ON THE USE OF CHURCHMAN’S INQUIRING SYSTEMS IN INFORMATION SYSTEMS DEVELOPMENT
Andrea Carugati, IESEG School of Management, a.carugati@ieseg.fr
Nathalie Demoulin, IESEG School of Management, n.demoulin@ieseg.fr
Mailing contact (both authors): 3 Rue de la Digue, 59800 Lille, France
Telephone contact (both authors): +33 (0)3 20545892

ABSTRACT
The paper presents a framework to interpret information systems development (ISD) as composed of activities for the creation and exchange of knowledge. The framework is based on the five inquiring systems presented by Churchman [13]. The main contribution of the framework is in showing that the inquiring systems are complementary rather than competing: rationalism, empiricism, idealism, dialectic, and pragmatism are, contrarily to Churchman’s perspective, used by participants in ISD projects in accordance to the scenarios they face. Consequently, for each ISD activity there is a predominant inquiring system. Since the framework focuses on the knowledge created during the interaction of individuals and groups (or the lack there of) rather than on the specific development process the framework is robust enough to be adaptable to different ISD methodologies. While methodologies are largely centered on specific paradigms for knowledge creation, the framework focuses on micro-level activities and suggests specific management practices to support the different knowledge creation situations.

Keywords: Information systems development, inquiring systems, knowledge creation.

INTRODUCTION
Information systems development (ISD) is usually a process that emerges from two interconnected activities: software producing activities and managerial functions [1]. Novel research tends to pair these two dimensions with the dimension of knowledge creation activities [24, 6, 4, 14, 12]. The general agreement is that a social constructivist approach to ISD where the systems and their requirements emerge from the interaction of multiple stakeholders is preferred to an objectivist approach that considers requirements as existing outside the individual [12].

This paper continues this research tradition presenting a framework that sets the spotlight on the knowledge creation and exchange facets of ISD projects. The goal of the paper is to provide a framework to interpret information systems development in terms of the inquiring systems proposed by Churchman [13]. The purpose is twofold. First, we show that in ISD multiple inquiring systems are used opportunistically and in dependence on the context in which ISD stakeholders move. Second, we also intend to help managers to see ISD activities as knowledge creation processes and to provide them with few principles for their management to improve the likelihood of ISD projects success.

The paper provides an introduction to Churchman’s inquiring systems and a robust framework that can be used in conjunction with multiple methodologies. The framework maps Churchman’s inquiring systems on the generic ISD activities proposed in the ISO 15704 standard [15]. The activities are intended as modules whose specific sequence and content varies according to the ISD methodology chosen. The paper concludes by presenting a primer on the managerial implication of using inquiring systems and knowledge creation activities as guiding light in ISD.
INQUIRING SYSTEMS

According to Churchman [13] there are five basic modes of inquiry to create knowledge. Described very shortly, these five modes are:

Rationalism or the Leibnizian Inquiring System: In this mode, the inquirer finds within himself, through formal logic, the truth governing the world. The system under study is considered as closed and governed by causal relationships. The inquirer creates networks of hypotheses and theses and proceeds to build on these hypotheses until logic reveals a counterhypothesis that invalidates a part of the network. These networks are called fact nets. A complete model of the world is the objective of this inquiring system and it is needed before empirical observations can be recorded. Brought into the world of ISD, the Leibnizian inquirer is the isolated individual reasoning on the causal relationships governing the system that he/she wants to improve. The individual’s logic becomes the guarantor of the truthfulness of the fact net. Pure structured methodologies are built on the ideas of the Leibnizian enquiring system: once requirements are specified the programmer proceeds on his/her own to develop the information system based on his/her internal logical reasoning. The information system will therefore reflect the fact nets developed by the inquirer (the programmer). Without external interference the information system will be delivered to the customer years after the requirements have been agreed upon.

Empiricism or Lockean Inquiring System: This mode of inquiry differs from the Lockean in that the inquirer gives more weight to data than theory. The inquirer is open to the environment and finds the truth in the surrounding world. The inquirer is not isolated but participates in creating knowledge with a community of inquirers. The inquirer uses data and expert opinion, especially the opinion of his community, to explain the world. The community acts as a source of information but also as guarantor of the truth through agreement and consensus. Lockean communities tend also to become defensive of the knowledge that they have generated and of the methods used to generate it. In ISD, the groups of users and developers can be considered as two Lockean communities. Agreement on the nature of the problem can become the truth for the users’ community and agreement on the nature of the solution can become the truth for the developers’ community. In Lockean inquiring systems there is only space for one truth but there is no guarantee that two communities will arrive to an agreement on a common truth.

Idealism or Kantian Inquiring System: This mode of inquiry gives equal weight to data and theory. It is recognized that there are multiple ways in which a problem may be analyzed using multiple models that can be applied to the observations. However the inquirer does not accept multiple truths; the objective is to find the ideal, and only, truth. The Kantian inquirer proceeds testing the fitness of a model to the data and creates knowledge by finding the model that best fit the data. In ISD it can be imagined that the Kantian inquiring system is used when developers have to find a model to represent something from the client system into the information system. Many possibilities are tried and recursively eliminated until only one, the one with the best fit, the truth, remains.

Dialectic or Hegelian Inquiring System: sees the truth emerging from opposing views. Debate between different worldviews is seen as the only way to develop theses and antitheses to arrive at a synthesis that accommodates both worldviews. Hegel introduces the idea of multiple interpretations of reality even though he believed in a final “grand synthesis”. He envisioned two groups of people engaged in an ardent debate respectively defending their thesis and trying to destroy the other group’s antithesis. The issue is than resolved by an impartial observer that
synthesizes the arguments to create a synthesis. The guarantor of the truth generated is the belief in the efficacy of debate between extreme points of view.

**Pragmatism or the Singerian Inquiring System:** this inquirer is based on the ardent debate of the Hegelian enquirer to create progress but accepts multiple sources of data and multiple interpretations of reality and consequently multiple truths. The multiple truths are found via an approach that continuously attacks currently held beliefs from multiple points of view. The world inquired is interpreted as an open system [2, 19] where all components interact with each other. The truth is temporary and context-dependent because the more elements are “swept in” the system the more the truth is likely to change. Progress is measured quantitatively whenever possible otherwise the groups’ intuition that progress is made becomes the guarantor for qualitative measures. In ISD, the use of the Singerian inquirer can be found in the more recent methodologies, for example in the methodologies emerged from Soft Systems Methodology [3, 24], Multiview2 [4] and in agile methodologies [6, 14]. In these methodologies, the purpose is to let the information system emerge from the debate between developers and users letting the truth be accepted pragmatically by the participants and not imposed top down.

From the individual perspective, it is acceptable to use different inquiring systems according to the context rather than according to an externally defined characteristic of the problem. Harrison and Bramson [cited in 17] suggest that individuals display one or two favorite modes of inquiry but that people can use different modes opportunistically and according to the situation. The individual working in isolation might tend to relay mostly on rationalism and idealism to move forward. Members of the same group will rely mostly on rationalism, idealism, and empiricism since its members possess a shared worldview [7]. Dialectic and pragmatism are instead inquiring modes used when two groups with different worldviews meet with the intent of learning from each other.

**FRAMEWORK DEVELOPMENT**

The ISD process can vary radically depending on which development methodology has been chosen. However, there are five basic activities that are shared by all methodologies. First, some individuals in an organization voice a desire for a new system; this desire is ultimately transformed into a concept for a new information system. Second, the groups of developers and users meet and discuss the requirement for the information system. Third, the developers discuss among themselves how to address the requirements. Fourth, each developer develops the part of the information system of his/her competence. Fifth, the developers and the users meet to test the system and eventually implement it. In the ISO 15704 standard [15] these activities are respectively called 1) Identification and Concept, 2) Requirement Definition, 3) System Design, 4) Implementation and 5) Operation. By interpreting the activities in terms of systems for creating knowledge, it is possible to allege a connection between them and the use of one or multiple inquiring systems. The following part maps the inquiring systems on the ISD activities. The **Identification and Concept phase** is normally carried out within an organization that experiences problems or has identified some opportunities that it wants to seize or solve with the use of information systems. The users will discuss about their situation and how they could expect an IS to help them to improve their situation. In this phase, the group is likely to consider their world as an open system and the most likely inquiring mode adopted is empiricism. The group members will seek consensus that acts as guarantor for the created knowledge.

The **Requirements Definition phase** is carried out by users and developers who debate and finally reach an agreement on the functional role of the information system and on the project’s goals. In this phase, the groups’ best choice is to behave as Singerian inquirers. The two groups discuss
problems while including elements from all interconnected systems and they will come with an idea of what the information system has to do. This idea is not necessarily shared by everybody, common with this inquiring system, and therefore decisions about rework can be made without hesitation. The guarantor of newly created knowledge is the intuition of the participants that progress has been made. Quantitatively progress can be measured by the increase in “size” of the requirement document.

The Design is the phase during which the developers work together as a group to plan how to bring the requirements into life. A variety of technical decisions is made here. These are usually decisions that allow the developers to work individually during the implementation phase. During this phase, developers enhance the knowledge “obtained” from the users. They reconsider the dialogue with the users and eventually create their own hypotheses about the users’ system and the solution proposed. Knowledge is thus created through interpretation of the gathered information from the users. Developers behave as Lockean inquirers like the users group in the identification phase. They are open to the environment but create the knowledge of how the new system has to operate, internally in the group. Members seek consensus within the group that acts as a guarantor for the created knowledge.

The Implementation is the third step in the course of which the developer, now isolated, begins to elaborate on what he learned from the users and from his colleagues. The system developer acts as a monad in a Leibnizian enquiring system. When writing code, he interprets the information in his possession and creates a network of hypotheses that bring him from ideas to actual code. New knowledge is created through a process of deduction. Fact nets are created based on the hypothesis on the behavior of the users’ system and grow as more and more elements are added into the program. Occasionally the developer faces uncertainty regarding how well a particular algorithm replicates a behavior observed in the users’ system. If multiple algorithms have to be compared the developer enter a Kantian mode of inquiry to find which algorithm fits best the users’ needs and behavior. In this inquiry mode, the developer might follow his own judgment or search for answers outside. Even though the answers come from the outside, the goal of the system is only to justify the use of a certain algorithm or solution, not to put under discussion the work done up to the point. The inquiry is purely Kantian. The quest finishes with the knowledge that a particular algorithm fits a particular situation. Once the best fit is found, the developer considers that a newer element has been added to the fact net and proceeds with his work.

The work in phase three and four results in a continuous evolution in the developer’s knowledge. These phases are not simply an operationalization of the agreed requirements; they all contribute to the creation of software artifacts that can be different from what the users expect.

The Operation is the phase during which developers show the software to users. The users operate the software and compare the experience with their expectations, which are debated to achieve new agreements: the Singerian inquiring system is used again. If the software is implemented, the users have the opportunity to use it for regular work and perhaps request additional features or modifications at the following meetings. New knowledge, created by the users in an operational environment outside the modus operandi of the debate is created using the Lockean inquiring mode.

The resulting framework is presented in table 1. Each activity (vertical axes) is mapped on the inquiring systems (horizontal axes) that are most likely to be active during the phase.
### Life Cycle Phases ↓ Inquiring Systems active during ISD Life Cycle Phases

<table>
<thead>
<tr>
<th>Identification and Concept</th>
<th>[\star]</th>
<th>[\star]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Definition</td>
<td>[\star]</td>
<td>[\star]</td>
</tr>
<tr>
<td>Design</td>
<td>[\star]</td>
<td>[\star]</td>
</tr>
<tr>
<td>Implementation</td>
<td>[\star]</td>
<td>[\star]</td>
</tr>
<tr>
<td>Operation</td>
<td>[\star]</td>
<td>[\star]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inquiring Systems →</th>
<th>Empiricism</th>
<th>Rationalism</th>
<th>Idealism</th>
<th>Pragmatism (incl. Dialectic)</th>
</tr>
</thead>
</table>

### Table 1. Theoretical Framework

### A PRIMER ON MANAGERIAL IMPLICATIONS

Considering ISD as a knowledge process changes the idea of what information system success is. The framework enables to see the ISD process as creation of knowledge rather than as creation of information systems. Below some managerial principles are provided deriving from this way of seeing the ISD process.

1. **Disagreement is inevitable.** Even though people seem to agree, there is likely to be a difference between what people say and what people do. This is because in the identification phase and design phase consensus is reached in groups with different worldviews. Even when using the Singerian inquiring mode, consensus is, a priori, not certain. However, the more the Singerian inquiring system is used, the more likely it is that the resulting information system would serve all involved interests. In ISD, this means going through the operation phase many times putting developers and users in the condition of debating the work done.

2. **Requirements are not the information system.** Developers will tend to develop the software partially basing their work on the discussion with the users and partially basing it on their own reasoning. In ISD, there is always an element of creativity. If this knowledge results in small fact nets, they can be easily analyzed and invalidated during testing but if the nets are very large discovering all the causal relationships that have been embedded is almost impossible. Consequently, time intervals between testing phases should be kept short to limit the uncontrolled growth of fact nets and the size of the development task should be controlled so that small prototypes can be developed in the short time allocated.

3. **Debate does not come automatically.** Entering in the Singerian inquiring mode is not easy. The Singerian inquirer requires cooperation and diffused authority among participants [21]. Presentations are a communicative genre that implies a well specified locus of authority [22] and therefore it does not fit the requirement of diffused authority. Project managers on both sides should agree to mediate the discussion instead of leading it and structuring it around frequent interaction.

4. **Testing should be done by users.** In ISD projects, most of the work on the code is done by developers. They know the code inside out and assumptions become easily black boxed [18]. In other words, developers cannot attack with critical spirit their own fact nets because these have emerged from their own logic. Therefore, users should test the information system in order to use the knowledge gained during testing as base for debate [11]. The information system prototype should be built as a boundary object [10, 11] to maximize learning across
On the Use of Churchman’s Inquiring Systems in Information Systems Development

groups. Boundary objects should be visual [10, 8], functional [9], and up-to-date [10]. Only by building the software like this and by letting the users operate it we can expect “ardent” debate to emerge.

These guidelines partially confirm the recent interest in emergent methodologies like extreme programming [6] that in fact include many of the principles proposed here. However, these guidelines can be applied also to structured methodologies where the knowledge dynamics presented in the paper are also active.

CONCLUSION

This paper presents a novel view of ISD where the traditional software creating activities and managerial activities are joined by knowledge producing activities. In doing so, this article continues and extends the tradition of research in prototyping and participatory design further specifying the knowledge aspects of these traditions.

The article shows that it is possible from a theoretical point of view to map basic ISD activities on the inquiring systems presented by Churchman. The article shows that, despite the recent focus of researchers on the Singerian inquiring systems as the superior system for knowledge creation, human nature and physical constraints prevent the application of this inquiring mode in all phases of information systems development. As a result, the different inquiring systems are opportunistically used in a scattered fashion and provide therefore diversified ways of creating knowledge which in turn require multiple management practices.

The extent to which ISD projects show the use of the inquiring systems has to be further investigated and tested. The other aspect that has to be further investigated is the extent to which inquiring modes may be influenced by problem solving modes. Indeed, the inquiring system mode at each phase may depend on the reasoning mode adopted by decision-makers to solve their problem. The use of a problem solving mode is determined by the context in which the decision is made, i.e. the problem characteristics, the decision environment and decision-maker’s characteristics [23]. The hope is that the issues raised in this article will prompt other researchers to pursue this effort in their future endeavors.

Given the history of successes and failures that characterizes ISD there is the need for a better understanding of the multiple dimensions that influence ISD projects. Dealing with new technology [11], novel organizational structures [20] and novel ways of working [5, 16] further increases the need to revise our knowledge of how ISD projects should be carried out including and highlighting new ways of seeing this process. The hope is that this article contributes to justify the need for further developing our understanding of knowledge creation activities at individual level, group level and their interaction to understand the process, to design better ones and to recommend managerial practices that are geared towards the recognition and the improvement of inquiring processes.

REFERENCES