.1.1. External Operators

K→C : Here, the properties will be added or subtracted from K to concepts in C. This operator creates disjunctions when it transforms a proposition into a concept. Which usually generate new alternatives, these alternatives are not concepts but potential seeds for alternatives. This operator expands the space C with elements form space K [27]. This is performed at every stage where we come up with a possible set of solutions. For example, in this research, there were two possibilities to use for collecting data, custom-built black-box device, and using a smartphone with OBD-II adaptor. Those possible alternative sub-solutions (Cn*(x)) were from the knowledge from the literature search.

C→K: This operator seeks for properties in K that could be added or subtracted to reach propositions with logical status. The validity of the alternative during C→K operation contributes to knowledge. It creates *conjunctions* which could be accepted as finished designs. The validation of a design concept by doing a test, prototype, a mock-up are examples of C→K operator. This operator expands knowledge with the help of concepts [27]. For example, in this research, we had several ML algorithms as alternatives to infer the vehicle weight. Neural Network, Bayesian, Decision Trees, Linear Regression were some of the alternative algorithms for the inference. Testing of each algorithm within our context leads us to select one best algorithm (Linear Regression) which performed better than the rest of the possible set. This updated our understanding of the behaviour of several candidates the algorithms in context.

4.1.2. Internal Operators

C→C: This operator is at least the classical rules in set theory that control partition of inclusion [27]. As shown in Figure 2, a new partition or branch of concept (sub-concept, Cn*) will be added when we test a concept and accept it (Cn). The new branch will be possibly partitioned if necessary. In general, a design solution is an artefact. The artefact is a combination of several sub-modules. If we consider given examples for external operators above, the overall artefact is an eco-system of data collection devices and the ML backend. After choosing the smartphone with OBD-II adaptor as the data collection device (in K→ C), we tested it to verify by collecting some sample data. This confirmed the selection of the data collection device (C→K). Next, we moved to a new branch (partition, C→C), focusing on the selection of ML algorithms for the backend.

K→K: This operator is at least the classical rules of logic and propositional calculus that allow a knowledge space to have a self-expansion. Proving new theorems, generalizing and formulating new hypothesis are some of the activities of this operator. This would be based on the expansion of existing knowledge to new knowledge (experience) from the development of the design artefact.

Hatchuel & Weil in [22], documented two major benefits of C-K design theory in real research and development (R&D), 1) better control of the design rationale, 2) an increase of the innovative power of the design work. The second benefit usually implies the first one. Shifting the research direction during the R&D process is common in design research. For example, in our DSR project, the research approaches and directions were shifted at different stages during development. Hatchuel & Weil stated that such shifts appeared easily understandable with C-K theory because they were the joint consequences of both concept and knowledge expansions.

Hatchuel et al. in their recent paper in 2017 concluded that "C-K theory appears today as a solid scientific ground for a transdisciplinary shift. Creative processes are better understood and modelled within Design theory and science. Then, such new science can contribute to research on human activities that were already seen as creative; it can also help to study creative forms in domains where they are less visible or hidden. Finally, creative thinking is no more reduced to a psychological and natural phenomenon; it reveals a forgotten class of scientific thinking, the generic design of unknown objects and its co expansion with the transformation of knowledge. Through the formalization of C-K theory, such a paradigmatic shift has already opened new ways of research and provided unexpected findings. Yet, all this could be only the early steps of a much wider scientific impact" [26].

4.2. Use of C-K Theory

Application of C-K theory in Science-based research was illustrated using the design and development of a new combustion system for Martian spacecraft. The paper [10], explained the use of C-K theory in the prototype design of the mars rocket project by the European Space Agency (ESA). The same example was discussed in papers [25,28,29]. An industrial application of C-K theory was discussed in [29]. The authors discussed the design of new bio-climate in cars. In that paper, they discussed three main factors, how C-K modelling accounts the explorations in a specific industrial situation, how C-K theory helps to understand the main design spaces, and how it enables to monitor the exploration process. Use of C-K theory in IT-based DSR is not visible in literature.

4.2.1. C-K in IT-based DSR

The phenomena studied in IT research are artefacts that are designed and built by a human to achieve the purpose of human needs [8]. Implications for IT research are threefold. First, there may not be an underlying deep structure to support an IT theory. Out theories may need, instead, to be based on theories of the natural phenomena (i.e., human) that are impacted by the technology. Second, our artefacts are perishable. Hence our research results are perishable. As needs change, the artefacts produced to meet those needs also change. For example, a theory of how programmers use a currently defunct language. Third, the IT artefacts are produced at an ever-increasing rate, resulting in numerous phenomena to study. Explicating and evaluating IT artefacts (such as constructs, models, methods, and instantiations) will facilitate their categorization so that research efforts will not be wasted building and studying artefacts that have already been built and studied [8]. Unlike engineering and science research, the use of C-K theory in IT-based research is not prominent in the literature. The use of C-K theory in IT-based DSR is discussed using the WIM system prototype mentioned earlier.

The artefact was developed with the aim of testing our idea. The developed artefact was then used to draw our cased-based conclusion. The development phase of the artefact comes under the design and engineering cycles of the design science research. This phase was carried out by the sequence of design cycles to refine the global design. Each design phase has its own reasoning type. According to Lu and Liu [30], a design process consists of three reasonings, inductive, deductive, and abductive. In the inductive type of reasoning, we come to a general conclusion from a specific observation. Similarly, in deductive reasoning, we make our conclusion from applicable rules. But, in abductive reasoning, we hypothesize based on some incomplete or smaller set of observations. Additionally, to minimize the logical extension and to avoid dealing with exceptions, the Circumscription [31] is used. "Circumscription is a type of commonsense reasoning and has been developed to deal with *exceptions.* In circumscription, exceptions for a given context can be determined by minimizing logical extensions of the predicates which represent *abnormality* with keeping the whole context consistent. Here abnormality is the implicit description of each piece of knowledge" [21]. Figure 5 shows a diagram of the design research processes, and the inferencing used at each step with the C-K operation on each step.

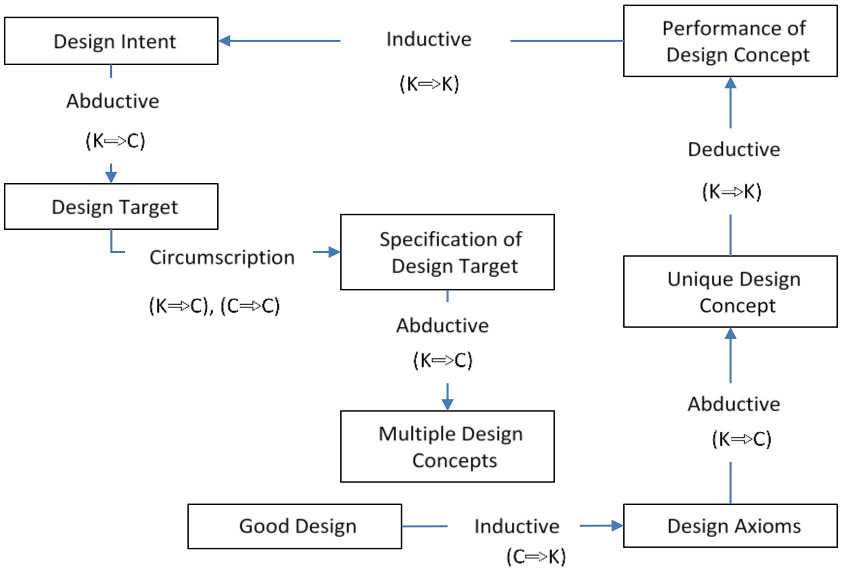

Figure 5: Reasoning in Design adapted from [30]

The relationship between the design science, knowledge and social context of the developed WIM system is shown in Figure 6. According to Wieringa [13], the design science research tries to solve a problem in society, by designing and deploying an artefact in the context. The design phase uses existing background knowledge, and knowledge gets updated over time. The C-K theory was adopted in this research as a framework to track and record the concept of design development. We also proposed and used a slightly different method for recording the design process.

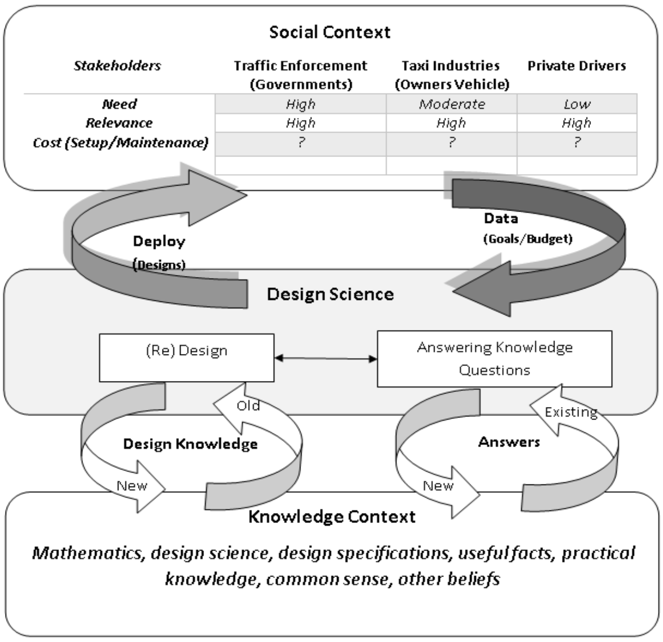

Figure 6: Design Science Research for Social Context using Knowledge Context adopted from [13]

We can view the final C-K tree as a Depth First Spanning Tree, where the levels are of the design spaces, and the nodes are the concepts in each design space. The C-K theory was used as a skeleton in the design portion of our research. During the design, the concepts were designed with existing knowledge and the knowledge was updated incrementally.

5. Concept Tree for recording design

In here we introduce a new method to represent design process, called Concept Tree. The notions we used are based on original C-K theory. We predominantly focus on the C space (concepts/ideas) in our C-K design recording method. The model graph of our proposed design recording is given in Figure 7.

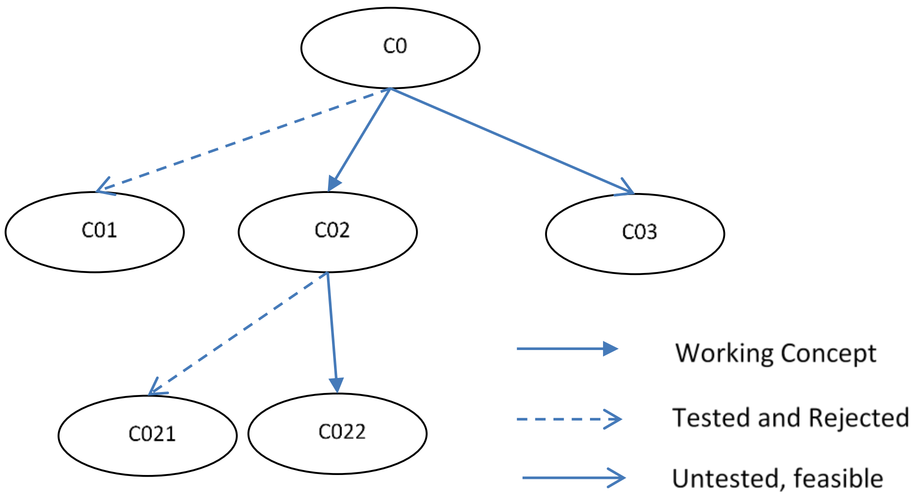

Figure 7: Recording design concepts (Concept Tree building)

Our proposed method contains the following rules:

- **All sibling nodes were the possible design solutions at that level (design space):**

Each new design space (level) starts from its predecessor. Thus, the design concepts are restricted to its predecessor.

- **The concepts must be tested from Left to right:**

Adaptation (tried/tested/implemented) concepts must be reported from left to right. This would allow designers to share their ideas which they tried first. They can move to the next possible idea if the previous failed and choose the next best working concept.

- **Three types of arcs (arrows):**

Depending on the implementation/testing, the concept, the possible alternative concepts may become unusable or useable. A concept is said to be unimplemented when we do not try to test or implement it. There are possibilities where we generate new ideas but only pick one (try) to implement (a common pragmatist approach) and forward to the next design space. The untested concept opens space for future research. The three different node types are addressed by a different type of arrows in the graph (see Figure 7).

- **Two classes of Nodes:**

The circles in the C-K graph represent concepts in general. But the initial concept, C0, is the root node of the tree which has propositions which are just vague statements. These propositions add the boundary to the thinking of the designer. This restricts the size of the possible design solutions at the next level. Each level is a new design space. The nodes (not C0) in each design space is an alternative candidate solution idea based on its predecessor.

5. Results and Discussion

The use of C-K design theory in the design and development phase is briefly discussed here. Figure 8 is the simplest representation of our research using the C-K theory. The tree is built starting from the initial concept C0 (Faster, Ubiquitous, Cost-effective, WIM System), by keeping the constraints (propositions) at the top level. The tree grows by listing the possible candidate solutions for the next level of implementation of the current selection. The concept tree cannot be created at the early stage of the research. But the Tree will start growing at each stage of design and development where we make important design decisions. These stages form new levels, i.e. design spaces.

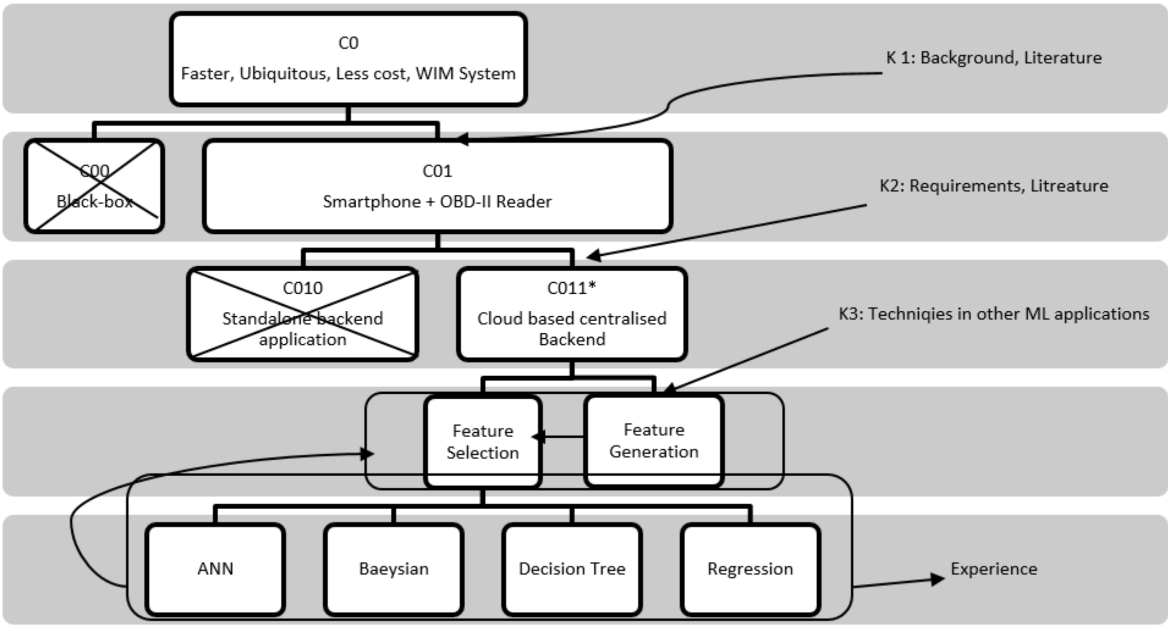

Figure 8: Representation of this research using typical C-K design diagram

In our research, we have recorded the following main stages and their substages:

Level 1. Data collection technique
1.1. Using a smartphone and OBD adaptor
1.2. Using an existing black-box device

Level 2. Backend design
2.1. Using a cloud-native backend application
2.1.1. Platform
2.1.1.1. Kubernetes
2.1.2. Language
2.1.2.1. Golang
2.1.3. DBMS
2.1.3.1. Cassandra
2.1.4. Stream processing
2.1.4.1. Kafka Cluster + Akka Streams
2.2. Using a simple backend application

Level 3. Feature Generation
3.1. Using the existing features as it is
3.2. Feature crossing
3.3. Applying various non-linear functions
3.4. Combining 3.2 and 3.3

Level 4. Feature selection
4.1. Choose all features
4.2. Use feature selection techniques

Level 5. ML algorithm
5.1. Bayesian Regression
5.2. Decision Tree
5.3. Neural Network
5.4. Linear Regression

Figure 9 shows the simple Concept Tree of our research. Unlike other nodes, selection of node in Level 3 is transitively depended on level 5. The creation of new features is depended on the performance of the ML algorithm. The ML model in Level 5 is determined by the selected features in Level 4; in the meantime, Level 4 is determined by the node in Level 3. The flow from level 3 to 5 was iterative until we selected a better ML algorithm (5.4) in Level 5. Different feature engineering (Levels 3 and 4) techniques were used iteratively. Finally, we stopped iterating after reaching a decent inference accuracy. Here we have used our Concept Tree to discuss the decisions made during the major sub-design concept spaces.

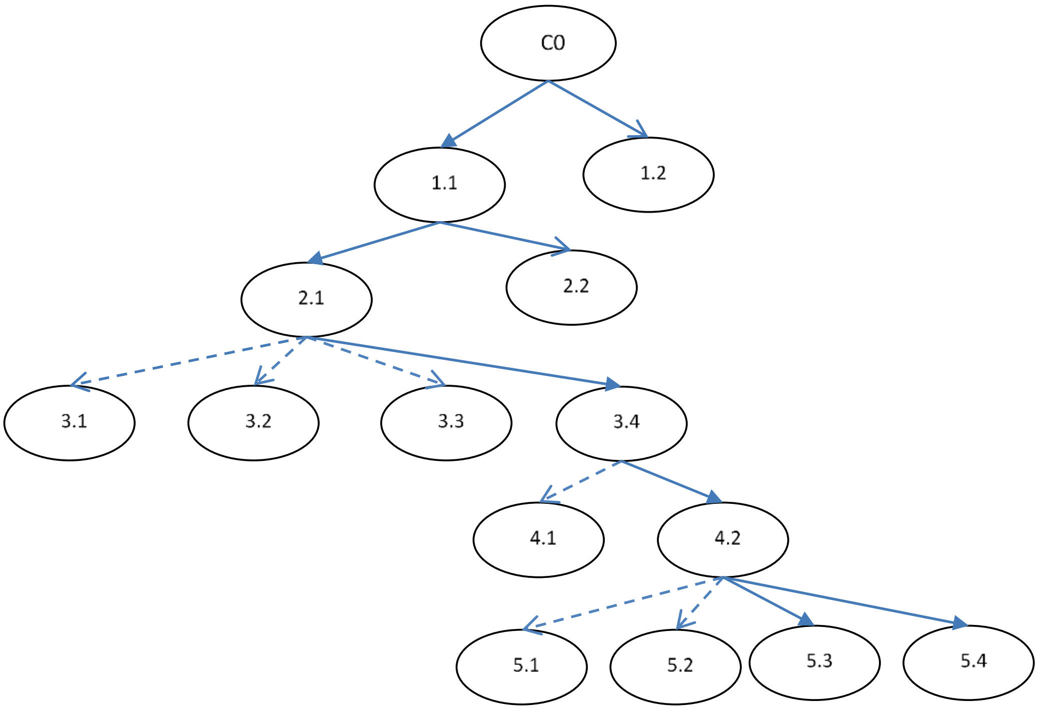

Figure 9: Concept Tree of new WIM system development

By looking at this concept tree, a researcher could understand the decisions made in each sub-concept space. This would enable a future researcher to choose a different design decision at any stage and continue the research. The adaptability of the Concept Tree in IT or IS DSR, producing instantiation type of artefact still needs to be researched. There must be more DSR to be tried to report their design phase of an artefact using Concept Tree. The main drawback of this Concept tree is that this could grow bigger, thus becomes unreadable. If there are many sub-concepts (nodes), there should be a mechanism to shrink the Concept Tree by grouping sub-concepts. It would not be able to report each sub design spaces in detail. It can only be drawn after the completion of the design. Unlike C-K theory, Concept Tree only captures the concept and sub-concepts in the design, not the Knowledge part of it.

6. Conclusions

In this paper, we have discussed the common practices in DSR research. Majority of the articles produced from DSR were not focused on reporting the design of the prototype as a DSR artefact, the use and appropriateness of C-K design theory in the context of a DSR producing IT-based prototype was discussed. The appropriateness of the system was discussed using model research carried to produce a prototype WIM system using VT and ML. We proposed a new DSR recording strategy named Concept Tree based on C-K theory. This paper contributes to the application of C-K theory in DSR using the implementation of Concept Tree with an example. The example used in this paper was a prototype design of a new WIM approach using VT and ML. Design decisions in each stage were captured using the Concept Tree. The adaptability of Concept Tree for a range of DSR still needs to be researched.

Author Contributions: Conceptualization, B.K. and S.K.; Methodology, S.K.; investigation, S.K, B.K.; resources, S.K., B.K; writing—original draft preparation, S.K.; writing—review and editing, B.K.; supervision, B.K.; project administration, B.K.; funding acquisition, B.K.

Funding: Please add: This research was partly funded from Cape Peninsula University of Technology (CPUT), University Research Fund (URF) and University Grants Commission (UGC) Sri Lanka Postgraduate Grant Scheme. The APC was funded by CPUT.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Donald E. Stokes *Pasteur's Quadrant*; Brookings Institution Press, 1997; ISBN 9780815781776.
2. Weber, S. Design Science Research: Paradigm or Approach? Design Science Research: Paradigm or Approach? In Proceedings of the AMCIS 2010 Proceedings; 2010.
3. Geerts, G.L. A design science research methodology and its application to accounting information systems research. *Int. J. Account. Inf. Syst.* **2011**, *12*, 142–151.
4. Joan, V.A.; Aravind, C.; Joop, H. Conducting and publishing design science research: Inaugural essay of the design science department of the Journal of Operations Management. *J. Oper. Manag.* **2016**, *47–48*, 1–8.
5. Haik, Y.; Shahin, T. *Engineering Design Process Second Edition Yousef Haik Tamer Shahin*; 2014; ISBN 9780495668145.
6. Venable, J. Incorporating Design Science Research and Critical Research Into an Introductory Business Research Methods Course. *Electron. J. Bus. Res. Methods* **2011**, *9*, 119–129.
7. Hevner; March; Park; Ram Design Science in Information Systems Research. *MIS Q.* **2004**.
8. March, S.T.; Gerald, F.S. Design and natural science research on information technology Salvatore. *Decis. Support Syst.* **1995**, *15*, 251–266.
9. Offermann, P.; Blom, S.; Schönherr, M.; Bub, U. Artifact types in information systems design science - A literature review. In Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); 2010.
10. Hatchuel; Masson; Weil C-K Theory in Practice : Lessons From Industrial Applications. *Des. Conf. 2004* **2004**, 245–258.
11. Kirushanth, S.; Kabaso, B. Telematics and Road Safety. In Proceedings of the 2nd International Conference on Telematics and Future Generation Networks; IEEE, 2018; pp. 103–108.
12. Kirushnath, S.; Kabaso, B. Weigh-in-motion Using Machine Learning and Telematics. In Proceedings of the 2nd International Conference on Telematics and Future Generation Networks; IEEE, 2018; pp. 115–120.
13. Wieringa, P.R. *Design science research in information systems and software systems engineering Research methodology accross the disciplines*; 2016;
14. Vaishnavi, V.; Kuechler, B. Design Science Research in Information Systems Available online: http://desrist.org/desrist/content/design-science-research-in-information-systems.pdf (accessed on Dec 21, 2017).

15. Nunamaker, J.F.; Chen, M.; Purdin, T.D.M. Systems Development in Information Systems Research. *J. Manag. Inf. Syst.* **1990**, *7*, 89–106.
16. Khandani, S. *Engineering Design Process : Education Transfer Plan*; 2005;
17. Offermann, P.; Levina, O.; Schönherr, M.; Bub, U. Outline of a design science research process. **2009**, 1.
18. Takeda, H.; Veerkamp, P.; Yoshikawa, H. Modeling Design Process. *AI Mag.* **1990**, *11*, 37–37.
19. March, S.T.; Smith, G.F. Design and natural science research on information technology. *Decis. Support Syst.* **1995**, *15*, 251–266.
20. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *J. Manag. Inf. Syst.* **2008**, *24*, 45–77.
21. Takeda, H. Abduction for design. *Form. Des. Methods CAD* **1994**.
22. Hatchuel, A.; Weil, B. Design as forcing: Deepening the foundations of C-K theory. In Proceedings of the 16th International Conferenece on Engineering Design; 2007; pp. 1–12.
23. Hatchuel, A.; Weil, B. C-K theory: Notions and applications of a unified design theory. In Proceedings of the Proceedings of the Herbert Simon International Conference on « Design Sciences »; 2002; pp. 1–22.
24. Kazakci, A.A.; Hatchuel, A.; Weil, B. A model of CK design theory based on term logic: a formal CK background for a class of design assistants. *Proc. Des. 2008, 10th Int. Des. Conf. Dubrovnik, Croat.* **2008**, 43–52.
25. Hatchuel, A.; Weil, B. C-K design theory: An advanced formulation. *Res. Eng. Des.* **2009**, *19*, 181–192.
26. Hatchuel, A.; Masson, P.; Weil, B.; Hatchuel, A.; Masson, P.; Weil, B.; Modelling, C.T. *C-K THEORY : MODELLING CREATIVE THINKING AND ITS IMPACT ON RESEARCH*; 2017;
27. Armand Hatchuel; B Weil A New Approach of Innovative Design: an Introduction To C-K theory. *Int. Conf. Eng. Des.* **2003**, 1–15.
28. Hatchuel, A.; Le Masson, P.; Weil, B.; Hatchuel; Masson; Weil C-K Theory in Practice : Lessons From Industrial Applications. *Des. Conf. 2004* **2004**, 245–258.
29. HATCHUEL, A.; MASSON, P. LE; WEIL, B. THE DESIGN OF SCIENCE BASED PRODUCTS: AN INTERPRETATION AND MODELLING WITH C-K THEORY. In Proceedings of the INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006; 2006.
30. Lu, S.C.-Y.; Liu, A. Abductive reasoning for design synthesis. *CIRP Ann.* **2012**, *61*, 143–146.
31. McCarthy, J. Circumscription—A form of non-monotonic reasoning. *Artif. Intell.* **1980**, *13*, 27–39.