System Development as a Rational Communicative Process

G.E. (Gert) VELDHUIJZEN VAN ZANTEN,
S.J.B.A. (Stijn) HOPPENBROUWERS, and
H.A. (Erik) PROPER
Department of Information and Knowledge Systems,
Radboud University Nijmegen,
Nijmegen, The Netherlands

ABSTRACT

System development is a process in which communication plays an important role. Requirements must be elicited from various stakeholders. But stakeholders also make decisions and must understand the consequences thereof. Different viewpoints must be reconciled, and agreements reached.
An important assumption we make is that all actions in the development process are (or should be) based on rational decisions. The quest for rationality is a driving force behind the communication that takes place within the development process, because it raises issues that may otherwise have remained in the subconsciousness of stakeholders. We zoom in on the role of vagueness in communication, and argue that there are good reasons not to try and formalize things too soon in the development process.
The purpose of this paper is to position our ongoing research, encourage discussion about the assumptions we make, and inspire novel approaches to system development. We work towards a comprehensive theory of rational system development, in which due attention is paid to development processes, communication, and the representations used therein.

Keywords: System development, Architecture, Communication, Rationality, Language, Vagueness.

1 INTRODUCTION

Organizations and their information systems are becoming more and more complex, making it increasingly difficult to control the process of system development, and even more so to make it effective. The *alignment* of information systems to organizations and the work practices therein is regarded as crucial [2]. In a rapidly changing world, where an organizations need to adapt quickly, alignment calls for highly flexible systems. Architectures are recognized as important tools that help control the information system development process and at the same time help achieve better alignment [1]. The potential role of architecture in the development process is underlined by the uses of architectural descriptions, as identified in [7]:

- Expression of the system and its (potential) evolution.
- Analysis of alternative architectures.
- Business planning for transition from a legacy architecture to a new architecture.
- Communication among organizations involved in the development, production, fielding, operation, and maintenance of a system.
- Communication between acquirers and developers as a part of contract negotiations.
- Providing criteria for certifying conformance of implementations to the architecture.
- Providing development and maintenance documentation, including material for reuse repositories and training material.
- Providing input to subsequent system design and development activities.
- Providing input to system generation and analysis tools.
- Supporting operations and infrastructure, configuration management and repair, redesign and maintenance of systems, sub-systems, and components.
- Enabling planning and budget support.
- Preparation of acquisition documents (e.g., requests for proposal and statements of work).
- Enabling reviewing, analysis, and evaluation of the system across its life cycle.

The ArchiMate project [11] advocates a generic architectural approach in which visualization and communication play a central role, alongside analysis. A primary goal of the ArchiMate project is to improve support for the design, communication, realization, and management of architectures. These are recognized as crucial areas in which support (methods, techniques and tools) is still lacking.
In our opinion, architectures should capture the *essentials* of the processes and (computerized) information systems within an organization as well as their (potential) evolution, in relation to the concerns of stakeholders [7]. Essentials concern only that which can be thoroughly motivated in terms of the important goals identified. Capturing essentials requires a thorough *understanding* of processes, systems, and stakeholder concerns. Such understanding also enables the negotiation process that is needed in dealing with different and contradicting concerns of stakeholders.
In this paper, we focus on the *communication* aspect of (information) system development, as this aspect is most tightly coupled with the goals of architecting. The concept of communication also takes an important place on the list of uses for architectural descriptions as discussed in [7]. In our view, understanding is a cornerstone for information system development. It can only be achieved through communication. Please note, however, that our communicative perspective should be seen as *complementary* to the descriptive perspective as taken in traditional approaches.
Although we focus on architecture in information system development, the ideas we present in this paper are more generally applicable. The rational communicative process is an important driving force in any deliberate system design or development activity.
An important assumption we make is that all actions in the development process are (or should be) based on rational decisions, even if this happens only in hindsight. These decisions are to guide the communication processes taking place within the development process. Hence the title of this paper: *System Development as a Rational Communicative Process*.
The structure of the paper is as follows. We start with a brief discussion of some background: communication and rationality. In section 2, the communicative perspective is contrasted to the representational (modeling) perspective on system development. In particular, we will discuss the limitations of the representational perspective that we intend to overcome by adopting a commu-

nicative perspective. In section 3, we discuss rationality and its potential role in the system development process. This is followed by the core of this paper: section 4. In this section we discuss the process of system development and its constituents from a rational communicative perspective. We continue with a discussion of three key issues in system development that come to the fore when taking the rational communicative perspective:

- We believe negotiation and understanding are key to the success of a development process. The rational communicative approach provides a natural starting point for negotiation among, and understanding by, the system's stakeholders (section 5);
- We feel that "vagueness" has a very important role to play in system development, leading to the notion of "just-in-time formality" (section 6);
- A rational communicative approach to system development poses specific requirements for tool-support. We provide a sketch of the functionality that should be provided (section 7).

2 COMMUNICATIVE PERSPECTIVE

Although the principle importance of communication in system development has been widely acknowledged, current practice fails to include concrete measures that actively support communication in line with advanced insights into communicative action [13]. Contemporary lines of thought are mainly concerned with representation, rather than with communicative action. Many current architectural methodologies are based on the IEEE standard 1471-2000 for recommended practices for architectural description of software intensive systems [7], or apply a similar line of thinking.

According to the IEEE standard, an architectural description consists of multiple views. Each view addresses one or more of the concerns of the system stakeholders. The term "view" is used to refer to the expression of a system's architecture with respect to a particular viewpoint. A viewpoint establishes the conventions by which a view is created depicted and analyzed.

In most cases, methodologies based on the IEEE standard ignore the fact that descriptions as such do not *actively* support the much sought after alignment between organizations, work-practices and information systems. The descriptions used may seem to indicate an alignment between the different elements, but this 'alignment' occurs purely at a representational level. The descriptions on their own cannot provide sufficient proof of the alignment of the *real* organization, work-practices and information systems.

We propose to extend the notion of view and viewpoint from a descriptive perspective to a communicative one, looking beyond the representations as such, taking serious the role that they play in the architectural communicative process. The mere existence of a representation does not imply that its meaning has been communicated to the intended audience. Metaphorically speaking, trying to construct architectural representations without thoroughly communicating their underlying agreements and motivations is like writing minutes of a meeting that never took place.

A point that is easily missed from a descriptive perspective is that the architecting process is really a creative and collaborative one, in which understanding and agreement is reached on subjects that emerge from the communication between architects and other stakeholders. The descriptive perspective tends to encourage the idea that the architecture is somehow already "out there" and only needs to be represented; it views architecting mostly as a modeling exercise. Architecture descriptions are the mere *result* of a process of developing understanding, negotiation, decision making, and creation. We are not alone in stressing the importance of the process itself. If we carefully look at the potential uses of architectural descriptions identified in the IEEE recommended practice [7] (as quoted in the introduction of this paper), then this list echoes our claim. Architecting is, therefore, essentially an ongoing process of negotiation, decision making, raising of awareness, etc, that *guides* and *follows* the co-evolution of organizations, work-practices and information systems. Architecting methodologies should support the process rather than just seek the shortest path towards elicitation of some "final representation".

Taking a communicative perspective implies that we view an architecture (description) as a result of a communicative process. In this process, information should flow from (all) stakeholders into the developing architecture. This contrasts the descriptive perspective in which information in the form of architectural descriptions flows towards the stakeholders in the organization. From the latter perspective, it appears as if stakeholder can do little more than validate the concoctions of the architects—assuming that they understand the representations offered to them, *including their consequences*.

In line with this incomplete perspective, viewpoints tend to be designed with the *information need* of a stakeholder in mind. We emphasize that it is at least as important to consider what information is to be obtained *from* the stakeholder. In other words, viewpoints should be designed (and selected) on the basis of the two-way *communication need* associated with them. In general, viewpoints should document the communicative purpose underlying them.

One of our goals is to raise the architect's awareness of the communicative perspective, and in particular some of its consequences. For instance, when some architectural diagram is presented to the board of directors, explained and approved, this tends to create a false sense of agreement: it is naive to assume that everyone in the board has understood the full consequences of what was presented. To make things worse, the diagram may have invoked unintended interpretations. It is extremely hard to create architectural representations that convey *only* the essentials, i.e. neither too much nor too little. It is an architect's duty to minimize unintended communication, and to persevere until communication has actually taken place. This is only verifiably the case when the stakeholder *demonstrates understanding*. Development of techniques to validate such understanding is one of the challenges we see for future research.

3 RATIONALITY

Understanding is deeply entwined with the question of *why* choices are made [14]. Argumentation is often much more directly connected to a stakeholder's concern than the architectural choices as such. Therefore, we advocate a rational approach that emphasizes the motivation behind representations.

In this paper, we view rationality as goal-directed reasoning, in which the overall goal is to achieve maximal results at minimal cost. It implies making decisions based on an assessment of their possible outcome, the likelihood of these outcomes, and their utility with respect to the goals held [9, 10]. This includes assessment of the resources spent in making decisions. Making rational decisions requires estimation of the costs involved in accomplishing goals by certain means.

Rationality, therefore, is a very convoluted principle. Cost estimations can in general be made more accurate by gathering information. The decision to gather information should, however, also be subject to rational contemplation, because the gathering itself also consumes resources. At first glance, it seems that this could go on forever; rational agents contemplating a decision by climbing meta-level after meta-level. However, rationality will

generally shortcut these meta-levels, because at some point the cost of rationalization becomes too high for the expected utility.
Rationality does not mean that every step of the way should be thoroughly underpinned with motivations, consideration of alternatives and deep assessments of possible consequences. What it does mean is that decisions are made with goals and the trade-offs between them in mind. The underpinning of the system development process should only go as far as is profitable for the goals at hand. It is important to take into account as much information as is possible but only if it is relevant to the decision, and only if it can be obtained at acceptable costs.
Information is relevant to a decision if it has the potential of changing the decision's outcome. Thus the relevance of new information depends on which information is already taken into account, and on the strength of the motivation with which the decision can be taken on the basis of the information already available. If one of the options in a decision has a much higher utility than the alternatives, then it is unlikely that the outcome of the decision will change on the basis of new information.
In sum, rational system development should lead to sensible trade-offs, for instance between costs and advantages of requirements traceability [5], or between provable correctness through formal specifications and the ability to communicate with stakeholders.

4 THE PROCESS

We will first describe the system development process irrespective of the communicative perspective. Thus, we supply the basis to which we can add communication in section 5. Our initial description of the process is rather neutral, although it already contains some devices which allow for extensions to rational and communicative principles.
The system development process can be seen as the interweaving of three main flows; the *why*-flow, the *what*-flow, and the *how*-flow—typically aiming at construction or formulation of business goals, requirements, and design, respectively.
The distinction between the flows is motivated by the need to become aware of the distinction between essential and arbitrary aspects of design artefacts. Aspects in the *how*-flow that are not motivated by the *what*-flow or the *why*-flow are unlikely to be essential. The same holds for *what*-aspects unmotivated by the *why*-flow.
Representationally, each flow contains statements about the system under development (or on a meta-level about the development process). Typically:

The *why* flow contains statements about business or organizational goals;

The *what* flow contains statements about requirements related to these goals;

The *how* flow contains statements about design decisions made to optimize conformance to these requirements.

We assume that the complete history of each flow is accessible. The flows are logically ordered (from 'high' to 'low'), but they progress in parallel.
The system development process consists of several types of action, extending the flows by changing the set of statements therein. Within each flow we have actions of refinement, backtracking, and assumption.

Refinement is the modification of earlier statements in the light of new insights, and the adding of detail. Refinement does not change the intention of statements; it corrects them, or makes them more precise.

Backtracking is the taking back of earlier statements, to open up alternative lines of development.

Assumption actions introduce statements without any logical connection to statements already present.

The interweaving of flows is also achieved through actions. The higher flows provide the rationale for the lower levels. This gives rise to two additional types of action:

Solution generation involves the descending from one flow to another. Various alternative solutions may be generated, so each flow may have several alternative solutions developing in parallel.

Selection involves a choice is made between these alternatives on the basis of criteria originating from the higher level flow.

In figure 1, the system development process with its flows and actions is depicted.

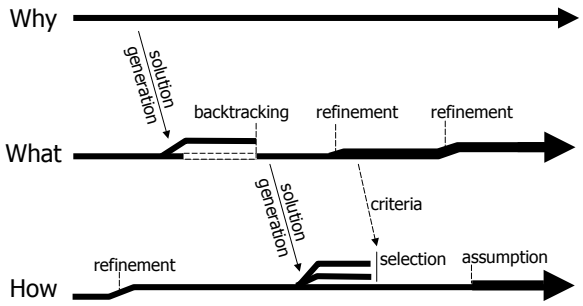

Figure 1: The system development process

The rationality is in the relations between the flows—*what*-statements are added *because* certain goals (*why*-statements) are identified. Understanding rests on the awareness of links between statements in the flows. This can be represented by adding argumentation links between statements in the various flows.
The importance of design rationale, argumentation, and contribution structures is stressed in the literature on requirements tracing (e.g. [8]). In this paper, we extend this view by not only looking at the automated system and its documentation, but also at the socio-technical system it is part of [3], including the mental states of the various stakeholders within it and the communication between them.
Rationality also plays an important role on the level of the development process. During the process, resources are spent. These resources must be put to good use, leading to maximal results at minimal costs—our adage of rationality.

5 NEGOTIATION AND UNDERSTANDING

In the description of the system development process so far, we have abstracted from the negotiation that is needed to come to an agreement. From a communicative perspective, it is important to note that everyone involved in the process has their own view and interpretation of the process—their own idea of what is in each flow. The views of different stakeholders are not necessarily in agreement. The mere existence of well-documented representations of the system under development does not guarantee communication. The representations and the system under development do, however, serve as a medium for communication between stakeholders.

By now, it should be evident that communication is an essential part of the system development process. Yet communication takes time, the spending of which is generally not considered to be the primary goal of system development. Therefore, communicative actions should also be scheduled in a rational way.

The main goal of system development is often taken to be the construction of some automated system. For a rational approach, however, it is important to identify communicative goals as well as constructive goals. Furthermore, these goals should be subordinate to higher level business or organizational goals.

Communicative goals are, for instance:

- to achieve agreement between stakeholders,
- to obtain commitment from stakeholders to supply the necessary resources,
- to validate parts of the system,
- to elicit information from stakeholders, and
- to define concepts in which requirements can be expressed.

The ultimate goal of system development is to change the business or organization in which the system is introduced. To bring about this change, however, it is not sufficient to make available a developed artefact. It requires that people in the environment (stakeholders) are aware of the system's implications, that they understand the system's relevance to their concerns, and that they are committed to its implementation and operation. They will have to change their way of working. So, an essential ingredient of system development is that stakeholders' attitudes, beliefs and perceptions change in the course of the development process. This *requires intensive communication* during that process.

Let us demonstrate the influence of the communicative perspective with an example. When a stakeholder formulates a requirement, various solutions meeting that requirement may be generated. When these solutions or consequences thereof are presented to the stakeholder, some of the alternatives may strike stakeholders as highly unwanted, even though no requirement excluding them had been formulated. Investigation of this situation may lead to the formulation of new requirements of which the stakeholder was previously unaware.

6 VAGUENESS

6.1 THE ROLE OF LANGUAGE

In conjunction with the development of a system, a language emerges that supports the communication about the system. Concepts are adopted and defined, and acquire an increasingly precise meaning during the development process.

The language that stakeholders initially use to communicate about the system domain is, in general, too 'fluid' to base the construction of computerized systems on. Concepts, functions, and entities have to be precisely defined. For example, modeling languages like UML aim at identifying the main concepts that make up the system domain, and the relations between those concepts. A common misconception is that diagrams define the meaning of concepts. Diagrams merely define the (static and dynamic) structure of concepts. The meaning of concepts is for a large part attached to the words naming them. This can easily be demonstrated by removing all words from the diagrams [6].

Meaning is essentially subjective. Formal techniques and diagrams transfer meaning to an objective world in which only pure structure remains, and where the relation to the (subjective, personal) concerns is lost. This implies that even in formal methods the statements expressed are vague.

6.2 THE MERITS OF VAGUENESS

Descriptive thinking generally opposes vagueness because it is seen as the opposite of clarity, and thus undermines understanding. Seen from a communicative perspective, however, vagueness is an instrument which invites further refinement *if and when the time is right*. In other words, rational system development implies just-in-time formality. It helps in scheduling the communicative actions of the architecting process. This means that vagueness enables us to make motivations as explicit as is considered helpful, so that one indeed concentrates on essentials. Achieving precision remains an important drive within the architecting process. Vagueness, however, has its merits, and should not be banned for the wrong reasons.

In line with the above, consider the following. When asked, architects in business environments name general purpose office tools, such as word-processors and generic drawing tools, as their primary means for architecture support. This fact is usually noted as a weird phenomenon that is symptomatic of the immaturity of the architecting community. It is assumed that in a more mature situation, formal and semi-formal modeling tools would be used. Although from a software development point of view, formal modeling has clear advantages, this assumption misses an important point raised by taking the communicative perspective. Office tools are popular because they supply a freedom of expression that is missing in more formal tools. The expression of a vague intuition opens a dialog in which new and more refined ideas emerge through communication. Vagueness therefore is a very effective means for creating understanding, i.e. it is essential for *real* communication. The conclusion is that architecting is chiefly done in between the creation of representations. This reduces representations to means of architectural development instead of main results. The main results are understanding and agreement between stakeholders.

In sum, traditionally and misguidedly vagueness is seen as a problem. In our approach, it is embraced as an invitation for refinement *when required*, and thus a driving force behind efficient and effective communication within the architecting process.

It is interesting to contrast the notion of vagueness with that of abstraction. Both are means to deal with complexity. Abstraction, however, presupposes the existence of something to be abstracted from, while vagueness is a communicative mechanism that presupposes that partial communication is sufficiently clear.

7 SUPPORT

As part of the ArchiMate project, architectural support tools are being designed in which the communicative perspective is ingrained. We envisage tools that support architecting as a communicative process. A central architecture repository is used to reflect the common ground [12, 4] in the negotiation and communication process. This repository contains architecture models, views, and visualizations, together with, importantly, statements concerning their underlying rationale and communicative function.

To support the communication process, the tools include a goal-driven mechanism based on principles of rationality. Perceived inconsistencies, misunderstanding, disagreement, and missing information can be signaled to the architect. These signals are collected in a device not unlike a to-do-list. The to-do-list contains suggestions for communicative action aiming to inform some stakeholders, initiate negotiations between them, or elicit missing information from them. The items in the list refer to representations in the repository.

Communication with stakeholders may not immediately yield definite and precise results. Nevertheless, even if tentative or im-

precise, such stakeholder contributions are significant and should be incorporated in the architecture repository. Therefore, the tools incorporate devices which allow vague expressions. Mere pieces of prose and sketches can be stored in the architecture repository, alongside formal diagrams.

Apart from storing representations themselves, the repository does some bookkeeping to track the communicative status of those representations. The communicative status tells us which stakeholders have agreed upon which aspects of the representations. Communicative status is an important parameter for architecture visualization. Care should be taken to present tentative models in a way that shows their tentativeness, for instance, by making them look like hand drawn sketches. This will invite stakeholders to ask questions and give feedback. Diagrams that are too neatly drawn may give the false impression that the presenter highly values the particular representation, and that comments will not be welcomed. In many cases, however, the diagram will be shown for the explicit purpose of inviting feedback.

8 CONCLUSION

We have described system development as a rational communicative process. In an approach that focuses merely on representations and modeling, some important points concerning the system development process are easily missed.

Reconsidering the system development process from a rational and communicative perspective has led to a number of conclusions:

- Architecture is about defining the essentials of an enterprise, business process or organization. Sifting the essentials from random variations requires understanding of the goals behind the choices.
- Rationality in the system development process implies striving for an overall cost-benefit analysis.
- Communication is essential for reaching understanding and agreement covering all relevant concerns and stakeholders. Rationality in system development requires communication, but also communication should be approached rationally.
- Modeling choices are often made without full comprehension of their consequences. Unintended side effects may start living a life of their own. In particular in architecture, the distinction between essential and arbitrary choices is crucial.
- Commitment of stakeholders is entwined with their understanding of the decisions made within the process.
- Viewpoints should be defined with respect to their communicative purpose.
- For negotiation it is important to identify and communicate the goals and motivations underlying options.
- A language is created within the system development process; concepts acquire an increasingly more precise meaning as the process develops.
- Representations are not the only product of architecting, agreement and understand are at least as important.
- Vagueness is not merely something to be condoned, it is a vital element in any communicative process that works towards real understanding. Vagueness is needed to enable rational communication as well as creative design.

Further development of the ideas presented in this paper, and validation thereof is called for. We plan to work towards clear and well formulated hypotheses in the shape of a formal theoretical model incorporating principles and strategies of rational communication. We intend to achieve empirical validation by analysis of real architecture and design efforts. In addition, we see possibilities for controlled experimentation with respect to the occurrence of the various action types such as backtracking, refinement and assumptions in design conversations.

9 ACKNOWLEDGMENT

This paper results from the ArchiMate project, a research initiative that provides concepts and techniques to support architect in the visualization, communication and analysis of integrated architectures. The ArchiMate consortium consists of ABN AMRO, ABP, the Dutch Tax and Customs Administration, Ordina, Telematica Instituut, Centrum voor Wiskunde en Informatica, University of Nijmegen, and the Leiden Institute of Advanced Computer Science.

REFERENCES

[1] L. Bass, P. Clements, and R. Kazman. **Software Architecture in Practice**. Addison Wesley, Reading, Massachusetts, 1998. ISBN 0201199300.

[2] B.H. Boar. **Practical steps for Aligning Information Technology with Business Strategies**. Wiley, New York, 1999.

[3] P. Checkland. **Systems thinking, systems practice**. Wiley, New York, 1981.

[4] H.H. Clark. **Arenas of Language Use**. Univ. of Chicago Press, Chicago, 1992. ISBN 0-226-10781-7 (cloth); ISBN 0-226-10782-5 (paper).

[5] R. Dömges and K. Pohl. "Adapting Traceability Environments to Project-specific Needs". **Communications of the ACM**, 41(12):54–62, December 1998.

[6] J.J.A.C. Hoppenbrouwers. **Conceptual Modeling and the Lexicon**. PhD thesis, Tilburg University, 1997.

[7] IEEE Computer Society. **IEEE Standard 1471-2000: Recommended Practice for Architectural Description of Software-Intensive Systems**, October 2000.

[8] M. Jarke. "Requirements Tracing". **Communication of the ACM**, 41(12):32–36, December 1998.

[9] H. Raiffa and R. Schlaifer. **Applied Statistical Decision Theory**. MIT Press, 2000.

[10] S.J. Russel. "Rationality and Intelligence". **Artificial Intelligence**, 94:57–77, 1997.

[11] The ArchiMate consortium. **Website of the ArchiMate project**. Telematica Instituut, Enschede, the Netherlands, http://www.telin.nl/NetworkedBusiness/Archimate/, 2003.

[12] D.R. Traum. **A Computational Theory of Grounding in Natural Language Conversation**. PhD thesis, University of Rochester, Rochester, New York, 1994.

[13] V.E. van Reijswoud. **The Structure of Business Communication**. PhD thesis, Delft University of Technology, 1996.

[14] E.S.K. Yu and J. Mylopoulos. "Understanding "Why" in Software Process Modelling, Analysis and Design". In **Proc. 16th Int. Conf. Software Engineering**, Sorrento, Italy, May 16–21 1994.