

## Refinement on Data, Information and Knowledge Lifecycles towards Service Economics Planning to Services Implementation

Yucong Duan<sup>1</sup>, Gongzhu Hu<sup>2</sup>, Xiaobing Sun<sup>3</sup>, Zhaoxin Lin<sup>4</sup>, Qiang Duan<sup>5</sup>

<sup>1</sup> College of Information Science and Technology, Hainan University  
Haikou, 570228, China

E-mail: duanyucong@hotmail.com

<sup>2</sup> Central Michigan University, Michigan, USA.

E-mail: hu1g@cmich.edu

<sup>3</sup> Yangzhou University, Yangzhou, China

Email: sundomore@163.com

<sup>4</sup> School of business, Iowa State University, USA

E-mail: zxlin@iastate.edu

<sup>5</sup> Pennsylvania State University, Abington PA, USA

E-mail: qxd2@psu.edu

### Abstract

Service Economics has been successfully proposed and implemented for promoting the macro service market especially in Global value chains which however don't provide guidance to and integrate directly with the IT side implementation to achieve coherent information coordination and robust adaptability which indicates maximization of business profitability. Therefore there leaves a gap of the refinement from the strategic business planning to ubiquitous services implementation in the API (Application Program Interface) economy. Recently Gio Wiederhold attributed this gap to the knowledge difference between economists and the IT professionals and proposed that IT professionals should act to fill the gap since they have irreplaceable data, information and knowledge concerning the specific implementation. As an instance of this guidance in the practice of Value Driven Design, we proposed a systemic formalization from the value calculation to the design quality measurement which binds the modification and change on the design artifacts with the business value strategy through a framework of managed quality properties in a service design process covering data, information and knowledge assets.

*Keywords:* Software Economics, Service Economics, Ubiquitous Service Design, Model Evaluation, Value Driven Design

### 1. Introduction

Boehm and Sullivan<sup>3</sup> have proposed that the essence of design is motivated by creating value added for maximizing the profit of stakeholders.

They proposed to guarantee the profit goal through a top down manner implementation from the strategic economic reasoning down to the specific development processes which include modeling activities and model evaluation. This integration between

business planning and IT implementation<sup>13</sup> will bring coherent base on data, information and knowledge interplays<sup>33</sup> in terms of enhanced data sharing, information exchanging and knowledge creation between business and technical domains to maximize the general profitability of all stakeholders. However there are many shortfalls in the state of the art practice and research which hinder the completing of the seamless implementation of the software economics goal.

Currently Service Economy<sup>22</sup> especially Software-as-a-Service economics<sup>20</sup> has been successfully proposed and implemented for promoting the macro service market especially in Global value chains which is becoming more dependent on efficiency of service construction on various "as a Service (aaS)"<sup>14</sup> and effective service delivery/brokerage<sup>12</sup>. To get the most value added from the efficiency improvement and strategic investment practice on the service economy with improved economic outcomes, we need to construct a seamless data, information and knowledge flow circle which goes through the economic planning to the design details to guarantee the refinement of the business consideration as decision making, design activities or resource allocations and model evaluation. Retrospectively we also need a counter direction feed back data, information and knowledge path to support the measure of the performance of the ongoing design process for further control planning and improvement. However there is few material which can be found from the software economics area<sup>16</sup> to construct this bidirectional data/information flow and control flow following the Value Driven ideology<sup>6</sup>. There are even fewer works which can be found to directly support the uni-formally modeling and evaluation/calculation of the service value<sup>10</sup> crossing business and information technology domains although materials such as service contracts have provided rich content for constructing the evaluation<sup>9</sup>. Most existing work focuses on empirical knowledge and statistical models<sup>21,27,28</sup> which are effective for project management and related to general software quality properties but not possible to related detail enough to specific design activities and model qualities. These approaches cannot avoid the

challenge of the complexity of the interleaving of the quality properties<sup>26</sup> which is related to specific design activities and changes constantly for different projects and inside a development process.

We identify that there is a gap of the refinement from the strategic business planning to ubiquitous services<sup>18</sup> implementation especially in the API (Application Program Interface)<sup>31</sup> economy trend.

Recently Gio Wiederhold<sup>32</sup> has identified the information imbalance of this gap and attributed this gap to the knowledge difference between economists and the IT professionals. He proposed to motivate the IT professionals to actively participate in filling this gap from the IT side. Some existing work<sup>30,25</sup> have devoted to bridge the business society and the computer science society on the service development evaluation. We proposed that knowledge is rules based on information and information is evaluation on observed data ontologically. However there are conceptual confusion among the relationship and semantics on these fundamental concepts of information science<sup>1,17</sup>, it makes many attempts to resolve the gap vulnerable to deep inquiries. In our response to Wiederhold's proposal, we proposed a systemic conceptual formalization from the value calculation to the design quality measurement which binds the modification and change on the design implementation artefacts with the business value strategy through a framework of manageable quality properties in a service design process under an investment perspective. This work differs from most existing work which are human experience centric<sup>29</sup> and focus largely either on service pricing<sup>2,7,23</sup> from the IT side or on cost<sup>19</sup> and benefits from the economics side<sup>4,15</sup> but neither covers both sides simultaneously which seamlessly convey the created value added. We model the whole system design process from data, information and knowledge introduction perspectives as stages of data sharing, information transfer, and knowledge creation<sup>24</sup> and control in an data, information and knowledge lifecycle. Thereafter we refine these data, information and knowledge processing stages as atomic data, information and knowledge processing activities and states, and relate these activities/states with their effects on system quality

properties which eventually are related to business values in computational manner. The evaluation on the adaptability and controllability of the related intermediate models and the modeling activities in the development process is enabled with a deviation measure centering core concepts of Under Design (UD) and Over Design (OD)<sup>11</sup> which represent the direction and distance of observable design degradation referring to the ideal model or best experience in an information/knowledge lifecycle. Based on this bundling mechanism between the design activities/states and their evaluations, we proposed to a top down decomposition of the economical goal to bridge the business planning and the technical sides in an unified process.

Finally, we conclude the paper with future work in Section 4.

## 2. Modeling of the expectation and its deviation

### 2.1. General idea

From an ontological development perspective, the coherent process from business planning to IT implementation can be viewed as the composition of basic conceptual activities of data observation/collection, information transferring from sender to receiver (could be human or machine) and knowledge creation. These basic activities comprise the superficial semantics of data, information and knowledge contained in requirement specification transferring to design artifacts accumulatively. On top of this accumulation, there are value added generation and business value balancing among stakeholders. Although there seems to be a gap between implementing the business strategies with IT techniques following the Value Driven ideology, from the ontological perspective the implementation are coherent in terms of the interplaying among the data, information and knowledge. What is necessary on this meta-level coherence to reveal the potential guidance for modeling and implementation activities in a model driven process is building a semantically pragmatic<sup>5</sup> measurement of the artifacts and activities on data sharing, information transferring and knowledge creation. To ease the distinguishing among the basic concerns such as the expectations of stakeholders,

we create a meta process in which the original data, information and knowledge space is full in the requirement side and empty in the artifact space. And during an implementation process top down from business planning to service composition, model artifacts grow from empty of data, information and knowledge to an expected state composing design artifacts properly reflecting the expectation of the stakeholders. For every design activity it either confirms to the expected effect or deviate from the ideal effect. We build the deviation measure mechanism to monitoring and measure the deviation for respective control purposes towards coherent implementation of business planning and IT implementation.

### 2.2. Core concepts preparation

We employed two core concepts of Under Design (UD) and Over Design (OD)<sup>11</sup> to identify the direction of the deviation of manipulation on data, information and knowledge and represent the measure of the extent of the deviation. UD and OD can provide shared semantics between business planning and IT implementation among stakeholders and are defined from the knowledge introduction perspective as follows:

**Definition: Over Design (OD)** refers to the state concerning design activities or design artifacts that the evaluated comprehensive value of the design product or product family or intermediate design model exceeds what is expected for its whole investment by stakeholders either from short run measurement or from a long run one. The produced extra quality or functionality such as more robust or complicated will cost extra resources including: project time, human effort, efficiency, effectiveness, consistency, clearness and deviation from optimized goals, with less comprehensive value gaining efficiency for the whole project usually.

From a variability management perspective, OD will leave less variability/constraint space<sup>8</sup> for further change or more content than expected from a cost efficient and effective criteria. Cognitively OD can be attributed to multiple scenarios including that:

(a) From the effect of the data, information and knowledge introduction behaviours' evaluation per-

spective, subjective knowledge created by designers instead of provided by stakeholders is introduced and transformed into data, information in design artefacts at design time which does not guarantee the objectiveness of introduced data and information and correctness of introduced knowledge, efficiency of the data/information quality, and exclude the useless/redundant/imprecise data/information.

(b) From the temporal perspective of data, information and knowledge introduction behaviours, objective data, information and knowledge from stakeholders is introduced earlier than its proper time at which the comprehensive value of the final system can be maximized as mapping to a series of behaviours following the best behaviour sequence and resulting in the optimal information structure. The implementation period of projects with OD might be either shortened because of the earlier data reuse, information communication and knowledge creation and exclusion of the optimized update, or extended because of the added effort.

**Definition: Under Design (UD)** is the design activity and planning side consequences that the evaluated value of the design product or product family or intermediate design model or protocol or mechanism in a certain design stage/state is less than what is expected for its whole project investment by some or all stakeholders either from short run measurement or from a long run one.

Cognitively UD can be attributed to that the data, information and knowledge provided by stakeholders is lost during the data sharing, information transferring, knowledge creation and their composition process unconsciously. The implementation period of projects with UD might be shortened because of the saved effort with reduced profit and satisfaction on stakeholders' sides. For example for Class model construction, if Class A shares every attribute and operation with Class B in an existing model, the to be added information is that Class C shares some attributes and operations with both Class A and Class B. If there is not enough introduced data, information and knowledge to properly decide whether Class C should inherit Class A or B, the decision to let Class C inherit both Class A and Class B will be OD if further information like "the attributes shared

by Class C with Class B all belong to Class A" is communicated. The decision to let Class C inherit Class A will be UD if further information like "the attributes shared by Class C with Class B do not fully belong to Class A" is communicated.

These two comparative concepts are labelled as the negation of each other in the labelling sense and expand on the contrary directions of each other in behavioural evaluation. We denote the actual design of a project as (AD) which represents the specific implementation of a project. We can understand reasons of the occurrence of UD and OD in the data sharing, information communicate and knowledge creation process through the concretion of the modeling of the expectation.

Evaluation of the quality of the design process: if the expectation of a system development process is composed of solely implementation of functionalities prescribed by the customers, the proper accomplishment is when all functionalities are implemented efficiently and the performance of the product is effective.

### 2.3. Hypothesis on the existence of the expected time and cost (HCos)

We make the following hypothesis on the existence of the expected time and cost. For a given fixed project goal ( $G$ ) which can be materialized as the efficient composition and effective performance of the interplaying of data, information and knowledge, there exists an objective period of time ( $T$ ), a fixed amount of human effort ( $H$ ) and a fixed amount of resources ( $R$ ) (storage, network, etc.) to accomplish the data sharing, information communicating and knowledge creation. These instances of  $T$  and  $H$  match to a proper design process plan ( $P$ ). By proper here it means that the the cost and investment ( $CosInv$ ) is minimized while reaching the optimized value maximization on stakeholders' sides.

$$\begin{aligned}
 HCos : \forall g \in G \\
 & ((\exists t \in T \wedge \exists r \in R \wedge \exists h \in H) \\
 & (t > 0) \wedge (r > 0) \wedge (h > 0) = true) \\
 & \wedge (\min(CosInv(P(t, r, h))) = true)
 \end{aligned}$$

If we take the context information of how long and how much human resources are used to accomplish a fixed set of functionalities and qualities into consideration, through the comparison between the actually used time and effort with the expected time and effort, we can label the excess of the expected period and effort in *HCos* as either OD or UD.

#### 2.4. Data, information and knowledge reuse/redundancy

The UD can come from the situation that some development activities are not planned efficiently and the resulting product is not effective as expected within the effort/resource limits. From the data sharing, information communicating and knowledge creation perspective, there are some data, information and knowledge utilization redundancy cases which produced the same data, information and knowledge repeatedly and unnecessarily. Repeated collection of the same data set instead of efficient data sharing, repeated production of the same information instead of efficient information communicating and repeated creation of the same knowledge will cost more time and waste more resource for originally avoidable processing. This situation of UD can be remedied with optimization through enhanced integration of the static data, information and knowledge reuse strategy and the dynamic data, information and knowledge transferring strategy which will coordinate the process of reproduction, transfer and reuse based on a trade-off of the gains vs. loss of respective processes. Taking information as an example of data sharing, information communicating and knowledge creation, we formalized the scenario of UD in this situation covering the ideal criterion, UD identification, measure and remedy strategy as follows:

$$\begin{aligned}
 &Data = ObservationBy(Human | Machine); \\
 &State = CompositionOf(Data); \\
 &Event = ObservableChange(State); \\
 &Change = IsA(Relationship); \\
 &Activity(x) = LabelOfEvent | ([a - zA - Z0 - 9 - ])^+;
 \end{aligned}$$

$$\begin{aligned}
 &Information(x) = \\
 &\quad (LabelOfInfo)^+ | EvaluationOn(Data);
 \end{aligned}$$

$$\begin{aligned}
 &\forall x \in Activity \\
 &\quad occurrence(x) \Rightarrow amount(x) = amount(x) + 1;
 \end{aligned}$$

$$\begin{aligned}
 &\forall x \in Activity, \forall y \in Information | Data, \\
 &\quad y = InfoCommunicating(x) | DataSharing(x), \\
 &\quad InfoCommunicating(x) \Rightarrow occurrence(x) \\
 &\quad \wedge (y \notin \phi) \wedge (H(x) \notin \phi) \wedge (R(x) \notin \phi) \wedge (T(x) \notin \phi); \\
 &\quad DataSharing(x) \Rightarrow occurrence(x) \\
 &\quad \wedge (y \notin \phi) \wedge (R(x) \notin \phi) \wedge (T(x) \notin \phi);
 \end{aligned}$$

$$\begin{aligned}
 &\forall x = InfoCommunicating(i) | DataSharing(x), \\
 &\forall y = InfoCommunicating(j) | DataSharing(x), \\
 &\quad (x = y) \Leftrightarrow (i = j);
 \end{aligned}$$

$$\begin{aligned}
 &IdealModel(InformationReuse) : \\
 &\quad \forall p \in P, \forall x \in Activity_p, amount(x) = 1;
 \end{aligned}$$

$$\begin{aligned}
 &UDofInformationReuse(Identification) : \\
 &\quad \exists x \in Activity, amount(x) \leq 2; \\
 &\quad \forall h \in H_{actual}, \forall t \in T_{actual}, \\
 &\quad \quad (\sum t > T_p) \wedge (\sum h > H_p)
 \end{aligned}$$

$$\begin{aligned}
 &UDofInformationReuse(Measure) : \\
 &\quad \exists x \in Activity, amount(x) \geq 2; \\
 &\quad h \in H_{actual}, t \in T_{actual}, \\
 &\quad DegreeofUD : (\sum t) / T_p * (\sum h) / H_p
 \end{aligned}$$

$$\begin{aligned}
 &RemPolicy : \\
 &\quad \forall x \in Activity, minimize(amount(x)_{actual}) \rightarrow 1
 \end{aligned}$$

#### 2.5. Data, information and knowledge flow control

The UD can come from the situation that some design and development activities are not arranged efficiently or effectively. From the data, information, knowledge and value transferring perspective, data, information and knowledge flow need to be well

planned to avoid the inefficient situations such as the sequential information transferring which otherwise can be transferred in a parallel mode or vice versa. Although improper information control might not influence the overall invested human effort, it will cause information delay for modeling activities. Then the time span of the development process will exceed the expected time in the ideal process of P and results in UD. In efficient information flow will increase the pressure on storage and bandwidth.

$$\begin{aligned}
 &UDofInformationFlowControl : \\
 &\forall x \in Activity, \\
 &\exists input(x), output(x) \in Information, \\
 &Activity(x) = (input(x), output(x));
 \end{aligned}$$

$$\begin{aligned}
 &IdealModel(InformationFlowControl) : \\
 &\forall x, y \in Activity, \\
 &(input(x) == input(y)) \Rightarrow parallel(x, y);
 \end{aligned}$$

$$\begin{aligned}
 &UDofInformationFlowControl(Identification) : \\
 &\exists x, y, z, \dots \in Activity, \forall h \in H_{actual}, \forall t \in T_{actual}, \\
 &\forall h \in H_{actual}, \forall t \in T_{actual}, \\
 &UD(x, y, z, \dots) \Rightarrow (\sum_{actual} t) > T_P;
 \end{aligned}$$

$$\begin{aligned}
 &UDofInformationFlowControl(Measure) : \\
 &\forall h \in H_{actual}, \forall t \in T_{actual}, \\
 &DegreeofUD : (\sum t) / T_P
 \end{aligned}$$

## 2.6. Data, information and knowledge loss

The improper design can come from the situation that some data, information and knowledge are lost during the information transferring process from stakeholders' side. Although data can be the target of transferring, data become information conceptually when it is transferred. Knowledge transfers data and information in incomplete sense to predictions of unknown background. Taking information as the example, the lost information can be classified as follows:

- Content information (*CTInfo*): direct content data/information will help to reduce the information gaps among existing model constructs. The accumulation of the content information will lead to the increase of the volume of the data/information of the target models in the form of either the amount of the entities and relationships directly connecting the entities or the effectiveness of the product mechanism.
- Constraint information (*CSInfo*): to distinguish from the content information, we restrain the constraint information as information which does not directly comprise the target model but logically categorizing on the existing information in terms of negative logical connectives of exclusion functionality. The accumulation of the constraint information on the target models will decrease the amount of entities and relationships directly connecting the entities.

Losing these two types of information will lead to different design deficiencies.

- Loss of *CTInfo*: losing *CTInfo* cannot be evaluated as a success in terms of finishing the project earlier cause it is incomplete compared to P. Losing certain information can also lead to the impossibility to accomplish the project. In absence of the validation of the completeness of the system, it is impossible to identify the deficiency as either UD or OD. If we assume the complete model as a state composing all expected information, which we denote as *CTInfo*, the loss of *CTInfo* will result in an information set with less information. We identify this lack of modeling as UD. The remedy might not bring full recovery of the efficiency and effectiveness as that of the ideal P since the supplying activity of the missing information might not be as efficient as that of the ideal P since its occurrence differs from the corresponding position in the ideal execution sequence.

$$\begin{aligned}
 &UDofLossCTInfo(Identification) : \\
 &\forall info \in Information \\
 &UD(actual) \Rightarrow (\sum_{actual} info) < (\sum_P info);
 \end{aligned}$$

$$UDofLossCTInfo(Measure) : \\ DegreeofUD : (\sum_{actual} info) / (\sum_P info)$$

$$remedy(LossCTInfo) : \\ \forall x \in \{(\sum_P info) - (\sum_{actual} info)\}, \\ supply(x, \sum_{actual} info);$$

- Loss of *CSInfo*: to distinguish from the content information, we restrain the constraint information as information which does not directly compose the target model but logically categorizing on the existing information in terms of negative expressions on the target model. The accumulation of the constraint information on the target models will decrease the amount of entities and relationships directly connecting the entities. The loss of *CSInfo* will result in OD of the final system and usually with shorter development time and less effort.

$$ODofLossCSInfo(Identification) : \\ \forall info \in Information \\ OD(actual) \Rightarrow (\sum_{actual} info) > (\sum_P info);$$

$$UDofLossCSInfo(Measure) : \\ DegreeofOD : (\sum_{actual} info) / (\sum_P info);$$

$$remedy(LossCSInfo) : \\ \forall x \in \{(\sum_{actual} info) - (\sum_P info)\}, \\ reduce(x, \sum_{actual} info);$$

### 2.7. Information error (referring to True/False vs. Yes/No )

We propose that information and data if not confirm to the objective justification which usually refers to the real world or logic directly or indirectly is false in comparison to true. We also propose that if the knowledge content confirms to the knowledge owner's subjective expectation, the knowledge is justified as correct in contrast to wrong. Sometimes false information or information generated from the

wrong knowledge might be introduced without notice. This kind of situation can result in any situation in the target model. Theoretically error information might be any improper information which does not match the expected information. Here we distinguish the situation of error information from the situation that after information loss subsequent information is introduced but not matching the expected information. We also doesn't extend the error information to cover the situation of imprecise information. Therefore we limit the error information as information which is negative or contrary to the expected information. The information error doesn't provide easy semantics for directly identification as UD or OD without context information. However sometimes, error information will change the direction of the development of the situations of UD and OD. In this paper, we don't focus on error information processing besides excluding it from the targeted information categories.

$$WrongInformation(DefinitionHere) \\ \forall infoWrong \in Information, \exists info \in Information, \\ infoWrong = \neg info;$$

### 2.8. Information useless

We restrict some information/data/knowledge as useless since it will not contribute to neither the positive construction nor the negative construction of the target model in the end. This kind of information introduction will incur extra effort, resource and time cost. A special case of useless information is that useful information is unnecessarily but repeatedly introduced and stored. The useless information introduction is considered as UD.

$$UDofInformationUseless(Identification) : \\ \exists x \in Activity, h \in H_{actual}, t \in T_{actual}, \\ (\sum t > T_P) \wedge (\sum h > H_P)$$

$$RemedyPolicy : \\ \forall x \in Activity, \min(amount(x)_{actual}) \rightarrow 0;$$

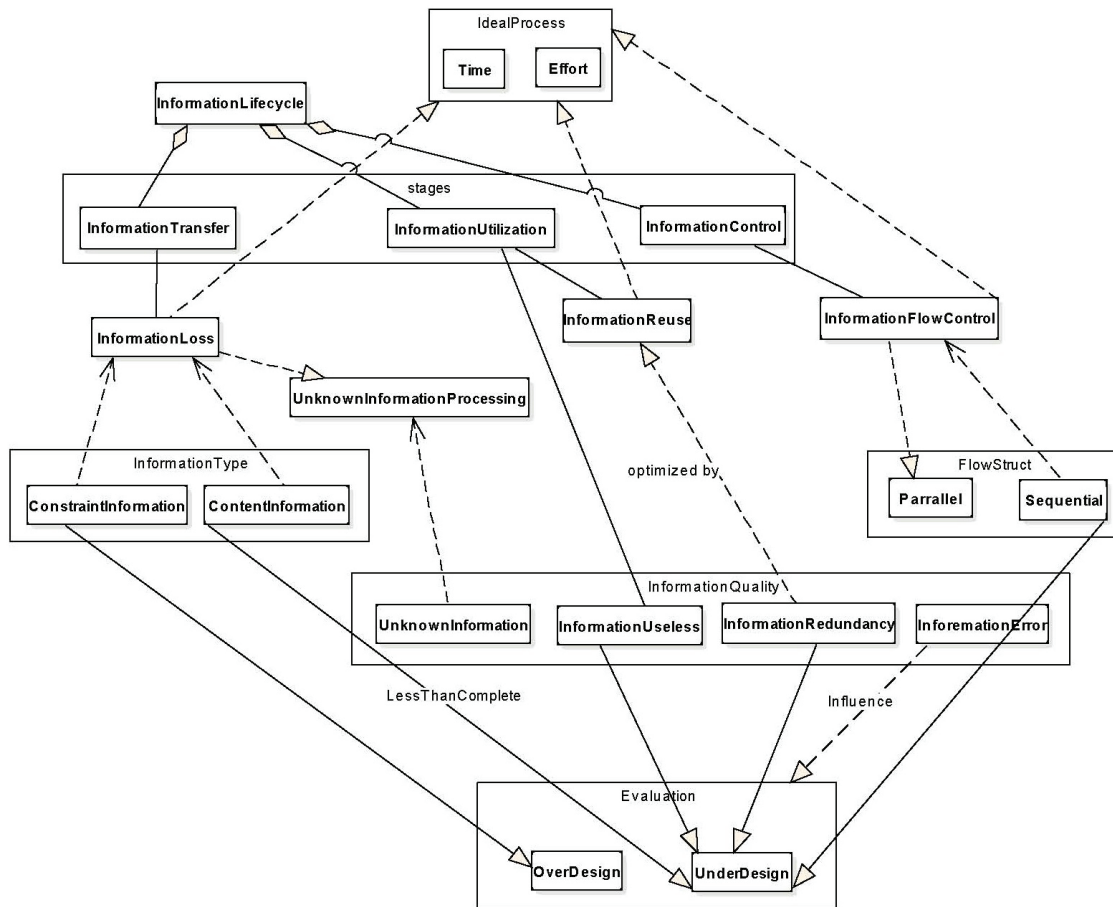


Figure 1: Revelation of the sources of UD and OD from an information life cycle perspective



### 2.9. Improper unknown information processing

Unknown information processing which deals with the information which is considered as unknown need to be considered during both the information introduction and model evolution stages. Improper unknown information processing will drive the design process away from the ideal process and bring extra cost to fix it if possible. There are several cases of unknown information processing. Among them, we limit the investigation here from the information lifecycle perspective to the information transfer modes of bidirectional transformation between known data/information and unknown data/information. Therefore this division avoid the overlapping on the other phases of the data/information lifecycle such as the unknown information transfer which is intertwined with the constraint information processing. If we restrict the unknown information expression with the Close World Assumption(CWA), the unknown information will be justified as a scope of undecided possibilities/choices which need to be determined with further information introduction. This form of unknown information can be matched to a limited amount of constraints. The expected information processing are expected to keep the information before processing and after processing consistent.

*ExpectedInformationProcessg(P)*

$$\begin{aligned} &\forall \text{known}_{\text{before}} \in \text{Information}, \\ &\exists \text{known}_{\text{after}} \in \text{Information}, \\ &\text{known}_{\text{before}} \rightarrow \text{known}_{\text{after}}; \end{aligned}$$

$$\begin{aligned} &\forall \text{unknown}_{\text{before}} \in \text{Information}, \\ &\exists \text{unknown}_{\text{after}} \in \text{Information}, \\ &\text{unknown}_{\text{before}} \rightarrow \text{unknown}_{\text{after}}; \end{aligned}$$

The unexpected situations can be classified as the following two modes. The first mode denotes the information transferring mistakes of losing the decision of known information which fails into the scope of content information loss. The result of content loss will result in that there is no determinant information to fill the gap left by the lost. Thereafter there are usually several undecided possibilities

which still indicate unknown. The second mode denotes the situation that unknown information which contains several choices is taken as solely choice. This process can be a result of content information of other choices are lost and only one choice is left in the end. Then it fails into the category of content information loss. Fig. 1 shows the whole picture of the revelation of the UD and OD from the perspective of the information lifecycle.

*UnknownInformationProcess(CloseWorldModes) :*

$$\forall \text{known} \in \text{Information}, \exists \text{unknown} \in \text{Information}, \text{known} \rightarrow \text{unknown};$$

$$\forall \text{unknown} \in \text{Information}, \exists \text{known} \in \text{Information}, \text{unknown} \rightarrow \text{known};$$

For the unknown to known information transformation, knowledge plays a proper role of predicting unknown information from incomplete data/information which is different from the erroneous information transferring.

### 3. Decompose system development economics

To measure and control the development and design processes of a service system, we propose to model the ultimate economics goal as for maximizing the general value added of the final system in the whole business process. We model this goal implemented through an actual design (AD) in compare to an ideal design (ID) in a business scenario as follows:

*IdealDesign(ID) :*

$$\begin{aligned} &\forall x \in \text{Information}_{\text{requirement}}, \exists y \in \text{Activity}_{\text{transfer}}, \\ &\forall z \in \text{Information}_{\text{requirement}_{\text{IdealSystem}}}, \\ &z = y(x) \wedge \text{optimized}(\text{sequence}(y)); \end{aligned}$$

$$\begin{aligned} &\text{optimized}(\text{sequence}(y)) \Rightarrow \\ &\text{minimize}(\text{error}) \wedge \text{maximize}(\text{reuse}) \\ &\wedge \text{maximize}(\text{parrallel}); \end{aligned}$$

UltimateEconomicGoal :

$$\begin{aligned} & ValueAdded(FinalSystem_{ID}) \\ & \geq Maximize(ValueAdded(FinalSystem_{AD})) \geq 0; \end{aligned}$$

$$\begin{aligned} & ValueAdded(FinalSystem_{AD}) \\ & = scale(user|market) \times efficiency(AD) \times \\ & \quad correctness(AD); \end{aligned}$$

This scenario takes into consideration of factors including the client/market scale of the final system, the efficiency of the design process and the correctness of design result. In the subsequent paragraphs, we propose to model the computation mechanism for the efficiency and correctness of the design from the data sharing, information communicating and knowledge creation perspective to compliment the value calculation here. We propose to model the correctness of the actual design as follows. The rate of completeness indicates the percentage of expected data/information/knowledge which has been transferred. The rate of wrong information indicates the reliability of the accomplished data/information/knowledge transfer.

$$\begin{aligned} & \forall WrongInfo, UsefulInfo \in Information, \\ & UsefulInfo \in Information_{ID}, \\ & Correctness(AD) \\ & = (1 - rate(WrongInfo)) \times rate(Completeness); \end{aligned}$$

$$rate(WrongInfo) = \sum_{AD} WrongInfo / \sum_{ID} Info;$$

$$rate(Completeness) = \sum_{AD} UsefulInfo / \sum_{ID} Info;$$

We propose to model the efficiency of the actual design as composing both structural efficiency and dynamic efficiency. Structural efficiency depicts the difference between the most efficient structure experience/knowledge and the actual structure in terms of reuse accomplishment, redundant information management and information organization. The quality of information organization

represents the conformance of the topological deployment of the information among classes, attributes and operations, with the optimized experience/knowledge/patterns. The quality of organization will not only influence the possibility of reuse but also the communication path length, maintenance and security, etc.

$$\begin{aligned} & efficiency(AD) = \\ & \quad efficiency(AD)_{structure} \times efficiency(AD)_{dynamic}; \end{aligned}$$

$$\begin{aligned} & efficiency(AD)_{structure} = \\ & \quad rate(reuse) \times rate(redundency) \times rate(organization); \end{aligned}$$

$$\begin{aligned} & rate(organization) = \\ & \quad rate(ClassDistribution) \times rate(AttributeDistribution) \\ & \quad \times rate(OperationDistribution); \end{aligned}$$

The dynamic efficiency depicts the data/information/knowledge usage for a specific data/information/knowledge structure set will be influenced by the mechanism of component/class communication, data/information/knowledge flow and control flow. Although the policies and restriction in real situation will influence the enforcement of the system, we exclude them as outer factors which aren't usually determined by developers.

$$\begin{aligned} & efficiency(AD)_{dynamic} = rate(ControlFlow) \times \\ & \quad rate(DataFlow) \times rate(Communication); \end{aligned}$$

#### 4. Conclusion and future work

We aims at coping with the challenge that there is few value oriented or value driven whole data, information and knowledge lifecycle evaluation approaches for integrated bidirectional evaluation and control on the efficiency and effectiveness of the design activities between detailed design activities/content and business modeling. By referring to current good practices in business planning and software economics, we proposed a conceptual systemic value calculation modeling centering this gap in terms of the coherent basic data, information and knowledge interplaying underneath the gap from business planning to IT implementation. The work

is carried out from the ontological formalization on the basic constructs of system design and evaluation and is expected to lay a foundation for further computational constructs which cross good economical knowledge and design experiences towards a Win-Win service mashup environment with increased Value Added for all stakeholders. Currently in this work some qualities such as reliability and precision are not explicitly proposed in the calculation. This is partially because of the difficulty from the reality that the meaning of these qualities lacks uniform definitions. We managed to consider these qualities in the calculation implicitly. For example, the precision of the transfer rate is represented in the correctness of the content transfer and the efficiency of the transfer process, and the reliability of the system is partially reflected in the efficiency of the information structure and the management of the information reuse and information redundancy. Some important factors are too specific and are not considered at this stage include the location, data format, transaction law and policies which depend on the context of the specific service transaction scenarios.

In the future we will also endeavour to implement the proposed approach for prototype system development to get feedback for further improvement.

### Acknowledgement

The authors acknowledge the support of the NSFC of China (No. 61363007 and No. 61662021 and No. 61662019), Hainan NSF (No. 20156234).

### References

1. R. L. Ackoff. From data to wisdom. *Journal of applied systems analysis*, 16(1):3–9, 1989.
2. M. Al-Roomi, S. Al-Ebrahim, S. Buqrais, and I. Ahmad. Cloud computing pricing models: A survey. *International Journal of Grid & Distributed Computing*, 6(5):93–106, 2013.
3. B. Boehm and K. Sullivan. Software economics: status and prospects. *Information & Software Technology*, 41(14):937–946, 1999.
4. G. P. Cachon and M. A. Lariviere. Supply chain coordination with revenue-sharing contracts: Strengths and limitations. *Management Science*, 51(1):30–44, 2005.
5. P. R. Carlile. A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization science*, 13(4):442–455, 2002.
6. P. D. Collopy and P. M. Hollingsworth. Value-driven design. *Journal of Aircraft*, 48(48):749–759, 2012.
7. M. A. Cusumano. The changing labyrinth of software pricing. *Communications of the Acm*, 50(7):19–22, 2007.
8. A. Demuth, R. E. Lopez-Herrejon, and A. Egyed. Constraint-driven modeling through transformation. *Software & Systems Modeling*, 14(2):573–596, 2015.
9. Y. Duan. A survey on service contract. In *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 805–810, 2012.
10. Y. Duan. Value modeling and calculation for everything as a service (xaas) based on reuse. In *Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), 2012 13th ACIS International Conference on*, pages 162–167, 2012.
11. Y. Duan, Q. Duan, A. O. Elfaki, and C. Ren. Formalizing over design and under design of value engineering for context-aware cloud service development. In *2015 3rd International Conference on Future Internet of Things and Cloud*, pages 72–79, 2015.
12. Y. Duan, K. Huang, D. Chen, Y. Wang, A. Kattapur, and W. Du. Service value broker patterns: An empirical collection and analysis. *International Journal of Networked and Distributed Computing*, 2(1):54–69, 2014.
13. Y. Duan, A. Kattapur, Q. Zagarese, and W. Du. Service value broker patterns: Integrating business modeling and economic analysis with knowledge management (short paper). In *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications, Koloa, HI, USA, December 16-18, 2013*, pages 140–145, 2013.
14. Y. Duan, X. Sun, A. Longo, Z. Lin, and S. Wan. Sorting terms of aas of everything as a service.
15. G. Gereffi and J. Lee. Economic and social upgrading in global value chains and industrial clusters: Why governance matters. *Journal of Business Ethics*, 133(1):1–14, 2014.
16. P. Gray. Software economics, history, theology. *Information Systems Management*, 29(2):161–164, 2012.
17. J. Hey. The data, information, knowledge, wisdom chain: the metaphorical link. *Intergovernmental Oceanographic Commission*, 26, 2004.
18. C. S. Hussain, C. S. Ahmed, A. H. Akbar, and A. K. Bashir. Ubiquitous service discovery in pervasive computing environment. *Information Technology Journal*, 7(3):533–536, 2008.

19. M. Jorgensen and M. Shepperd. A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering*, 33(1):33–53, 2007.
20. H. Katzan Jr and W. A. Dowling. Software-as-a-service economics. *The Review of Business Information Systems*, 14(1):27, 2010.
21. B. Kitchenham and E. Mendes. Software productivity measurement using multiple size measures. *Software Engineering IEEE Transactions on*, 30(12):1023–1035, 2004.
22. P. P. Maglio, S. Srinivasan, J. T. Kreulen, and J. Spohrer. Service systems, service scientists, ssme, and innovation. *Communications of the ACM*, 49(7):81–85, 2006.
23. M. Mathew and S. Nair. Pricing saas models: Perceptions of business service providers and clients. *Journal of Services Research*, 2010.
24. I. Nonaka. *The knowledge-creating company*. Harvard Business Review Press, 2008.
25. L. O'Brien. A framework for scope, cost and effort estimation for service oriented architecture (soa) projects. In *2014 23rd Australian Software Engineering Conference*, pages 101–110, 2009.
26. A. Parasuraman, V. A. Zeithaml, and L. L. Berry. A conceptual model of service quality and its implications for future research. *the Journal of Marketing*, pages 41–50, 1985.
27. P. C. Pendharkar and J. A. Rodger. The relationship between software development team size and software development cost. *Communications of the Acm*, 52(1):141–144, 2009.
28. D. Rodriguez, M. A. Sicilia, E. Garca, and R. Harrison. Empirical findings on team size and productivity in software development . *Journal of Systems & Software*, 85(3):562–570, 2012.
29. M. Shepperd. Software project economics: a roadmap. In *Future of Software Engineering*, pages 304–315, 2007.
30. B. Tansey and E. Stroulia. Valuating software service development: Integrating cocomo ii and real options theory. In *International Conference on Software Engineering Workshops*, pages 8–8, 2007.
31. M. Vukovic, J. Laredo, and S. Rajagopal. Api terms and conditions as a service. In *2014 IEEE International Conference on Services Computing (SCC)*, pages 386–393, 2014.
32. G. Wiederhold. Unbalanced data leads to obsolete economic advice. *Communications of the Acm*, 59(1):45–46, 2015.
33. C. Zins. Conceptual approaches for defining data, information, and knowledge. *Journal of the American society for information science and technology*, 58(4):479–493, 2007.