Liet. matem. rink, **45**, *spec. nr.*, 2005, 180–189

The differences of B2C and B2B workflow exception handling

Dovilė VOJEVODINA, Genadijus KULVIETIS (VGTU)
e-mail: d.vojevodina@erp.lt, genadijus_kulvietis@gama.vtu.lt

Abstract. As business competition is getting stronger companies are seeking good relationships with their clients through the Internet technologies. The type of this relationship (B2C or B2B) usually depends on the business strategy of this company. Using e-business technologies companies expect quick and qualitative contact with clients. Still, the quality of this solution depends on many aspects, but the main is application performance. The medium for these applications is Internet and it is not always reliable through all the engagement process. This paper is dedicated to the distinctions of B2C and B2B e-business application exception handling in a user friendly way.

Keywords: e-business, B2C, B2B, workflow, exceptions.

1. Introduction

The main purpose of any e-business is to bring together clients/partners and companies/suppliers into one medium with a vision to improve business relationships and performance. The main instrument in any business is timely and qualitative information. In this context, e-business enhances information demands giving people proper tools, in this case, internet and internet technologies. These technologies can slightly differ if used for B2C (business-to-consumer) and B2B (business-to-business) cases. The definitions of B2C and B2C are given in Definition 1 and Definition 2.

DEFINITION 1 (B2B). B2B *application focus on using the Internet and/or extranet to improve business-to-business partnerships and transform inter-organisational relationships* [8].

DEFINITION 2 (B2C). B2C *application is for serving customer needs as they exchange information, goods and services over the Internet* [2].

Basically e-business applications consist of many tasks, which purpose is to generate and manage information and supply the right people with this information. The chain of performing these tasks or the need of outside system integration into e-business application also depends on e-business type. B2C and B2B applications are both modelled using workflow technologies (the definition of workflow is given in Definition 3), which main distinction is inter-organisational relationships, which are most common in B2B applications. The other difference is user interface. In B2C applications users’ access needed information simply through the web pages. In B2B, the special internet applications are build for business partners, which gives special and selected information for every partner separately (sometimes such applications can even be installed as local/offline systems at partner place).

DEFINITION 3 (workflow). *The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules* [5].

Still, to this day, all internet technologies lack the reliability and trust of the consumer. These aspects focus on two main problems: (1) internet connection reliability (performance and correct information presentation) and (2) the lack of real world situations management in systems (unpredicted business cases, ad-hoc workflow management). Such cases are usually called workflow or business process exceptions (the definition of exception is given in Definition 4) and mostly managed/solved on-demand. For this moment, all of the existing workflow management systems lacks of these features, because they usually depend on business context and specifics.

DEFINITION 4 (exception). *Any business activities that can not be executed in a predefined manner is referred to as an exception* [9].

Once again, we should notice that these exceptions are different in B2B and B2C as technologies for processing these processes differ and raise different exceptions. We should not forget also the business process specifics which also forms distinct situation for exception emerging. Here we raise the assumption, that in order to deal with exceptions in B2C and B2B, we should treat them as separate business cases and think of separate solutions. This is the main purpose of this paper, to identify the differences of B2C and B2B exception handling.

2. Main differences of B2C and B2B

As pointed above, though B2C and B2B serve the same purpose – to make the business more successful, they have slight differences which point to different projections of them on technologies.

B2C applications are accessed through the thin client (usually internet page) and user can directly reach needed information from the business logics application (see Fig. 1). This business logic application can present information to the client/user from different applications, which information transmission is processed in the backend of the system. Due to the information demand, it can be transmitted to the business logic application constantly/online or use time to time synchronisation mode. In such cases all participating applications have their own databases for temporal or constant data storage.

Speaking of B2B applications, information is reached through the application on both nodes (partner and company/consumer) (see Fig. 2). These applications can be used using Internet connection or be installed as separate applications on the partner computer. In such case, information needed for offline using must be stored on local disk (or data storage device) and transmitted for merging from time to time to the main company/consumer server. During the replication of this data, information is checked for validity and uniqueness, so that replication data would be completely new

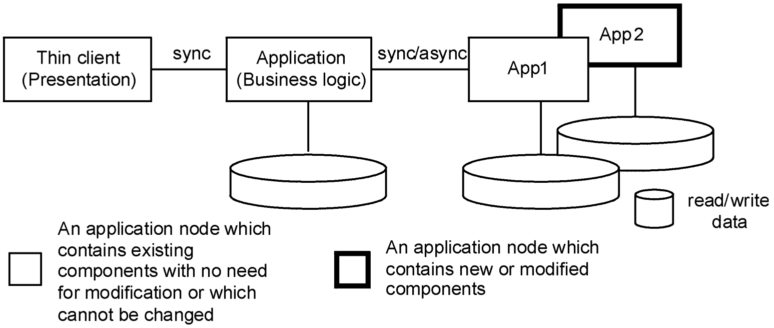
Thin client
(Presentation)
sync
Application
(Business logic)
sync/async
App1
App2
read/write
data
An application node which contains existing components with no need for modification or which cannot be changed
An application node which contains new or modified components

Fig. 1. Business-to-Consumer application conception.

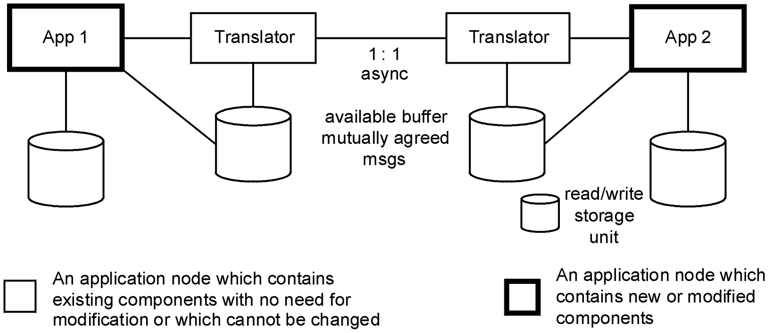
App 1
Translator
1 : 1
async
available buffer
mutually agreed
msgs
Translator
App 2
read/write
storage
unit
An application node which contains existing components with no need for modification or which cannot be changed
An application node which contains new or modified components

Fig. 2. Business-to-Business application conception.

or changed with a purpose to minimise the replication time and amount and to lessen network loads.

One of the main differences between B2C and B2B are purchase cost, frequency and number of customers. These factors are more of human factor nature and points to real world/ad-hoc decision making subtlety. B2B is of wholesale nature so here we have cost and client number as main business factors, which dictate business demands and supply. B2C is simple one person shopping, who concentrates on small items and

Table 1. Differences of B2C and B2B criteria

Involved parties	B2C (business – to – consumer)	B2B (business – to – business)
Buyer/Seller Relation	1 – 1	1 – many
Product Source	Non – Intermediary – Based	Non – Intermediary – Based
Evolution Level	Publishing	Triggering
Functional Integration	Single Function, no integration	Multiple, partially integrated functions

very often. In this case, cost is not always the risk factor, but quality, satisfaction or even delivery time.

Table 1 depicts main factors, which influence the used technologies and system differences. As it can be seen, main differences are in information presentation and gathering points [4].

3. Architecture differences

Before speaking of the key problems, rising in B2C and B2B applications, we should point to the main differences between applications architecture.

B2C application architecture does not differ from simple client-server application architecture; where client is internet user, connecting to the system thought the web using secure connection. As B2C applications can involve third party applications for sensitive data exchange (like internet payments using transactions to bank systems), the customer in this case can use various security features for protecting its personal data (like PKI). The e-business application usually resides on company side and presents needed information and services thought the web application server, which should reside in demilitarised zone, with a purpose to protect company's applications from intruders.

B2B applications transform inter-organizational relationships and are dedicated for partnership. They can be used to exchange information between businesses and for collaboration, sharing information and knowledge between businesses for mutual benefit [2]. In other words, B2B application architecture is more of external communication mechanism nature. As in this case information is exchanged between two similar systems which serve the same purpose and are of inter-organisational relationship type. They exchange information using messages, which carries needed data for process execution. B2B applications also use secure connection to the main business application server and usually does not limit with encrypted usernames and passwords (usually e-signature or e-certificate technologies are used).

Summarising, we can point that the main B2B exception lot rises from inter-organisational relationships and deals with different manufacturer workflow management systems, which exchanges information using only messages. In B2C the main

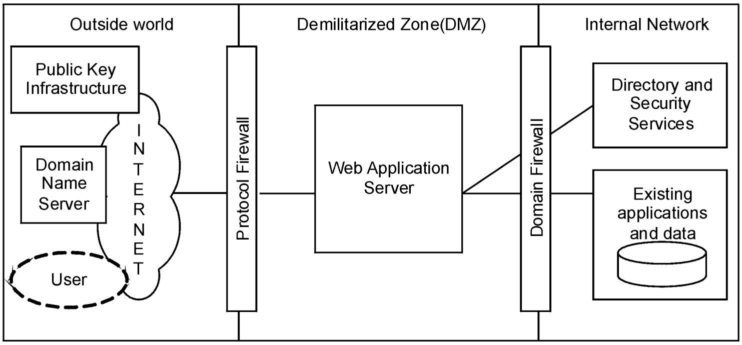

Fig. 3. Business-to-Consumer Application Framework for e-business architecture.

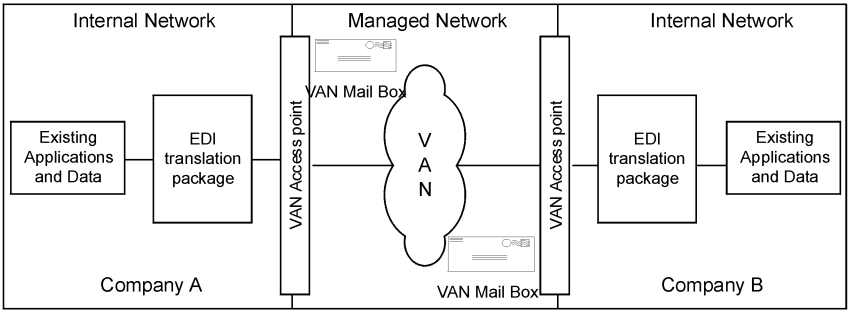

Fig. 4. Business-to-Business Application Framework for e-business architecture.

exception source, rising from architecture level, is network access performance and application data refresh problems.

4. Exception differences

Generally speaking of e-business exceptions nature, we can notice that the reasons of exception emerging are quite the same in B2C and B2B - unpredicted real world situations. Thought the triggers and spots of exception emerging are different and this points to different exception handling methods.

Business activities, which cannot be executed in a predefined manner and cannot reach their desired outcomes, are referred to as exceptions. As e-business applications are generally built using workflow methods, it requires specification of both, normal process flow and possible variations due to exceptional situations that can be anticipated and monitored [10]. Seeking to better understand the dependencies in exceptions and their nature, we will use exception classification which later can be detailed with B2C and B2B exception features. We separate two types: failures and exceptions. Failures are corresponding to exceptions which rarely impact business goal, just technologies.

- Failures can also be divided into [11]:
  - basic failures – corresponding to failures at the system level (e.g., DBMS, operating system, or network failure); and
  - application failures – corresponding to failures of the applications invoked by the Workflow Management System (WFMS) in order to execute a given task. This kind of failures sometimes tends to become exceptions.

These types' failures are closely related, since basic failures can cause application failures. The failure itself is not considered an exception, because it is usually a system failure and can be ascribed to an infrastructure level exception. The failure itself usually is a result of some core exception and must be solved through the core issue, rather than at the business process level.

- Exceptions can be grouped into:
  - expected exceptions, which are [11, 12]:
    - workflow exceptions – related to starting or finishing tasks;

  - data exceptions – related to workflow data change;
  - temporal exceptions – time related exceptions;
  - external exceptions – related to external events.
- unexpected exceptions can be classified as [3, 13]:
  - useful exceptions – a "key to effective and flexible" processes, usually easily handled;
  - unanticipated exceptions – the result of an unexpected, infrequent and non-repetitive event.

In essence, expected exceptions cannot be called true exceptions, simply because they do not impact the final goal of the business process – a method of handling them is already defined using workflow rules, which allows achieving the final goal. Here we have a very slim line between exception and rule because; as soon as exception is identified and included into the workflow graph it becomes a rule. And vice versa, the predefined workflow rule sometimes can turn into exception if all the factors are not taken into account. This part of exceptions, expected exceptions, have more to do with performance issues (the correct and timely information supply), which are more subscribed to B2C exceptions.

Useful exceptions are always identified / solved during the build-time. The dedicated process graph is supplemented with special branches, which would be used during the initiated process instance should the need arise. These are predefined exceptions. This part of exceptions is quite rare.

The unanticipated and unexpected exceptions are usually corresponding to B2B exceptions and are of inter-organisational nature. They are the most dangerous to the business goals as they are difficult to catch, because the main information link between partner and company connecting them and the process is passed through the public internet and can be influenced by the unexpected outside factors (like connection reliability, malice damage, partner or company inside unpredicted factors).

5. Exception handling differences

Speaking of any exception, the most important characteristics is the place of exception, which usually points to causes of its emergence. That means, to handle an exception, one has to know where it occurred, why it occurred and how to solve it. In other words it is exception detection, diagnosis and resolution [14, 15, 16].

Detection phase always starts from the model of correct behaviour, both for entire system and for each individual component. This model must be drawn such, which achieves business process goal and is considered ideal process model.

Diagnosis phase is used to determine which methods or tools should be used to solve unacceptable situation. In this phase exception specifications can be processed, which are generally made in the detection phase. The results later are used to make the exception handler specification. For the systems efficiency exception handler should be generated by the autonomous process or system, which could have an exception and exception handlers' knowledge base. In this way, information and solutions are collected all the time and reused if possible in the future. This would shorten exceptions handling time and cost, which are critical for every business process model. In generally, exception detection and diagnosis needs a lot of precise and timely information, so this information can be also stored using ontology. It can be used to describe the ideal business process model in the terms of the ontology, so that it clears communication between system developers and domain experts. In the ontology-based case management there are used three phases: abstraction, retrieval and adaptation. Each phase uses different specialized ontology to represent exception case, or to retrieve similar cases or to adapt the existing cases. Use of ontology clarify and frame business process terminology and logic, so that the exception handling can be achieved more precise.

Exception handlers are usually modelled as autonomous processes and are always enacted in the place of exception occurrence, not appearance. These processes can implement three different types of actions: masking, propagation and recording [14]. These actions can be used separately or together. The exception process can ignore the exception, modify the process according to changed data and record the situation for the future reuse. Also exception process can retry corrupted task, execute appropriate workflow recovery operation or modifications.

Having such a general algorithm for exception handling, we can identify distinctions between B2C and B2B processes. The main structural difference between these process systems is that B2C uses direct Internet connection to the e-business application and B2B connection is system to system. In both cases we need workflow engine and business logic applications, and in both cases it is better if exception handling application is separated from them.

Speaking of B2C exception handling we can notice, that main business logic, workflow engine and exception handling applications can use the same storage devices and have only slim lines between these components as B2C application include all necessary components needed for process realisation. Still, B2C exception handling appli-

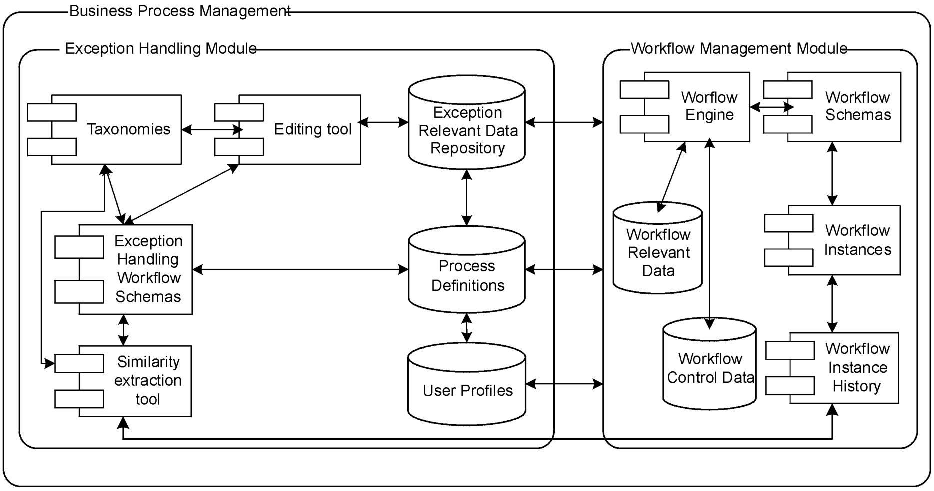
Business Process Management
Exception Handling Module
Taxonomies
Editing tool
Exception Relevant Data Repository
Exception Handling Workflow Schemas
Process Definitions
Similarity extraction tool
User Profiles
Workflow Management Module
Worflow Engine
Workflow Schemas
Workflow Relevant Data
Workflow Instances
Workflow Control Data
Workflow Instance History

Fig. 5. Business-to-Consumer exception handling technological solution.

cation should be separated from main business application and only have slim links with this application. In B2C usually there is no other business application integration and it uses thin client for connection. That means exception handling information databases can be located together with exception handling system components as they do not require separate security insurance or secure connections to them from external networks or systems. So in this case we have only integration between workflow management system (which realise B2C business process) and exception handling application (which generates exception specification, checks for symptoms in predefined taxonomies and looks for proper existing handlers).

The only necessary and useful information, which workflow management system can present is workflow instance history data, because unexpected exceptions occur only in workflow instances. That means during process run-time. Build-time exceptions are more like process cases and are managed as simple design matters. So here we should have a constantly running history data check procedure which must evaluate the correctness of it, using ideal process model and its input/output data specifications.

B2B exception handling is a bit more complicate as here we have system to system architecture which sometimes must have complex and unique integrations between them. In B2B processes, the inter-organisational relationships points to many factors which must be taken into consideration, because two or more different consumer workflow management systems can participate in one business process and exchange information only using messages, which are the main insurer of the correct behaviour of each of these systems.

Having several different workflow engines we even more need to ensure that exception handling application will be able to deal with the exceptions, which rise in each of these engines. That means, we need to ensure that exception handling application will not be dependent on any workflow engine specifics or use particular workflow management system syntax or data exchange methods. But still we have to ensure,

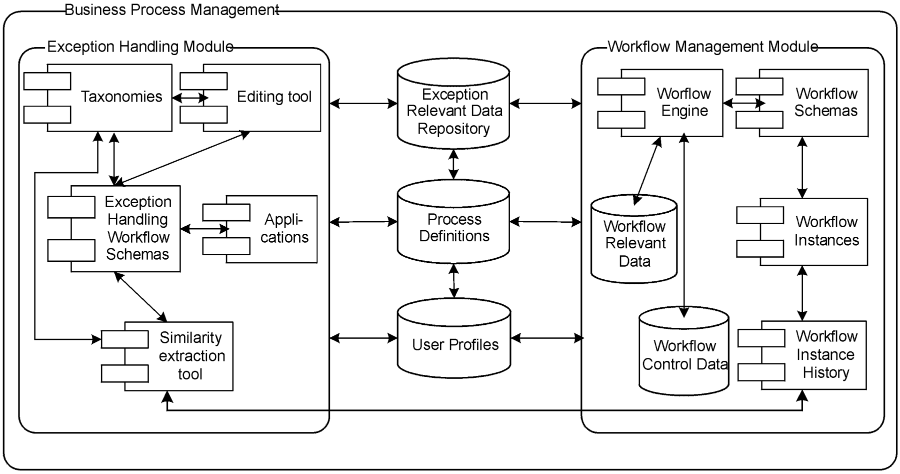
Business Process Management
Exception Handling Module
Taxonomies
Editing tool
Exception Handling Workflow Schemas
Appli-cations
Similarity extraction tool
Exception Relevant Data Repository
Process Definitions
User Profiles
Workflow Management Module
Worflow Engine
Workflow Schemas
Workflow Relevant Data
Workflow Instances
Workflow Control Data
Workflow Instance History

Fig. 6. Business-to-Business exception handling technological solution.

that each separate workflow engine will be able to reach already existing exception definition and handling specifications.

So in B2B exception handling case, we must separate workflow management system, exception handling application and exception relevant data storage so that they could be reached by any of them even from outside network or system connections and share/reuse already gathered exception handling knowledge.

In so doing we can ensure that inter-organisational relationships are not burdened with additional procedures of exception detection or exception data merging/transformation for other workflow engines. The other important factor is, that we ensure main business application security and availability of necessary exception handling data on time. The B2B exception handling application itself can be located on company side or on each partner side, if they use distinct workflow engines for connection with main business application.

6. Conclusions

Generally speaking, B2B and B2C application performance insurance is the same important from the user point of view. So application constructor main task is only to identify the lot of differences between these business processes and technological instruments, used to realize these processes into internet existence.

These research points that the main differences reside in user and business application connection link as here we have inter-organizational link (in B2B) and direct internet link (in B2C), which points to particular technologies usage. Speaking of raised exceptions differences, the types of them can be influenced not only by these connections types, but also by internal technology architecture and construction. As exceptions always influence the business decision and goals, the main system constructor task is also to ensure needed and newest information supply for user. And this task points to some differences in B2C and B2B exception handling.

To this moment, there are no existing systems which would satisfy pointed exception handling criteria and fit for any workflow management system without special adapting.

References

1. K. Rutledge, *The Business Case for E-Business*, IBM Press (2000).
2. L. Ennser, P. Leo, T. Meszaros, E. Valade, *IBM Redbooks*, *The XML Files*: *Using XML for Business-to-Business and Business-to-Consumer Applications*, ITSO, IBM Corp. (2000).
3. W.S. Sadig, M.E. Orlowska, *On Capturing Exceptions in Workflow Process Models*, The university of Queensland, Australia, Department of Computer Science and Electrical Engineering.
4. H. Kuhn, S. Junginger, F. Bayer, How business models influence the development of e-business applications, in: *Proceedings of eBusiness and eWork*, Madrid, Spain (2000).
5. *Workflow Management Coalition, Terminology & Glossary*. WFMC-TC-1011. Issue – 3.0. Winchester Hampshire, United Kingdom (1999).
6. K. Дэвид, *Электронная коммерция*, Мицрософт Пресс (1999).
7. J. Adams, S. Koushik, *Patterns for E-Business*, IBM Press (2002).
8. I. Naick, M. Berkhoff, D.V. Bosnich, *Business-to-Business Integration Using MQSeries and MQSI*. Petterns for e-business series. IBM Redbooks, SG24-6010-00 (2000).

9. D. Vojevodina, G. Kulvietis, Change and exceptions in business process handling differences, *WSEAS Transactions on Systems*, **10**(4), 1658–1668 (2005).
10. F. Casati, M.G. Fugini, I. Mirbel, An environment for designing exceptions in workflow, *Information Systems*, **24**(3), 255–273 (1999).
11. F. Casati, *Models, Semantics, and Formal Methods for the Design of Workflows and their Exceptions*, *PhD Thesis*, Department of Electronics and Informatics, Politecnico de Mi-lano (1998).
12. D. Worah, A. Sheth, K. Kochut, J. Miller, *An Error Handling Framework for the ORBWork Workflow Enactment Service of METEOR*, Large Scale Distributed Information Systems Lab, The University Of Georgia, Atsens.
13. W.S. Sadig, M.E. Orlowska, *On Capturing Exceptions in Workflow Process Models*, The University of Queensland, Australia, Department of Computer Science and Electrical En-gineering.
14. L. Zongwei, A. Sheth, K. Kochut and J. Miller, Exception handling in workflow systems, *Applied Intelligence*, **13**(2), 125–147 (2000).
15. C. Dellarocas, Toward exception handling infrastructures in component-based software, in: *Proceedings of the International Workshop on Component-based Software Engineering*, *20th International Conference of Software Engineering* (*ICSE*), Kyoto, Japan, April 25–26 (1998).
16. C. Dellarocas, M. Klein, A knowlwdge-based approach for handling exceptions in business process, *Information Technology and Management*, **1**(3), 155–169 (2000).

REZIUMĖ

D. Vojevodina, G. Kulvietis. V-V ir V-K darbų sekų nukrypimų tvarkymo skirtumai

Konkurencijai didėjant dauguma verslo kompanijų siekia kuo geriau užtikrinti santykius su savo klientais. Tam dažnai pasitelkiamos internetinės technologijos. Verslo santykių su klientais tipas (Verslas-verslui ar verslas-klientui) priklauso nuo kompanijos verslo strategijos. Šiomis technologijomis kompanijos stengiasi užtikrinti greitą ir kokybišką ryšį/kontaktus su klientais. Visgi, tokių sprendimų kokybė dažnai priklauso nuo daugelio faktorių, bet ko gero svarbiausias iš jų yra sistemų efektyvumas. Kaip terpė šioms sistemos realizuoti dažniausiai pasirenkama Internetas, kuris deja ne visada gali užtikrinti kokybišką ir patikimą ryšį viso bendravimo su klientu metu. Šio straipsnio tikslas yra identifikuoti V-V ir V-K sistemų skirtumus ir kartu juose aptinkamų nukrypimų sprendimo vartotojui priimtinu būdu skirtumus.