8

Coordinating Global Software Development Activities

Requisite Variety in Information Systems as a Dependent Variable

Gambel O. Wiredu
Univeristy of Limerick
Limerick, Ireland

Abstract. In this paper, I explain how globally distributed software development subunits can coordinate their activities with information systems (IS). The basis of this explanation lies in the contemporary proliferation of global software development (GSD) activities that suggests an unexplained reality: organizations practicing GSD are somehow regulating their IS to cope with increasing and varied uncertainties. Through an empirical example of an organization's subunit's regulating and coping, I make the case that requisite variety in a subunit's information systems is a dependent variable for managing uncertainties leading to optimal coordination. In this example, I show varied uncertainties that faced the subunit, and I explain how variety in its information system was requisite for managing the uncertainties satisfactorily. Based on these explanations, I suggest four characteristics of variety in IS that will be requisite for managing uncertainties in GSD: developers' agility; developers' continuity and traveling; high frequency of communications; and varied communication modes and technologies.

1 Introduction

It is well known that the increased virtuality of global software development (GSD)–exemplified in the global-distribution of developers, of development processes, of information and of technology–induces new organizational challenges [1, 2, 3, 4, 5, 6, 7]. Usually, this knowledge is derived from comparisons with collocated software development in which resources are not distributed. Thus, although virtual communications are implicit in any modern software development activity, virtuality becomes more pronounced in the face of distribution that engenders spatial and temporal distances.

In software development, as in many other types of work, interdependencies can engender uncertainties [8, p. 565, 9], and uncertainties can undermine interdependencies. This is often the case in GSD in which developers in different locations depend on each other mainly for information because "uncertainties" is, fundamentally, a characteristic of information. Managing uncertainties is, therefore, essential for managing interdependencies; and coordinating GSD activities is, fundamentally, an information processing exercise aimed at managing uncertainties.

New coordination challenges in GSD are implied in the intuition that greater technology-based information processors, due to the pervasiveness of advanced communication technologies, are increasing and varying uncertainties that ironically need more behavior-based information processors for coordination. In other words, software development, an epitome of research and development (R&D), is uncertain and complex enough, wanting for more organismic or behavior-based information processing [10, 11, 12, 13]; yet more software organizations are drawing upon technology-based information processors to globally-distribute their software development activities [1, 7].

This trend constitutes a puzzle that raises awareness to the unexplained reality that many software organizations' subunits are regulating their information systems (IS) to cope with increasing uncertainties that accompany development tasks and their global-distribution. It also suggests that in spite of increased mechanization of information processors through virtualization of GSD, such subunits are somehow able to blend technology-based mechanisms with behavior-based processors to deal with uncertainties within and without development subunits. How do they do these? Stated differently, how do globally distributed subunits make their information systems more capable of coordinating their GSD activities? How they accomplish these is yet unexplained in the IS development literature, hence the puzzle. This paper aims to demystify this puzzle by showing that matching varied uncertainties facing a GSD subunit with varied IS is essential for coordinating activities. I show how Gamma[1] (a subunit of a multinational information technology organization) matched the varied uncertainties facing it with its varied IS to coordinate its activities optimally.

Although a concise definition of an information system has not yet been settled upon by scholars of the IS field [14, 15], there is general agreement within the field that an information system is not just a technical system but rather an interactive and teleological relationship between hardware, software, information, people, and communications – that is, between a technical and a social system [16, 17]. Lee's [17, p. 11] description of this relationship, in particular, is lucid:

> In addition to the information technology comprising the technical system, there is also the organization comprising the social system. Just as there are information requirements

[1] A pseudonym. All names related to the empirical example have been disguised to hide the identity of the organization.

> that the social system poses to the technical system, there are organization requirements that the technical system poses to the social system. . . .
>
> Once the technical system is designed and implemented so as to provide the information required by the social system, the technical system itself would be changed, where the change would then trigger new and different organization requirements for the social system to satisfy. Then, once the social system is designed and implemented as to deliver the organization required by the technical system, the social system itself would be changed, where the change would then trigger new and different information requirements for the technical system to satisfy. These mutually and iterative transformational interactions can be expected to continue without end. Hence whatever results from them is not determinate but emergent.

Using Lee's description interactions, between the various technologies, types of information, modes of communications, and types of developers, denote the IS in Gamma's GSD activities. Based on this denotation, I provide explanations of how varied uncertainties in Gamma's activities were managed by its varied information system with the aim of espousing the idea that a major dependent variable for optimal coordination of GSD activities is requisite variety in information systems.

According to the law of requisite variety "the variety within a system must be at least as great as the environmental variety against which it is attempting to regulate itself" [18, p. 495]. This implies that the variety in IS in a subunit must at least be regulated to match the variety in uncertainties in its internal and external environment. The basis for matching lies in managing IS' capacity to sense, register and respond to the subunit's environment accurately [19, pp. 188-193]. Thus, the requisite variety in IS subsumes their optimal technology-based and behavior-based capacities for dealing with varied information processing requirements within and without the subunit's environment [20, 12, 9]. To avoid an axiomatic treatment of requisite variety [21, p. 307], I premise my explanations on evidence of its characterization of Gamma's IS. Then, I use my explanations to discuss characteristics of requisite variety in information systems and associated coordination functions in GSD. This paper contributes to filling a gap in our understanding of the relationship between coordination challenges in GSD and information systems requirements.

2 Uncertainties, Information Processing, and Coordination

Organizational research literature is replete with constructs of coordination that espouse insights about the problem notably in terms of dependencies [22], interdependencies [23, 19], uncertainties [24, 25, 13] and mechanisms [26]. In spite of the diversity of these insights, coordination can be perceived from two broad perspectives—*processes* and *mechanisms* that are used to manage uncertainties and interdependencies.

Coordination processes denote the pure human and non-physical arrangements and actions. They are conceptualized as “coordination by feedback” by March and Simon [27] and as “coordination by mutual adjustment” by Thompson [23]. Coordination mechanisms are the reified, standardized or crystallized versions of the processes, and they are conceptualized as “coordination by programming” by March and Simon [27] and “coordination by plan” or “coordination by standardization” by Thompson [23].

Mechanisms are formalized, impersonalized, and standardized versions of processes in the form of “pre-established plans, schedules, forecasts, formalized rules, policies and procedures, and standardized information and communication systems” [13, p. 323]. This suggests that coordination processes and mechanisms are closely interrelated because repetitive and recurring processes can easily be transformed into mechanisms while the breakdown of mechanisms leads to the reformulation of new processes or the amendment of old ones. The manifestations of these transformations and reformulations would reflect the varied uncertainties confronting an organization or its subunit.

Tushman and Nadler [9] argue that uncertainties in an organizational subunit come from three main sources: the subunit’s task characteristics, its task environment, and the task interdependencies between itself and other subunits in the organization. They proposed that information processing, “the gathering, interpreting and synthesis of information in the context of organizational decision making” (614), is the means for managing uncertainties. Their emphasis on task-related or work-related sources of internal and external uncertainty suggests that the nature of work (analyzability and variety) and how it is affected by environmental factors (internal and external) are determinants of the nature of information processing and hence of coordination [see also 28, 29, 12, 30].

Task variety refers to the amount and frequency of exceptional events in work while analyzability refers to the amount of exceptional actions and of time required by workers to deal with work exceptions. Thus, tasks which are characterized by low analyzability and high variety would engender greater uncertainties, and therefore require more behavior-based information processing (human-based communications) while highly analyzable and lowly variable tasks would engender less uncertainties and require more technology-based information processing. Software development, an epitome of R&D, is characterized by low analyzability and high variety, as witnessed in the high emphasis on teamwork, high reliance on developers’ intellect, less routineness, and high degrees of coordination by feedback and by mutual adjustment. This signifies that the sources of task uncertainties would be greater and, perhaps, more diverse in software development [10, 12].

The task environment of software development is also a source of task uncertainty because it is an area that lies outside the control domain of the software development subunit. Customers’ requirements and feedback on prototypes as well as requirements for integration of a final product into a bigger application are typical sources of task-environmental uncertainties. The more dynamic the task environment, the greater and more diverse the uncertainties faced by the development unit.

Closely related to the task environment are inter-unit interdependencies that constitute another source of uncertainties because an organizational subunit's software development outcome normally has to be integrated into a larger application. Because other units' development outcomes also have to be integrated into the application, inter-unit interdependencies are pervasive [12]. When the focal software development subunit depends on other subunits to get work done, the greater the degree of instability on the part of the other units the greater and more diverse will be the degree of task uncertainties.

This understanding of the nature of software development uncertainties, based on Tushman and Nadler suggests that a subunit's task characteristics, its environment and inter-unit interdependencies are all predominant sources of uncertainties in software development. Against this backdrop, Tushman and Nadler proposed "as work-related uncertainty increases, so does the need for increased amounts of information, and thus the need for increased information processing capacity" [9, p. 616]. Furthermore, they argued in harmony with Perrow [29], Van de Ven and colleagues [30, 13] and Daft and McIntosh [28] that greater work-related uncertainties required more organismic or behavior-based coordination modes, as compared with a less work-related uncertainties scenario in which mechanistic or technology-based modes of information processing and coordination would suffice.

The problem with the logic behind the need for more behavior-based coordination modes seems to contradict the reality of GSD because GSD organizing largely displaces the behavior-based information processing required for dealing with increased work-related uncertainties. For example, research on distributed organizing in general and GSD in particular is replete with distribution-related and virtuality-related problems such as inadequate mutual knowledge [31, 32, 33], attribution errors [34, 35], mistrust [36, 37], and ethnocentrism [38], to mention the most notable. The manifestation of any of such human-centered problems in a GSD activity is likely to worsen uncertainties because it will engender conflicts and undermine interdependencies between distributed team members. These problems, nonetheless, allude to a fourth source of uncertainties–*intra-unit (cross-site) interdependencies*–and confirm the logic that greater and more varied uncertainties are prevalent in GSD.

In spite of these, distributed organizing and GSD are proliferating [39, 1, 7]. Given this contradiction between existing logic and the reality, one can believe that in spite of increased and varied uncertainties, GSD subunits are somehow managing them satisfactorily. In other words, even though more technical systems are deployed to support social systems' information requirements [17], the organizing requirements that technical systems pose to social systems are being managed somehow.

The theoretical challenge facing GSD research, however, is that this belief is yet unjustified in terms of nuanced analyses of how the increased and varied uncertainties in GSD activities are being managed satisfactorily. In short, the global-distribution or increased virtualization of software development embodies a puzzle. This paper aims to demystify the puzzle. Through my analysis of the Gamma case, I provide explanations of how requisite variety in IS facilitates the management of increased and diverse uncertainties in GSD. It is hoped these explanations, will contribute to demystifying the puzzle and justifying the belief.

3 Research Setting and Methods

From early March to early September 2006, Gamma, a globally distributed team or subunit within Bork (a multinational information technology organization) upgraded a data mining application (also called Gamma) for remote data collection from its customers' servers. This application contributed to the broader application, Supporter, that at the business or organization level, was aimed at supporting Bork's services to its customers. Several other subunits in Bork, called Release Partners (RPs), were involved in Supporter development, and together they constituted a bigger meta-unit called GammaServ.

Bork aimed to reduce the cost of warranty on its hardware products–4% of 2005 revenue was put in the pot for warranty. Thus, driving down warranty cost was a priority, and supply chain cost and delivery costs had to be managed in this cost reduction. It was hoped that this cost reduction would be achieved through remote connectivity in which automated proactive data mining and diagnosing will manifest in customers' servers. It was also hoped that cost reduction would be achieved by relying on Bork's expertise around the world and on information technology to develop software. Such reliance manifested in the composition of globally-distributed teams in GammaServ with the expectation that developers would engage in both intra-team and inter-team technology-mediated communications to accomplish their tasks.

Gamma was made up of twelve engineers headed by the project manager (PM). Three developers and one Architect were based in Kerry, Ireland. One support person and one developer were based in Watertown, South Dakota (SD), USA. The Technical Lead (TL) and four developers were in Bloomington, SD, and one product release manager was based in San Francisco, California, USA. All twelve engineers reported to the PM who was also based in Kerry in the same work area with the other four. Also, all twelve had been working as part of the Gamma team on earlier versions of Gamma before my empirical study. In April 2004, the team was formed specifically to develop the Gamma application. Thus, during the period of my study, all its engineers had been working together since the team's inception.

The time difference between Kerry and SD is 7 hours, thus there were few overlapping hours of work between the two locations. Gamma's very frequent project meetings were usually held between 3.30pm and 6.00pm Kerry time. When the PM had to interact with SD developers, he usually worked from home (late in the Kerry day) to make use of more overlapping hours. The SD developers were more experienced in developing remote connectivity applications and in agile development than the Kerry developers.

With theory development in mind [40, 41], I adopted an inductive and interpretive approach to my empirical study and analysis. My empirical study focused on understanding how GSD activities are coordinated in the face of uncertainties that are engendered by increased virtualization. I aimed to collect qualitative evidence on the various uncertainties facing Gamma development and ascertain how the subunit managed these uncertainties through its information systems. Because this study was idiographic and required an in-depth analysis [42], it was necessary for me to produce qualitative evidence. That is, the continuity and richness of qualitative evidence was deemed crucial to the validity of the study outcome.

Thus, I collected data through observations (or silent participations) in virtual meetings conducted by the Gamma team, through document and e-mail analyses, through short conversations, and through one long face-to-face meeting with the PM. All the evidence was collected at the Kerry site for the entire application upgrade period (approximately six months). The long face-to-face meeting came first, followed by all of the document analyses, observations, and short conversations concurrently in twenty days out of the six months. These methods were mutually complementary and contributed to collection of rich qualitative data.

3.1 Varied Uncertainties Facing Gamma Development

Note that although these empirical results are categorized under sub-headings to make reading easier, in reality they are less categorical and even overlapping.

Task Characteristics: Gamma development was characterized by complexity, that is, by low analyzability and high variety. On the one hand it was characterized by high variety and a high degree of exceptions and non-routineness, which is attributable to the iterative nature of the software development process. On the other hand, Gamma development was lowly analyzable because developers needed more thinking time and had to depend on the Bloomington developers who had greater experience in remote connectivity applications development. The Gamma team also had to collaborate collectively, in pairs, in threesomes, and so on, to be able to deal with the exceptional character of Gamma. Typical of people working on a R&D task, Gamma developers were usually uncertain about knowable outcomes of the non-routine development process, signifying complexity [8, 13].

Task Environment and inter-unit interdependencies: The PM witnessed that one source of the unstable task environment facing Gamma was the continuous changes in customers' demands. Such changes continuously induced changes in business requirements, and this affected Gamma as well as other RPs because such requirements served as inputs for development. As the PM stated, "business requirements baselines are changing continuously in Bork." This unstableness in business requirements further engendered problems in Gamma's interdependent relations with its RPs. Thus, Gamma's inability to predict the changes in the state of business requirements was a typical instance of task-environmental uncertainty; and this translated into uncertainties in inter-unit interdependencies.

According to the PM, inter-unit interdependencies:

> between Gamma and release partners (RPs) [was] not that good; each partner [had] a different motive; commitment from them [was] not certain; engagement with them [was] continuous but the business requirements [could] be changed by a RP arbitrarily; there [was] competition for shared resources by RPs; interdependencies [were] not smooth at all.

It is also interesting to note that these release partners were operating from locations such as India, Brussels, parts of the USA outside of South Dakota, and Britain. The spatial and temporal distances between them worsened the unsmooth interdependencies between Gamma and its RPs. Unsmooth inter-unit interdependencies constituted an instance of unstableness (uncertainties) in the source of inputs for Gamma development: the developers' coding had to align with other RPs' coding to facilitate smooth integration of their efforts to make Supporter a success.

A significant variation that was related to constantly changing requirements in Gamma concerned the highly critical nature of eleventh-hour changed requirements. In the early days of development, changing requirements were easier to deal with because there were enough time resources at developers' disposal. On the contrary, when the release was approaching, it was more difficult to deal with changing requirements because of the obvious time limitation. This means that the uncertainty engendered by the changing requirements for Gamma development was more critical when the release was approaching.

Intra-unit Interdependencies: Exceptional actions in resolving Gamma problems manifested in intra-unit interdependencies as witnessed in the numerous one-to-one, one-to-many and many-to-many communications (for example, teleconferences) among Gamma developers. Intra-unit interdependencies that occurred between Kerry and Bloomington developers were predominant because of the differences in experience between both sets of developers and because each developer was working on some specific component that was interlinked with what others were doing. This difference and need for continuous mutual awareness, combined with spatial and temporal distance between these sets, translated into continuous uncertainties on the part of both sets of developers.

In view of these varied uncertainties facing Gamma, the key question is: to what extent was its information system varied, and how requisite was this variety for managing the uncertainties?

3.2 Explanations for Requisite Variety in Gamma's IS

Uncertainties engendered by task characteristics required more collective thinking time and dealing with higher levels of developer expertise. Gamma's response to collective thinking time requirements was to draw on the 7-hour time difference between Kerry and SD to engage in serial analyses of particular problems. The following scenario reflected serial analysis. Kerry developers would work on aspects of the problem while SD developers would be sleeping. When Kerry developers closed from work, SD developers would take over actions on the problem. Then Kerry developers would go to sleep and return to the problem the next day. This continued until the problem was resolved. Thus, the needed time to deal with exceptional actions was enhanced by the temporal distance between Kerry and SD because it facilitated continuous actions on lowly analyzable and highly variable problems. The so-called round-the-clock or follow-the-sun development [39] was in typical display in such scenarios.

Since the source of task uncertainties engendered by inter-unit interdependencies was external. To deal with them required high agility on the parts of Gamma developers. The PM's witness corroborated my observation that developers' response was in drawing upon their agility to deal with these variations. This increase was largely facilitated by the Bloomington developers who were more experienced in agile development. Although Bork's regulations demanded Gamma's adoption of formal methods that entailed less operational costs, Gamma's challenges, and its capacities for agile development within operational cost limits was crucial for dealing with such uncertainties.

Although the increasing criticality of frequently changed requirements close to release time required high agility levels, the challenge also required high degrees of mutual understanding between Gamma's distributed developers. In this respect, the developers' continuous relationship building, since the beginning of Gamma development, had resulted in high mutual understanding, which they exhibited to deal with eleventh-hour changed requirements. Only two Bloomington developers had met the Kerry developers face-to-face, so relationship building, mainly within technology-mediated communications, was the foundation for developing this mutual understanding. Developing mutual understanding is essentially a learning process. For example, the PM lamented about "guys making assumptions" in the early days of the project; and the three Kerry developers added later that they had learned continuously about the preferences of Bloomington developers.

In instances where higher levels of development knowledge were required from remote experts in other Bork subunits, Gamma developers relied mainly on e-mailing, telephone calling, and/or instant messaging to source knowledge to deal with the lowly analyzable problems. The communication mode depended on the nature of the problem and the explicitness of the information required. Typically, developers used instant messaging for very short queries, they used telephone calling for queries that required more time for interactions, and they used e-mailing when the explicitness of the expert's response demanded a corresponding explicit query.

Uncertainties engendered by intra-unit interdependencies required more frequent technology-mediated interactions between the sites. Gamma developers, thus, relied heavily on technology-mediated communications to achieve mutual awareness of the state of the task at all times. Very frequent teleconferencing by all Gamma developers (including the PM), conducted in virtual rooms with desktop sharing and instant messaging, were the predominant mode of such communications. This was complemented by e-mailing, telephone calling, and instant messaging. These communication modes were applied in various times to match parameters such as the detail of information needed; the reckoned length of the communication; whether the communicator wanted the communication to be obtrusive or unobtrusive; the necessary number of people who needed to get the information being communicated; whether the information needed to be stored or not; and whether the communicated issue required an immediate or delayed response.

Teleconferencing was predominantly used because it supported rapid notification of changing requirements, mutual awareness of others' tasks, reduced information overload, and reduced communication redundancy.

The varied measures and facilities that dealt with varieties in uncertainties are distilled partially to show varieties in people, information, technology, and communications–the parts that define the interactive and teleological relationships of Gamma's IS (see Table 2). The presumption underlying this distillation is that variety in each of the parts signifies variety in the information systems they constitute.

Table 1. Various characteristics of communications, information, and technologies

	Obtrusive	**Unobtrusive**	**Persistent**	**Ephemeral**	**Asynchronous**	**Synchronous**
One-to-one	• Telephone • IM	• e-mail	• e-mail	• Telephone • IM	• e-mail	• Telephone • IM
Broadcast	• Instant Messenger (IM)	• Teleconference • e-mail • Bugzilla	• e-mail • Bugzilla (bug management e-mail)	• Teleconference • IM	• e-mail • Bugzilla	• Teleconference • IM
Unobtrusive			• e-mail • Bugzilla	• Teleconference	• e-mail • Bugzilla	• Teleconference
Obtrusive				• IM • Telephone		• Telephone • IM
Persistent					• e-mail • Bugzilla	
Ephemeral						• Telephone • IM • Teleconference

Varieties in people were reflected in three main capacities. First, the Bloomington developers' greater remote connectivity application development experience and greater experience in agile development proved invaluable in dealing with the high degree of exceptional actions requirements that were associated with the low analyzability characteristic of Gamma development. In particular, the experience in agile development was invaluable in dealing with exceptional actions demands that were associated with continuously changing business requirements. Second, variations in people was also exemplified by the developers' continuous relationship building and mutual learning leading to high degrees of mutual understanding over time. Note that continuous relationship building was achieved mainly through technology-mediated learning, and this facilitated their handling of the highly critical requirements change when release was looming large. Third, traveling across the Atlantic even by few engineers was very important both for sustaining high levels of understanding in cross-site interactions and for enhancing team cohesion and collective decision-making.

Varieties in communications were signified by the different communication modes adopted by Gamma developers to facilitate, for instance, their agility. Because variations in communication modes embody variations in people, information and technology, a complete distillation is an almost impossible task. Thus, much of my explanations of varieties in communications and the parameters underlying those varieties would apply to varieties in information, technology and people implicitly. In Gamma development, I discerned four main parameters that defined the varieties in technologies: synchronicity, obtrusiveness, information exchange mode, and information life. Synchronicity is about whether or not communication is concurrent in terms or sending and receiving information. Obtrusiveness is about whether or not communication prompts (aurally and/or visually) the interlocutor about the arrival of information. Exchange mode is about whether communication is one-to-one or broadcast. Information life is about whether information exchanged is persistent or ephemeral. Examinations of each of these parameters with each other produce six 2-by-2 matrices (see Table 1) that help in explicating the characteristics of variations in communications, information and technology used by Gamma for dealing with uncertainties.

The matrices plus the varieties in people's capacities are matched against the varied uncertainties facing Gamma to develop Table 2 which is a precursor to the explanations of requisite variety in IS in the following section.

Table 2: How varied information systems matched varied uncertainties in the internal and external environment

		Uncertainties facing Gamma		
Information Systems Requirements		**Task Characteristics** Task variety Immediacy of query response Availability of interlocutor Traceability of communication Spontaneity of communication Formality of communication	**Task Environment and inter-unit interdependencies** Changing requirements Eleventh-hour requirements	**Intra-unit (cross-site) interdependencies** Varying Communication preferences Mutuality of awareness and knowledge Mutuality of understanding
Part	**Variety**			
People	**Agility and experience**	• Agility and experience address task variety	• Experience increases expectation; agility facilitates resolution of changing requirements • Experience increases expectation; agility facilitates resolution of 11th-hour requirements	
	Continuity and learning	• Continuity and learning increase informal interactions	• Continuity and learning enhances collective agility •Learning enhances more efficient and effective ways of resolving 11th-hour requirements	• Awareness of others' communication preferences • Continuity and learning facilitate mutual understanding through relationship development

Table 2 Continued: How varied information systems matched varied uncertainties in the internal and external environment

Communication Mode (including information and technology)	**Teleconferencing** • Synchronous • Ephemeral • Broadcast • Unobtrusive	• Clarifies task variety • Facilitates immediate response to queries • Induces informal communications • Induces informal interactions	• Facilitates notification and collective discussions to resolve changing requirements • Facilitates task allocations to resolve 11th-hour requirements	• Also has instant messaging, and document sharing and editing facility that facilitates various communication modes • Facilitates task verifications • Brings all engineers to the 'same page' more efficiently
	Normal e-mailing • Asynchronous • Persistent • 1 – 1 and broadcast	• Clarifies task variety • Facilitates problem solving that requires delayed responses • Addresses non-availability of interlocutor • Facilitates traceable communications • Facilitates formal communications	• Supports broadcast of teleconferences scheduled to resolve changing requirements • Facilitates task allocations for resolving 11th-hour requirements in the absence of teleconferencing	• Facilitates mutual awareness at both personal and collective levels • Brings all engineers to the 'same page' less effectively
	Bug management e-mailing (Bugzilla) • Asynchronous • Persistent • Broadcast • Unobtrusive	• Task variety: Broadcasts new bugs, priorities, severities, and assignments • Addresses non-availability of interlocutor • Facilitates traceability • Facilitates formal interactions		• Facilitates mutual awareness of bug fixing, priorities, severities and assignments • Formalizes bug-related information through categorizations; facilitates sorting by categories
	Instant messaging • synchronous • ephemeral 1 – 1 and broadcast • Obtrusive	• Facilitates immediate response • Notifies availability • Potentially facilitates traceable communication • Facilitates spontaneous communications		• Facilitates personal-level mutual awareness • Facilitates personal-level mutual understanding
	Telephone calling • Synchronous • Ephemeral • 1 – 1 • Obtrusive	• Facilitates immediate responses to queries • Facilitates spontaneous communications		Facilitates personal-level mutual awareness Facilitates personal-level mutual understanding

4 Discussion

How can globally distributed subunits make their information systems more capable of coordinating their GSD activities? I discuss answers to this question by evoking, from the above explanations, four characteristics of variety in IS and associated coordination functions that depict their requisite coordination capabilities in GSD environments (see summary in Table 3.

First the *agility of developers* would always ready them for dealing with uncertainties that are engendered by continuously changing requirements and by eleventh-hour changed requirements that are critical. Developers are integral in the social system of a globally distributed subunit's information systems, and their agility will facilitate information systems' responsiveness to such uncertainties. Gamma had an agile capacity because it was constituted by a globally distributed team of developers most of whom were experienced in agile software development. However, bearing in mind that Bork was a large multinational organization that was bent on reducing costs by insisting on formal methods, Gamma developers' agility was not absolute. Rather, agility was exhibited amid the discipline in Bork's required formalisms such as adherence to plans and processes that were key to the organization's cost-reduction strategy. In short, agile development must be balanced with the required discipline [43] to achieve responsiveness to customers' changing requirements and organizational formal requirements.

Second, the *continuity of developers*, in the same development team over a long period, coupled with developers' traveling across sites is fundamental for their mutual learning and understanding and to their relationship building. Mutual understanding is crucial in the GSD context where communications aimed at problem resolutions are technology-mediated. Technology-mediation normally slows down mutual understanding between people, and the process takes a relatively longer period to manifest satisfactorily. Thus longevity of developers in a GSD team will help in achieving high degrees of mutual understanding needed, especially when dealing with eleventh-hour changed requirements. Furthermore, high degrees of mutual understanding will continuously ease communications between distributed developers and enhance their agility, and hence increase the social system's capacity for responding to problems through technology-mediated interactions. Together, these are essential to information systems' capacity for responsiveness to emergent and eleventh-hour requirements.

Third, *high frequency of communications* by distributed developers is necessary for continuous mutual awareness. On the presumption that distributed developers will use various modes of communications to match various contexts and information needs, high frequency of technology-mediated communications will facilitate information systems' accurate sensing of its environment. Accurate sensing is the basis for accurate registration and responsiveness, and these capabilities of a subunit's information systems are particularly necessary when dealing with intra-unit uncertainties between globally distributed developers. The high frequency of communications between Gamma's distributed developers contributed significantly to continuous mutual awareness and responsiveness in the subunit.

Fourth, *varied technologies and communication modes* that will facilitate all obtrusive/unobtrusive communications, broadcast/one-to-one communications, synchronous/asynchronous responses, and/or persistent/ephemeral information are necessary for two main reasons: they represent flexibility in technical systems of information systems, a flexibility in technical systems will enhance developers' (social systems') natural flexibility, and make information systems more capable of sensing and registering accurately the subunit's internal and external environments. In Gamma, these varieties were clearly manifest, and they helped the developers obtain the right information in the right format from SD and elsewhere.

Table 3. Characteristics of IS variety and their coordination modes

Characteristic of IS variety in GSD	**Coordination function**
Agility of developers	Responsiveness to changing requirements and eleventh-hour changed requirements
Continuity of developers	Facilitates developers' relationship building and enhances their agility
High frequency of communications	Accurate registration of task variations in the internal subunit environment
High variety in technologies and communication modes	Accurate registration of variations in technology preferences in the internal and external environment

4.1 Implications

The preceding analysis and discussion show that the variety in Gamma's IS was requisite for matching the varieties in uncertainties in its internal and external environment. They also show the predominance of uncertainties borne of intra-unit (cross-site) interdependence. Their predominance obviously reflects the fact that distance (spatial and temporal) does matter [44]. Interestingly, most of the literature on information processing in R&D does not give any significant consideration to intra-unit interdependencies because previous research has largely dealt with collocated R&D [10, 45, 46, 9]. For example, Tushman and Nadler's [9] very notable information processing model suggests subunit task characteristics, subunit task environment, and inter-unit interdependencies as sources of uncertainties. However, as my explanations in this paper show, it is important to regard intra-unit interdependencies as a main source of uncertainty in GSD and integrate it into their model. Giving regard to this source of uncertainty will make information processing theory more relevant and valuable for analysis of information processing in distributed R&D activities.

One feature of Gamma that distinguishes it from other GSD teams discussed in the mainstream GSD literature is the insignificant cultural differences between Kerry and SD developers as the two sets largely shared the English language and Western values. In the mainstream literature, most GSD teams are constituted by globally distributed developers who have perceived significant cultural differences. There are arguments about what specifically constitutes culture, which is typical in the low-paradigm sociology and psychology fields, and these will undoubtedly affect any discussion of culture. Nevertheless, my analysis and discussion of the usefulness of continuity of developers and mutual learning and understanding for relationship building is applicable to GSD teams with significant cultural differences among distributed developers. It is applicable in the sense that such relationship building processes have to be accentuated in such contexts to reduce uncertainties and conflicts.

The discussion also suggests that other uncertainty-related problems apart from socio-cultural differences in the virtualization of software development through global-distribution can be more dominant. Many socio-cultural problems have been talked about in the GSD literature [1, 4, 7]. Therefore, this discussion brings to ongoing research on GSD the instance of constantly changing requirements from globally distributed release partners as a dominant source of uncertainties. It also confirms two issues about agile development that are being increasingly advocated for GSD by some researchers [47, 48]. First, it is a dependable source of managing uncertainties related to constantly changing requirements because it enhances learning. And second, since the total agility of a subunit depends on continuity of developers and soundly built relationships, it may not satisfactorily manifest in subunits where relationships are not soundly built or are slower to build (for example, when socio-cultural differences are more dominant).

There is, nonetheless, a caveat in the generalizability of requisite IS variety that pertains to the fact that Gamma was a subunit of a large multinational organization. Thus, the nature of uncertainties and hence of information systems in small- and medium-sized enterprise (SME) would, most likely, exhibit different characteristics and functions. For example, SMEs are less likely to afford the variety of technologies that Gamma could afford. They are also less likely to retain staff to enjoy the economies of continuity. Contrarily, developers in SME's are likely to exhibit or embrace greater agility because they are, naturally, more flexible than large multinational organizations. My theorization of requisite information systems variety is, therefore, more applicable to globally-distributed subunits in large software organizations. However, aspects of the characteristics and functions will be useful to smaller organizations.

5 Conclusion

The purpose of this paper was to conceptualize how and why requisite IS variety constitutes a dependent variable for coordinating GSD activities. In the steps to achieve this purpose, and in my aim to avoid treating requisite variety as an axiom, I have shown that global distribution of Gamma's software development activities entailed a variety of uncertainties, that these uncertainties required variety in information systems to manage, and that Gamma's information system entailed the requisite variety. Beyond these steps, I have also shown four characteristics of variety in an IS that are requisite for managing uncertainties in GSD activities, and have drawn a few theoretical and practical implications from these characteristics. This is just one explanation of how subunits can make their information systems more capable of managing varied uncertainties. Future research in this area will be needed to fully justify the belief that satisfactory management of uncertainties in GSD is a reality and a significant cause of GSD proliferation these days.

Acknowledgements

I thank Gabriela Avram, Daniel K. Sullivan, Anders Sigfridsson and Kofi Agyenim Boateng whose comments on earlier versions of this paper have proven invaluable. This paper is a product from GSD research that was funded by Science Foundation Ireland under PI grant 03/IN3/1408C.

References

1. E. Carmeland and P. Tjia, *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce* (Cambridge University Press, Cambridge, MA, 2005).
2. R. E. Grinter, J. D. Herbsleb, and D. E. Perry, The Geography of Coordination: Dealing with Distance in R&D Work, in: *GROUP'99* Phoenix, Arizona pp. 306-315, 1999.
3. J. D. Herbsleb and R. E. Grinter, Splitting the Organization and Integrating the Code: Conway's Law Revisited, in: *ICSE'99* Los Angeles, CA, pp. 85-95, 1999.
4. S. Krishna, S. Sahay, and G. Walsham, Managing Cross-cultural Issues in Global Software Outsourcing, *Communications of the ACM* 47 (4), 62-66, 2004.
5. B. Nicholson and S. Sahay, Some Political and Cultural Issues in the Globalization of Software Development: Case Experience from Britain and India, *Information and Organization* 11, 25-43, 2001.
6. W. J. Orlikowski, Knowing in Practice: Enacting a Collective Capability in Distributed Organizing, *Organization Science* 13 (3), 249-273, 2002.
7. S. Sahay, B. Nicholson, and S. Krishna, *Global IT Outsourcing: Software Development Across Border* (Cambridge University Press Cambridge, UK, 2003).
8. R. L. Daft and R. H. Lengel, Organizational Information Requirements, Media Richness and Structural Design, *Management Science* 32 (5), 554-571, 1986.
9. M.L. Tushman and D. A. Nadler, Information Processing as an Integrating Concept in Organizational Design, *Academy of Management Review* 3 (3), 613-624, 1978.
10. T. J. Allen and S. I. Cohen, Information Flow in Research and Development Laboratories, *Administrative Science Quarterly* 4 (2), 12-19, 1969.
11. J. Hage, M. Aiken, and C. B. Marrett, Organization Structure and Communications, *American Sociological Review* 36 (5), 860-871, 1971.
12. M.L. Tushman, Technical Communications in R & D laboratories: The Impact of Project Work Characteristics, *Academy of Management Journal* 21 (4), 624-645, 1978.
13. A.H. Van de Ven, A. L. Delbecq, and R. Koenig, Determinants of Coordination Modes Within Organizations, *American Sociological Review* 41 (2), 322-328, 1976.
14. I. Benbasat and R.W. Zmud, The Identity Crisis Within the IS Discipline: Defining and Communicating the Discipline's Core Properties, *MIS Quarterly* 27 (2), 183-194, 2003.

15. J.L. King and K. Lyytinen, (ed.) I*nformation Systems: The State of the Field* (Wiley, Chichester, 2006).
16. S. Alter, The IS Core XI - Sorting Out Issues About the Core, Scope and Identity of the IS Field, *Communications of the Association for Information Systems* 12, 607-628, 2003.
17. A.S. Lee, Thinking about Social Theory and Philosophy for Information Systems, in: *Social Theory and Philosophy for Information Systems*, edited by J. Mingers and L. Willcocks (Wiley, Chichester, 2004), pp. 1-26.
18. W. Buckley, Society as a Complex Adaptive System, in: *Modern Systems Research for the Behavioral Scientist*, edited by W. Buckley (Aldine, Chicago, 1968), pp. 490-513.
19. K.E. Weick, The Social Psychology of Organizing (McGraw Hill, New York, 1979).
20. R. Kling, K.L. Kraemer, J.P. Allen, Y. Bakos, V. Gurbaxani, and M. Elliott, Transforming Coordination: The Promise and Problems of Information Technology in Coordination, in: *Coordination Theory and Collaboration Technology*, edited by G. M. Olson, T. W. Malone, and J. B. Smith (Lawrence Erlbaum, Mahwah, NJ, 2001), pp. 507-533.
21. J.P. Walsh, Managerial and Organizational Cognition: Notes from a Trip down Memory Lane, *Organization Science* 6 (3), 280-321, 1995.
22. T.W. Malone and K. Crowston, The Interdisciplinary Study of Coordination, *ACM Computing Surveys* 26 (1), 87-119, 1994.
23. J.D. Thompson, *Organizations in Action* (Transactions Publishers, New Brunswick, NJ, (First published in 1967), 2003).
24. J.R. Galbraith, *Organization Design* (Addison-Wesley Reading, MA, 1977).
25. M.I. Tushman, Work Characteristics and Subunit Communication Structure: A Contingency Analysis, *Administrative Science Quarterly* 24 (1), 82-98, 1979.
26. P. Carstensen and C. Sørensen, From the Social to the Systematic: Mechanisms Supporting Coordination in Design, Computer Supported Cooperative Work, *The Journal of Collaborative Computing* 5 (4), 387-413, 1996.
27. J.G. March and H.A. Simon, *Organizations* (Blackwell, Cambridge, MA, (First published in 1958), 1993).
28. R.L. Daft and N.B. MacIntosh, A Tentative Exploration into the Amount and Equivocality of Information Processing in Organizational Work Units, *Administrative Science Quarterly* 26, 207-224, 1981.
29. C. Perrow, A Framework for the Comparative Analysis of Organizations, *American Sociological Review* 32 (2), 194-208, 1967.
30. A.H. Van de Ven, and A.L. Delbecq, A Task Contingent Model of Work-Unit Structure, *Administrative Science Quarterly* 19 (2), 183-197, 1974.
31. C.D. Cramton, The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration, *Organization Science* 12 (3), 346-371, 2001.
32. R.M. Krauss and S.R. Fussell, Mutual Knowledge and Communication Effectiveness, in: *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*, edited by J. Galegher, R. E. Kraut and C. Egido (Lawrence Erlbaum, Hilldale, NJ, 1990), pp. 111-145.
33. S. Sarker and S. Sahay, Implications of Space and Time for Distributed Work: An Interpretive Study of US-Norwegian Systems Development Teams, *European Journal of Information Systems* 13, 3-20, 2004.

34. M.J. Abel, Experiences in an Exploratory Distributed Organization, in: *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*, edited by J. Galegher, R. E. Kraut and C. Egido (Lawrence Erlbaum, Hillsdale, NJ, 1990), pp. 489-510.
35. C.D. Cramton, Attribution in Distributed Work Groups, in: *Distributed Work*, edited by P. J. Hinds and S. Kiesler (MIT Press, Cambridge, MA, 2002), pp. 191-212.
36. S.L. Jarvenpaa and D.E. Leidner, Communication and Trust in Global Virtual Teams, *Organization Science* 10 (6), 791-815, 1999.
37. S.L. Jarvenpaa, T.R. Shaw, and D.S. Staples, Toward Contextualized Theories of Trust: The Role of Trust in Global Virtual Teams, *Information Systems Research* 15 (3), 250-267, 2004.
38. C.D. Cramton and P.J. Hinds, Subgroup Dynamics in Internationally Distributed Teams: Ethnocentrism or Cross-National Learning, *Research in Organizational Behavior* 26, 231-263, 2005.
39. E. Carmel, *Global Software Teams: Collaborating Across Borders and Time Zones* (Prentice Hall, Upper Saddle River, NJ, 1999).
40. A. Schutz, Concept and Theory Formation in the Social Sciences, *The Journal of Philosophy* 51 (9), 257-273, 1954.
41. K.E. Weick, Theory Construction as Disciplined Imagination, *Academy of Management Review* 14 (4), 516-531, 1989.
42. H. Tsoukas, The Validity of Idiographic Research Explanations, *Academy of Management Review* 14 (4), 551-561, 1989.
43. B. Boehm and R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed* (Addison-Wesley, Boston, MA, 2004).
44. G.M. Olson and J.S. Olson, Distance Matters, *Human-Computer Interaction* 15 (2/3), 139-178, 2000.
45. P.G. Gerstberger and T.J. Allen, Criteria Used by Research and Development Engineers in the Selection of an Information Source, *Journal of Applied Psychology* 52 (4), 272-279, 1968.
46. R. Katz and M.L. Tushman, Communication Patterns, Project Performance, and Task Characteristics: An Empirical Evaluation and Integration in an R&D Setting, *Organizational Behavior and Human Performance* 23, 139-162, 1979.
47. B. Fitzgerald, G. Hartnett and K. Conboy, Customising Agile Methods to Software Practices at Intel Shannon, *European Journal of Information Systems* 15 (2), 200-213, 2006.
48. K. Lyytinen and G.M. Rose, Information System Development Agility as Organizational Learning, *European Journal of Information Systems* 15 (2), 183-199, 2006.

About the Author

Gamel O. Wiredu holds M.Sc. and Ph.D. degrees in Information Systems from the London School of Economics. His current research concerns are the coordination challenges posed by spatial and temporal distances, as well as socio-cultural differences in global software development. Gamel can be reached at gamel.wiredu@ul.ie.