A Cybernetic Theory of the Impact of Implementers’ Actions on User Resistance to Information Technology Implementation

Suzanne Rivard
HEC Montréal
Suzanne.Rivard@hec.ca

Liette Lapointe
Desautels Faculty of Management
McGill University
Liette.Lapointe@mcgill.ca

Abstract

This paper focuses on the information technology (IT) implementers’ role in the dynamics of user resistance to IT implementation. We adopt the notion of cybernetic control and conceptualize an IT implementation as a limited system made up of users and an IT application. We propose a cybernetic theory wherein the implementers are the control device, and their objective is to keep the intensity of user resistance at a level that is acceptable from an organizational point of view. To this end, implementers engage in various actions in response to user resistance behaviors. These actions constitute the feedback sent to the system by the control device. The theory posits that some implementers’ actions have a negative feedback effect and maintain user resistance within an acceptable range. Other implementers’ actions have a positive feedback effect that will lead to significant organizational disruption, which may ultimately require the abandonment of the IT implementation.

1. Introduction

The management of information technology (IT) implementation projects is often depicted as a set of “management controls needed to impose discipline and coordinate action to ensure goals are met” [31, p. 159]. Several researchers have adopted a control perspective to study the management of IT projects [12, 18, 30, 31]. The term “control” is used here as it is defined in management research, which conceptualizes organizational control as a set of mechanisms aimed at ensuring that an organization moves toward its objectives [32]. In an IT implementation project, the implementers – who are either managers of organizational units or IS professionals – are responsible for exercising control so as to ensure that the organization’s objectives for the project are met.

In a review of this research stream, Kirsch [18] comments that researchers have “typically targeted pre-defined controller-controllee dyads, emphasized control relationships within IS units, studied control modes needed to achieve preidentified project goals such as on-time and within-budget system delivery, examined modes of control singly rather than simultaneously, and taken a ‘static’ or ‘snapshot’ view of control” [18, p.374]. She also deplores the fact that although these studies have produced important insights, they are limited in terms of how well they capture control in a setting that is non-routine, complex, and dynamic.

In this paper we conceptualize control in IT implementation projects in a way that departs from how it has traditionally been examined by IT researchers and that addresses the concerns raised by Kirsch. We: (1) propose a theory of control relationships as they evolve outside IS units, (2) theorize on how control modes influence resistance behaviors rather than the project time-line or its budget, and (3) acknowledge the dynamic nature of the environment in which control is exercised. More specifically, we propose a theory of the dynamics of implementers’ control on user resistance during an IT implementation. Our theory is based on the General System Theory (GST); it conceptualizes implementers as a control mechanism whose objective is to maintain the level of user resistance within an acceptable range from an organizational point of view.

In IT research, although resistance is identified by many as a key concern during IT implementation, relatively few authors have specifically studied the phenomenon. Moreover, the small number of models proposed to explain user resistance are user-centric in nature, in that they include antecedents that are closely related to the immediate user environment: either the users themselves, or their immediate work system [14]. By focusing on the implementers’ actions as control mechanisms, our study goes beyond the immediate user environment and provides a better...
understanding of the relationships between implementers’ actions and user resistance.

2. Theoretical background

The theory that we propose is grounded in the domains of user resistance to IT implementation and cybernetics.

2.1 User resistance to IT implementation

Resistance to IT implementation is said to occur when users feel threatened by the system being implemented or by its effects on their environment. Resistance materializes in a variety of behaviors that can be covert (e.g., being passively uncooperative [23]) or overt (e.g., attacking the credibility of the implementers [34] or voluntarily committing errors when using the system [9]). Resistance behaviors can be classified in four categories that differ in terms of the intensity of the resistance: apathy (e.g., inaction and lack of interest toward the system being implemented), passive resistance (e.g., delay tactics, excuses, persistence of former behaviors), active resistance (e.g., voicing opposite points of view, forming coalitions) and aggressive resistance (e.g., engaging in sabotage, infighting and making threats) [19].

Although user resistance is sometimes portrayed as a barrier that ought to be removed, we espouse the view, advocated in previous research, that resistance is neither good nor bad [13, 19, 23, 24, 25]. Indeed, there are times when resistance is a means by which users communicate the fact that problems exist with the IT being implemented or with its effects; in such instances, resistance is functional. At other times, when it prevents the adoption of an IT that could benefit the organization, resistance becomes dysfunctional. In either case, implementers have to deal with this resistance.

In previous studies, resistance to IT is deemed a complex phenomenon that cannot be reduced to a simple rejection of a new technology and has been conceptualized as the result of a complex interaction among a number of antecedents. A number of authors have proposed models to explain how resistance to IT implementation develops. For instance, Markus [24] portrays resistance as resulting from the interaction of system features with the intraorganizational distribution of power. Hirschheim and Newman [13] state that the causes of resistance are multiple and diverse, and that they occur “in a tangle of different threads” [13, p. 400]. Joshi [16] proposes a model that posits that resistance stems from negative users’ assessments of the fairness of the exchange between their inputs and the outcomes of their interaction with a given technology. Marakas and Hornik [23] argue that passive resistance misuse is situational, the result of the interaction between the uncertain conditions created by the introduction of a new system and individual traits. Martinko, Henry and Zmud [25] suggest that resistance to IT depends on the interaction of a number of factors: internal and external influences as well as the individual’s prior experience with the technology. Lapointe and Rivard [19] conceptualize resistance to an IT implementation as behaviors that occur following perceptions of threats associated with the interaction between an object and initial conditions. Finally, Ferneley and Sobrepererez [9] suggest that resistance can be either positive or negative, and that it often manifests itself in user workarounds, or deviations from set procedures. They propose a dynamic model in which four antecedent conditions play a key role: enforced proceduralisation, organizational and personnel issues, discipline, and non-engagement with the system. The authors suggest that any of the four conditions can lead to resistance, be it positive or negative, which in turn may result in other kinds of workarounds.

The literature is quite sparse in terms of identifying implementers’ actions in response to user resistance and of the effect of these actions on the level of intensity of the resistance. Indeed, in reviewing the literature, we did not find any study that focused on the implementers’ reactions to user resistance to IT. We nevertheless identified a number of such reactions that authors mentioned in discussions of other aspects of resistance to IT.

The vast majority of IT implementers’ reactions to user resistance mentioned in the literature pertain to reactions intended to improve the situation in a supportive manner. We grouped these reactions under the label remedial reactions. They include actions such as unearthing the causes of resistance and determining which corrective actions can be undertaken [16, 22]. Remedial reactions also include efforts to divert users from exhibiting resistance behaviors. Some strategies are aimed at changing individuals’ perceptions of the system being implemented or its environment [25] through training, communication, and fair procedures [16]. Strategies to influence users’ attitudes have also been considered [27]. They include: reciprocity – granting a favor to users and expecting that corresponding advantages or privileges will be returned; commitment and consistency – binding users by making them assume a position that is aligned with
the desired behavior; social proof – showing users that prominent others have already accepted the IS; liking – putting esteemed individuals in positions of responsibility in the implementation process; and scarcity – making information about the system scarce and granting users privileged access to it so as to induce positive feelings towards the IT. Other remedial actions are aimed at modifying: the system being implemented (e.g., IT redesign or restructuring [25]), the user environment (e.g., adding temporary, awards and job reclassification [16]), or some characteristics of the users themselves (e.g., training to reduce learning effort and frustration [16]).

The literature suggests that IT implementers will sometimes exert pressure on users to force them to stop resisting. We grouped these reactions under the heading of antagonistic reactions. They include actions such as isolating pockets of resistance to prevent resistance from spreading “into a full-blown mutiny or coup” and forcing implementation [22, p. 1302] through authority – using formal power to ensure user compliance [27]).

We created a third category of IT implementers’ reactions, which we labeled lack of reaction. Although this type of reaction is not discussed much in the literature, Lauer and Rajagopalan [22] hint at such inaction in their discussion of acceptance of and resistance to IT. Also, Lapointe and Rivard [20] refer to management’s lack of response to user resistance to an IT implementation when they describe situations in which implementers explicitly chose to ignore users’ complaints about system features that they had deemed inappropriate.

2.2 Cybernetic systems

Cybernetics is part of General Systems Theory (GST), whose objective is twofold: “[GST] seeks to classify systems by the way their components are organized (interrelated) and to derive the ‘laws,’ or typical patterns of behavior, for the different classes of systems” [5, p. xvii]. Although GST dates back to the 1940s [40], it is still considered an appropriate theoretical foundation for studying organizations in general [37] and IT in particular. For instance, some researchers have used mechanistic, organic, and colonial systems metaphors to explain resource allocations to the IT functions of large firms [33]. Others have used the basic concepts of GST to theorize on the formation and value of IT-enabled resources [28].

Boulding’s [4] typology of systems is widely acknowledged in GST [37]. In this typology, systems vary according to the level of complexity of their parts and the nature of the relations among the parts. The typology organizes nine types of systems within a hierarchy of complexity. While each level of the hierarchy is distinct from the others, systems are not mutually exclusive because each higher-level system incorporates the features of lower-level systems.

As illustrated in Figure 1, the level of least complexity is that of frameworks, which correspond to static structures, such as the anatomy of a human or animal or the arrangement of atoms in a crystal. The second level, clockworks, is the level of simple dynamic systems with predetermined motions, such as the lever and the pulley. The third level, cybernetics, represents systems “capable of self-regulation in terms of externally prescribed targets or criteria such as a thermostat” [37]. The fourth level is that of open systems, which are characterized by self-maintenance and self-reproduction; they are exemplified by the living cell. The fifth level, blueprinted-growth systems, are systems that reproduce through preprogrammed instructions for development; they are typified by the plant. The sixth level is the animal level, which is characterized by mobility, goal-seeking, and self-awareness; it is the level at which animals function. The seventh level is the human level; in addition to the characteristics of the lower levels, this level has the characteristic of self-consciousness. The eighth level is that of social systems that comprise several human actors who share a common social structure. Finally, the ninth level is that of the transcendental systems; it is the level of the unknowables, the ultimates, and the absolutes [4].

Figure 1 – Boulding’s Typology of Systems
Our theory is situated at level 3, the cybernetics level. The contribution of cybernetics has been to link control mechanisms as studied in natural systems to those engineered in man-made systems [6]. Cybernetic systems are characterized by the notion of control, which has as its primary requirement the need to maintain the level and kind of output necessary to achieve the system’s objectives [15]. More specifically, control consists in stabilizing the outputs produced by a system so that the latter reaches a steady state in which its outputs show only small, random variations around a desired value [7].

As shown in Figure 2 there are four basic elements of control. The first is a characteristic or condition to be controlled: that is, the variable in the system’s behavior that has been chosen to be monitored and controlled [36]. The second element is the sensing function, which involves measuring the value of the characteristic or condition to be controlled. The third element is comparing, which consists in weighing the value of the characteristic against the objective in order to determine whether the value falls within an acceptable range [36]. The fourth element is a corrective function, which consists in evaluating the significance of the variation, determining whether the system is under control or out of control, and evaluating alternative corrective inputs that can be fed back to the system to restore stability [15]. Although stability is the long term goal of the control process, authors contend that short-term and periodic instability is necessary for system adaptation and learning [36].

The corrective inputs are referred to as feedback. There are two types of feedback. The first is negative feedback; its effect is to dampen or reduce fluctuations around the desired values of system outputs. The second type is positive feedback; its effect is to reinforce the direction in which the system is moving. Negative feedback leads the system to reach a steady state, while positive feedback may lead to system instability. Indeed, if the output of the system falls outside the range of acceptable values, positive feedback will reinforce the system’s behavior, which may lead to instability and even to the destruction of the system [39]. Note that the terminology employed here may appear counterintuitive. Indeed, negative feedback tends to have a positive net effect in that it contributes to reaching the targeted values for the system outputs. Positive feedback tends to have a negative effect in that it contributes to moving away from the targeted values for the system output.

The ability of the control mechanism to exercise appropriate control over the system is subject to the Law of Requisite Variety [2]. This law states that in order to control the system, the control mechanism has to have as many types of responses as there are states in the system. These responses can be preprogrammed, provided by decision rules, or generated by the control mechanism’s ability to generate control responses [7].
3. A Cybernetic Theory of the Impact of Implementers’ Actions on User Resistance to IT Implementation

As per Gregor’s [10] taxonomy of theories in IS, we propose a theory for explaining and predicting the impact of implementers’ reactions to user resistance behaviors on the intensity of user resistance following the enactment of such reactions.

The focal construct of our theory is user resistance: more specifically, the intensity of user resistance behaviors. We set the boundaries of our theory as follows. Although resistance and acceptance have been said to be at either end of a continuum of IS adoption, an explanation of the antecedents of the acceptance end of the continuum is outside the boundaries of our theory. Indeed, while we recognize that some of the implementers’ actions may have the effect of increasing user acceptance of a system, the proposed theory does not pertain to those relationships. Also, our theory is limited to the effect of a single antecedent of user resistance: the implementers’ actions. Finally, the theory does not aim to explain the mechanism by which implementers’ actions influence user resistance. Rather, it focuses on the resulting level of user resistance.

The theory is dynamic in nature. It conceptualizes an IT implementation as a limited system constituted of users interacting with an IT application – one that is either in the process of being implemented or has already been implemented. As shown in Figure 3, the implementers are the system’s control device, and their objective is to keep the intensity of user resistance within a range that is acceptable for the organization, that is, at a level that does not create organizational disruptions.

As in any cybernetic system, the control device comprises three basic functions: sensing, comparing and correcting. As a control device, implementers have the ability to sense the level of intensity of user resistance behaviors. Whether or not the implementers will accurately assess the level of resistance depends on the acuteness of their sensing function. The implementers also have a comparing function, which assesses the intensity of resistance behaviors against values that are deemed organizationally acceptable. Once again, the accuracy of this assessment is likely to vary with the implementers’ competence or experience. After comparing, the implementers engage in a correcting function, which consists in assessing the scale of the gap between the intensity of resistance and an acceptable level and determining whether the system is in or out of control. In the latter case, implementers will evaluate alternative corrective actions, which will become inputs fed back into the system to restore stability.

Our theory posits that some implementers’ responses to user resistance behavior have the effect of negative feedback; that is, they dampen or reduce fluctuations around acceptable levels of intensity of
user resistance. Other implementers’ reactions to user resistance behavior, however, have the effect of positive feedback; they reinforce user resistance and intensify resistance behaviors, which may lead to organizational disruption and, eventually, to project abandonment.

The relationships between implementers’ actions and the intensity of user resistance behaviors are synthesized in five propositions. In order to provide insights into the relationships between these constructs, we provide examples from the literature, with cases describing IT implementations. Although the main focus of these cases is not the management of resistance per se, the vignettes extracted nevertheless provide some evidence for our propositions.

**Proposition 1:** Implementers’ remedial reactions to user resistance to an IT implementation have a negative feedback effect on user resistance.

In Proposition 1, the implementers’ three control functions (sensing, comparing and correcting) operate appropriately. Indeed, implementers effectively perceive the value of the level of resistance, properly determine whether it is within the range of acceptable values, and decide on the appropriate corrective inputs, which they then feed back into the system.

Implementers’ responses are said to have a negative feedback effect when their impact is to reduce the level of user resistance behaviors. This is what is to be expected when implementers’ reactions to users’ resistance behaviors are remedial in nature. In the literature, we found several case studies that illustrate this negative feedback effect.

A first illustration is that of the case related by Massaro [26] of the implementation of a medical information system at the University of Virginia Medical Center. Following the hospital-wide installation of a system to manage laboratory functions, the medical staff voiced negative responses. Concerned, the developers chose to deactivate the features that the users did not like and worked closely with the medical staff to accommodate their demands, which ultimately resulted in less user resistance.

Another illustration is found in Joshi [17], who reported the implementation of an order management system in a custom furnishing organization. In this case, the production personnel initially expressed reservations about the new system. Indeed, these employees saw the system as an additional burden, which led to an unfavorable initial judgment and resistance. The implementers were able to successfully communicate the value of the system, explaining how crucial it was for the competitive survival of the organization. In the end, the benefits of the system became clear for the production staff, and this led to less resistance.

Finally, Prasad and Prasad [34] present a detailed account of the implementation of a new computer system in a Health Maintenance Organization. Following the introduction of the system, some employees – who feared that the system would be detrimental – tried to disrupt the implementation process by flooding the basement storage room, resulting in the destruction of some display terminals. In response, the managers – wanting to minimize resistant actions – made informal concessions, such as turning a blind eye towards unpunctuality, allowing some employees to leave the organization earlier than usual, and permitting more flexible work schedules. These compromises appeased the employees and helped reduce user resistance.

**Proposition 2:** Implementers’ antagonistic reactions to user resistance to an IT implementation have a positive feedback effect on user resistance.

In Proposition 2, while the implementers’ sensing and comparing functions may be appropriate, the correcting function is deficient, and implementers’ reactions do not operate appropriately. Indeed, the antagonistic reactions of the implementers have a positive feedback effect, in that they reinforce the users’ resistance behaviors. Such are the situations illustrated by the following examples from the extant literature.

A first illustration is provided by the case of the implementation of TransLease, an interorganizational information system designed to facilitate electronic commerce between motor vehicle leasing companies and repair agents [1]. In this case the target users were the repair agents. When the repair agents experienced problems with TransLease, their use of the system declined. In response, the motor vehicle leasing companies decided to employ coercive power and attempted to leverage their buying power to force the repair agents to use the system. As a result, organizational disruption was observed: the relationships between the lease companies and the repair agents degraded. There was a significant increase in the level of distrust between the two groups and increased user resistance.
Another example is provided by Hirschheim and Newman [13] in their Warwick case study, which describes the implementation of a system for policy processing (SPP) at the Cambridge branch of a medium-sized insurance company. With time, the underwriters, who were the target users, stopped using the computerized system and by-passed the changes by continuing to use printed documents. The project manager then decided to impose system use on the underwriters “by removing the top sheet of the underwriting document, forcing her staff to retrieve the information from SPP” (p. 403). This form of coercive strategy led to greater user resistance.

Finally, Newman and Robey [29] also provided an illustration of such a positive feedback effect in the case of the implementation of a student information system at Middlesex State University. Here, the staff of the undergraduate program was criticizing the new system’s features while continuing to use the former system. In reaction, the implementers decided to turn off the old system, hoping that it would force the staff to use the new system. This led the staff to petition a formal complaint to the administration, an indication of greater resistance.

**Proposition 3:** A lack of reaction on the part of implementers’ to user resistance to an IT implementation has a positive feedback effect on user resistance.

In Proposition 3, any of the implementers’ control functions may be inappropriate. In some instances, the sensing function is deficient and the implementers are not aware of the resistance. In other situations, the sensing function is appropriate but the comparing function is not; the implementers are thus not sensitive enough to recognize that resistance does not fall within the range of acceptable values. Finally, there will be some instances in which the sensing and comparing functions will be appropriate but, because the correcting function is deficient, the implementers cannot identify what are appropriate corrective inputs. A lack of reaction to user resistance has a positive feedback effect, in that it reinforces users’ resistance behaviors. As illustrated in the following examples from the extant literature, implementers’ nonresponse to user resistance behaviors will be followed by increased resistance.

This course of action is clearly illustrated by Silva and Backhouse [38] in their case presenting the implementation of a new administrative information system for the management of research project contracts at the Center for the Study of Food and Nutrition Sciences of Central America. In this case, initial mild resistance from the researchers was ignored by the implementers. The case reports that some derisory comments were made by the researchers after the system was implemented, but that no action whatsoever was taken in response. The case goes on to show that after a few months, the researchers were becoming increasingly frustrated with the system. Resistance grew and, with time, the humor turned into criticisms, formal complaints and, eventually, actual threats.

Another example is provided by Wilson and Howcroft [42] in a case relating the implementation of the Zenith Nurse Management System at the North England Eldersite Hospital. After the system became available in the nursing wards, even those who had been enthusiastic about the implementation became disappointed with Zenith. Indeed, it was considered a hindrance, distracting the nurses from their work rather than improving it. Despite a “catalogue of negative opinions concerning the system” [42, p. 243], the implementers did not formally recognize the nurses’ concerns. In the end, the level of resistance grew and the nurses reverted to using the manual files; the system even went unused in many wards.

Yet another illustration is provided in a case relating the implementation of a computerized stock control system in the Barrington’s chocolate factory [35]. When the system was introduced, the operations staff was irritated by its deficiencies. With time, the operators became increasingly frustrated and claimed that they had lost trust in the system. The managers “were bewildered, and felt impotent to rectify the situation” [35, p. 69]. As a result, the level of resistance grew. The operators reverted to their former, manual methods, and some sabotaging of the system was even suspected.

**Proposition 4:** Over the course of an IT implementation, a succession of implementers’ reactions with positive feedback effects will lead to significant organizational disruption and, ultimately, require abandonment of the IT implementation.

This proposition reflects the dynamic nature of our theory. As per this proposition, a single action with a positive feedback effect is not likely to lead to a situation requiring project abandonment. Indeed, under systems theory, although stability is the long-term goal of the control process, short-term and periodic instability may promote system adaptation and learning. For instance, as per Proposition 3, a lack of reaction to user resistance is likely to lead to greater resistance. Yet if, following this increase, the implementers appropriately sense, compare, and correct the situation, the level of resistance may very
well decrease. What Proposition 4 suggests, however, is that if a succession of responses increase resistance, the intensity of such resistance may very well reach a “point of no return,” at which it may be too late to correct the situation, no matter what means the implementers may take.

The literature helps illustrate this situation. For instance, Lapointe and Rivard [19] report the case of a clinical information systems (CIS) implementation in a hospital. When the functionalities related to patient admission and radiology or laboratory requests and results were implemented, some nurses and physicians manifested some reluctance to use the system because they found it difficult to learn and use. The implementers – the hospital administration – did not explicitly address this reluctance and continued to pursue the project with the implementation of the Nursing Notes module (lack of reaction). One particular feature of this new module was not well received by physicians: only they were allowed to enter test and treatment prescriptions. The hospital’s accepted practice – before the CIS implementation – had physicians dictating prescriptions to the nurses. Although the nurses appreciated this new feature, numerous physicians refused to comply, and wanted to continue dictating prescriptions to nurses. Conflicts between the two groups resulted. The administrators took the side of the physicians (antagonistic reaction). As a result, all the physicians signed a petition to have it removed. Ultimately, the module was withdrawn and the hospital director had to resign.

**Proposition 5:** As per the law of requisite variety, implementers who have a wider range of responses to user resistance behaviors are more apt at exerting negative feedback type of actions.

Simply put, the law of requisite variety refers to the fact that, in order for a control mechanism to achieve its objective of maintaining the value of the system’s output within the range of acceptable values, the mechanism must possess responses that correspond to all the potential values of the system’s output. In an IT implementation context, the law of requisite variety suggests that in order for implementers to respond to user resistance behaviors appropriately, they need to have the means to respond to a wide array of resistance behaviors. Proposition 5 suggests that implementers who possess a broader range of responses are more likely to better react to user resistance. For instance, the literature suggests that it is sometimes necessary to react to user resistance by modifying the work environment. This may mean modifying the rewards system, undertaking job reclassification, or placing esteemed individuals in positions of responsibility. This suggests that in addition to adequate sensing and correcting functions, implementers ought to have the legitimacy for implementing the corrective actions.

This is illustrated by Wagner and Newell’s [41] narration of the implementation of an enterprise system (ES) in a university. In this case, the target users were faculty members, whose first manifestation of resistance was a lack of interest in the ES. The implementers involved at this point in time were IS professionals. They did not appear to pay much heed to this lack of interest, and continued deploying the ES. Once the implementation was complete, the resistance became active. The whole university community of users “refused to work with the financial management module” [41, p. 515] and the faculty formed a coalition, demanding meetings with the project staff and the provost to express their discontent. Resistance only diminished after the provost intervened, demanding that the project team negotiate a workable solution with faculty members.

### 4. Conclusion

In this paper we have used Cybernetics – a part of General Systems Theory – to propose a dynamic theory of the impact of implementers’ actions on user resistance to IT implementation. We put forth five theoretical propositions in which implementers are conceptualized as a control mechanism. Our propositions explicate how implementers’ responses to user resistance behavior can have either a positive or a negative feedback effect on levels of resistance, either reinforcing or tempering fluctuations around acceptable levels of intensity of user resistance. To advance these theoretical insights, we propose two potential future research strategies.

One promising research strategy would be to look for rich longitudinal data. These data could first be used with a theory testing purpose. They could also be used to search for new and potentially contradictory or conflicting evidence and as such serve as a basis for developing the theory further [11]. One research method that could detect such rich data is the case survey strategy. Sometimes labelled “case meta-analysis,” this research method requires the systematic selection, coding, and analysis of written case studies [21]. Given the availability of so many rich case studies of IT implementation in the extant literature, a case survey strategy would open a path for the interpretation of empirical data and for the close examination of resistance behaviours,
implementers’ actions and reactions, and the feedback effect of these actions on resistance, as well as the potential identification of new constructs or relationships.

A second promising research avenue would be to use our theoretical propositions as a basis for formal hypotheses on the effect of implementers’ actions on user resistance behavior. Given the system-theoretical foundations of the model, simulations of the systems’ behaviors [3] could be conducted to test the hypotheses. Recently, simulation has been recognized as a methodological approach that is making significant contributions to successful theory development and refinement [8].

The theory that we propose has some limitations that we wish to acknowledge. First, it might be argued that portraying an organizational system as a closed system – “encompassing a set of stable and easily identifiable participants” [37, p.28] – lacks a certain realism. Indeed, we recognize that many actors other than those we have identified in our theory may play roles in IT implementation. We would argue, however, that a given IT implementation project is sufficiently circumscribed that we can indeed identify the key participants, and that the identity of those participants remains rather stable over the course of the project. Second, it might be suggested that a closed system does not appropriately capture the fact that organizations are “open to and dependent on flows of personnel, resources, and information from outside” [37, p. 28]. Although we acknowledge that multiple environmental conditions other than implementers’ actions may have an effect on user resistance – many of which have been identified in extant models of user resistance – we contend that the aim of our theory is to explain and predict the effect of implementers’ actions on user resistance rather than to provide a complete explanation of how resistance emerges and evolves and its source.

Notwithstanding these limitations, we believe that our theory possesses enough explanatory power so as to warrant further investigation.

5. References


[17] Joshi, K. "Understanding User Resistance and Acceptance During the Implementation of an Order


Acknowledgements: The authors would like to thank the Review Team for their useful comments and suggestions. This study was supported by a grant from SSHRC (Social Sciences and Humanities Research Council of Canada).