

ПРОЕКТИРАНЕ НА БАЗА ДАННИ ЗА СИСТЕМА ЗА УПРАВЛЕНИЕ НА КАТОДНА ЗАЩИТА

Мила Илиева¹, Теодора Христова¹

¹Минно-геоложки университет "Св. Иван Рилски", 1700 София, e-mail teodora@mgu.bg

РЕЗЮМЕ. Въз основа на вече определени обекти и функции на обектите от системата за катодна защита са определени атрибутите, необходими за изграждането на база данни за управление на процесите по защита от корозия.

DESIGN OF DATABASE MANAGEMENT SYSTEM CATHODIC PROTECTION

Mila Ilieva¹, Teodora Hristova¹

Mining and Geology University "St. Iv. Rilski", 1700, Sofia, e-mail teodora@mgu.bg

ABSTRACT. On the basis of certain objects and features of the objects of cathodic protection system defined attributes necessary to build a database for managing processes on corrosion protection.

Въведение

В съвременните условия на развитие на информационните услуги е препоръчително изграждането на единни системи за управление и контрол на обектите. Това включва не само техническото обслужване на системата, но и определянето на икономическите параметри, възприети като политика на фирмата. Съвременните производители предлагат пълно техническо оборудване за защита на газопроводите от корозия и следене на нужните параметри. Тенденциите са насочени към развиване на дистанционни информационни мрежи и бази данни за следене на характеристиките на защитата, като има само един диспечерски пункт, от който тя се управлява. За постигане на ефективност на процесите е необходимо да се изгради обща система за наблюдение и контрол на параметрите на цялата тръбопроводна мрежа, както и да се отчете ефективността от приложените превантивни мерки. Една такава система трябва да включва в себе си не само технически, но и икономически параметри като брой работници, цена на поддръжката, планови ремонти и други. Данните от тази система се записват в база данни. Въз основа на история на процесите, запазени в базата, чрез управляващ алгоритъм или мениджърско решение се изработват управляващи въздействия.

- логическа и физическа независимост
- минимално излишество на данни в БД
- удобен и мощен интерфейс
- ефективна обработка на заявките
- осигуряване на цялостността на данните т.е. логическа непротиворечивост и секретност на данните на отделните потребители.

В една база данни основното понятие е обект. Като обект може да се определи всяко множество с общ набор от характерни свойства, които го представят достатъчно точно. Прието е множествата от еднотипни обекти да се наричат *класове от обекти*, а свойствата които ги характеризират – *атрибути*. Всеки атрибут е величина, която е определена от име и множество от допустими стойности или, както по-често се нарича, *област на атрибута*. Допустимо е различни атрибути да имат една и съща област. За всеки клас от обекти може да бъде определен един или група от няколко атрибута, които еднозначно идентифицират всеки обект от класа. Този атрибут или група от атрибути се нарича *ключ за класа от обекти*. Всеки клас от обекти има поне един ключ. Това следва от факта, че в класовете от обекти се съдържат само различни помежду си обекти.

Методология

1. Разработване на проект

За постигане на независимост на данните при обработка, търсене и обновяване на данните към СУБД се налагат следните изисквания:

- разделяне на описанието на данните от тяхната обработка

За създаването на база данни, обслужваща системата „катодна защита“, първо се определят класовете управлявани обекти :(УО) за участниците, обекта на управление и начина на управление. Участниците са: Мениджър Катодна защита – УО ManagerCP и Изпълнител Катодна защита – УО EmployeeCP. Обект на управлението са: катодна защита – УО Cathode Protection, мениджър Катодна защита и изпълнител Катодна защита. Параметрите на обектите за управление се задават чрез профил – УО Profile. Елементите на управлението се

изразяват чрез функционалните области: YO Configuration, YO Maintenance, YO Security, YO Accounting и YO Performance (Илиева-Обретенова, М., Христова, Т).

Проектирането на базата данни (Ернандес, Майкъл Х,2007) е свързано със създаването на проект т.е. на определен модел на структуриране на данните. Етап от този процес е изборът на класове от обекти, съответните им набори от атрибути, както и отношенията и връзките, които съществуват между обектите на различните класове. Утвърдили са се три модела данни в съвременната теория и практика на бази данни:

1. Релационен модел, основаващ се на понятието отношение или релация.
2. Мрежов модел, основаващ се на ориентираните графи.
3. Йерархичен модел, ползващ "гори" от дървовидни структури.

За постигане на ефективност на работа на системата за управление на катодната защита и проектираната към нея база данни е избран релационен модел.

2. Избор на език за бази данни

Изборът на език за описание на данните зависи от различни величини – атрибути, логически записи и файлове, релации, честота на обновяване на данните.

SQL включва в себе си голям брой оператори за достъп и дефиниция на данните в базата. Има възможност чрез операторите си и променливите на обкръжаващия го главен език да предава данни в собствените си отношения (релации). Най-новите бази данни с графичен интерфейс са MS ACCESS, Oracle и други. Недостатък на Access е, че при по-големи бази данни се наблюдава значително забавяне на програмата.

Програмата FoxPro позволява да се извлича информация за определени записи от различните таблици в базите данни, чрез заявки. Тази форма позволява и сортиране и извличане на исканите записи според подбран от нас принцип, аритметични действия с цифровите записи и др. Базата дава възможност за оформянето на отчети.

При необходимост от голям брой записи в таблиците и релации тип 1:n и n:m обикновено се използват комерсиални бази данни като ORACLE, SYBASE или свободни разработки - POSTGRESQL, MYSQL.

За разработване на база данни за обслужване на системата на катодна защита се избира език java, който е обектно-ориентиран. Негова основна концепция са класовете, които се дефинират чрез свойствата (атрибутите) и поведението (методите) на обектите. Това прави този език подходящ за обслужването на информационните потноци при управлението на катодната защита. Няма глобални променливи и функции – всичко се дефинира в класове. За разлика от C++, където има множествено наследяване, в java то е заместено от интерфейси. Синтаксисът на java е сходен с този на C++ и преминаването от този език към java е лесно. На java се разработва изключително разнообразен софтуер: офис приложения, уеб приложения, настолни приложения, приложения за мобилни телефони, игри и много други.

Кодът, написан за java не се компилира до машинен код за определен процесор, а до специфичен за езика код, наречен байт код. Поради това за изпълнението на програма, написана на java, е необходима т. нар. Виртуална машина. Подобна реализация има своите предимства и недостатъци. Сред главните предимства са:

- Лесна преносимост между различните платформи (софтуерни или хардуерни) – веднъж написана и компилирана, една java програма може да бъде стартирана на компютри независимо от архитектурата или от операционната им система. За целта е необходимо само да се инсталира виртуална машина за съответната платформа;

- Допълнителни действия, извършвани от виртуалната машина са: освобождаване на паметта от обекти (класове), които не се използват, проверка за размерността на масивите и др.

- Възможности за контрол на правата на потребителя на ниво виртуална машина;

- Първоначално заделяне на heap, част от паметта, резервирана за java;

- Висока степен на сигурност поради факта, че програмистите не работят директно с паметта.

Главен недостатък е необходимостта от допълнителни ресурси (под формата на процесорно време и памет) за изпълнението на самата виртуална машина.

При създаването на информационна система е необходимо да се определят приоритетите на достъп и да се осигури защита. Контролът на данните е вграден в описанието на защитата на базата от нерегламентиран достъп. Администраторът на създадената база данни трябва да има следните функции:

- следене за правилното поддържане на данните и недопускането на конфликтни точки между различните групи потребители;

- определяне на паролната дума, позволяваща пълния достъп до всички релации в базата данни, в документацията на системата, като изрично обяви, че използването ѝ крие опасност за нарушаване целостта на данните;

- смяна на паролните думи при необходимост или при желание на възложителя, без да злоупотребява с това;

- защита на данните, която да осигури максимална ефективност при използването им от различните групи потребители;

- следене на работата на крайните потребители, за да установи реалните им изисквания към базата данни;

- архивиране и възстановяване на базата от данни.

Резултати

В този параграф са описани класове управлявани обекти с атрибути и операции – права за достъп на действащите лица. Представени са функционални области: Поддържане на катодна защита (Edwards, Anthony V., Robert J. Whitaker, 2008), Технически характеристики на катодна защита (Baggiolini, Vitto,2008) и Достъп на изпълнител до системата (Ford, David,2003).

Таблица 1.

Класове управлявани обекти за Поддържане на катодна защита

Обект	CPMaintenance – Поддържане на катодна защита
Атрибути	CPMaintenanceId – Идентификатор на поддържане на катодна защита EMaintenanceIdList – Списък с идентификатори на Поддържане от служител
Бележки	CPMaintenanceId се чете от мениджъра. EMaintenanceIdList съдържа списък с идентификаторите на задълженията на различните служители за безотказна работа на системата. Списъкът се попълва от мениджъра, а всеки служител чете своите задължения.

Обект	PreventiveFunction – Превантивна функция, която се стартира през определени интервали от време
Атрибути	PreventiveFunctionId – Идентификатор на превантивна функция FunctionDescription – Описание на превантивната функция, кои елементи наблюдава и какво наблюдава AggregationFrequency – Интервал от време (честота), през който се стартира функцията AvailabilityTime – Време за изпълнение на функцията WorkLoad – Работно натоварване по време на изпълнение на функцията PossibleMonitoring – Възможно наблюдение по време на изпълнение на функцията
Бележки	PreventiveFunctionId и FunctionDescription, AvailabilityTime, PossibleMonitoring се четат и попълват от мениджъра. AggregationFrequency се избира от мениджъра. WorkLoad се записва автоматично в момента на стартиране на функцията.

Обект	Result – Резултат след стартирането на превантивната функция
Атрибути	ResultId – Идентификатор на резултат ResultDescription – Описание на резултата
Бележки	ResultId се чете от мениджъра и служителя. ResultDescription се попълва автоматично от системата. Съобщението може да бъде "Всички елементи са в изправност" или "Открит е проблем в участък Стартирай тестваща функция!" и се чете от мениджъра.

Обект	TestingFunction – Тестваща функция
Атрибути	TestingFunctionId – Идентификатор на тестваща функция TestingInstance – Реализация на тестването: съдържа обекта на тестващата функция, кой елемент се тества
Бележки	TestingFunctionId – се чете от мениджъра и служителя. TestingInstance може да се попълва автоматично от системата, когато се тестват няколко елемента последователно, или се попълва от мениджъра, когато се тества само един елемент

Обект	Localization – Локализация на повреда
Атрибути	LocalizationId – Идентификатор на локализация LocalizationDescription – Описание на елемента, в който е открита повреда: номер, име и местоположение FaultInfold – Идентификатор на повреда
Бележки	LocalizationId и FaultInfold се четат от мениджъра и изпълнителя. LocalizationDescription се попълва автоматично от системата.
Обект	AlarmLog – Журнал Аларми, В него се записват аларми за повреди, които не са открити от превантивната функция
Атрибути	AlarmLogId – Идентификатор на журнал Аларми NumberOfRecords – Брой записи
Бележки	AlarmLogId и NumberOfRecords се четат от мениджъра и изпълнителя.

Обект	AlarmRecord – Запис Аларма, аларма за повреда, която не е открита от превантивната функция
Атрибути	AlarmRecordId – Идентификатор на запис Аларма PerceivedSeverity – Приоритет ListOfSuspectObjects – Списък на заподозрените обекти MessageText – Текст на съобщението RepairAction – Действие по възстановяването MonitoredAttributes – Наблюдавани атрибути/величини BackupStatus – Статус на напомняне FaultInfold – Идентификатор на повреда
Бележки	Атрибутите се четат от мениджъра

Обект	HandLog – Журнал Ръчно открити повреди
Атрибути	HandLogId – Идентификатор на журнал Ръчно открити повреди NumberOfRecords – Брой записи FaultId – Идентификатор на повреда, открита ръчно
Бележки	Атрибутите се четат от мениджъра

Обект	EMaintenance – Поддържане от изпълнител, съдържа задълженията на изпълнителя
Атрибути	EMaintenanceId – Идентификатор на поддържане от изпълнител
Бележки	Атрибутът се попълва от мениджъра, а се чете от изпълнителя

Обект	FaultInfo – Информация за повреда за специфичен изпълнител
Атрибути	FaultInfold – Идентификатор на информация за повреда ErrorCount – Брояч на повреди Date – Дата OperationUnits – Видове използвани елементи за отстраняване на повредата VolumeOfUsedUnits – брой използвани елементи
Бележки	FaultInfold, OperationUnits, VolumeOfUsedUnits се попълват от мениджъра, а се четат от изпълнителя. ErrorCount и Date се попълват автоматично от системата и се четат от изпълнителя.

Обект	Fault – Повреда, открита ръчно
Атрибути	FaultId – Идентификатор на повреда, открита ръчно FaultList – Списък на повреди, които могат да се откриват ръчно
Бележки	FaultId се попълва от мениджъра, а се попълва от изпълнителя. FaultList се попълва от мениджъра, а се чете от изпълнителя. Могат да се добавят нови повреди или да се изтриват такива, които вече се откриват автоматизирано.

Таблица2.

Класове управлявани обекти за Технически характеристики на катодна защита

Обект	CPPerformance – Технически характеристики на катодна защита
Атрибути	CPPerformanceId – Идентификатор на технически характеристики на катодна защита EPerformanceId – Идентификатор на технически характеристики за изпълнител
Бележки	Атрибутите се попълват и се четат от мениджъра.

Обект	CPOperationLog – Журнал Технически характеристики на катодна защита
Атрибути	CPOperationLogId – Идентификатор на журнал Технически характеристики на катодна защита NumberOfRecords – Брой записи
Бележки	Атрибутите се четат и попълват от мениджъра.

Обект	ElementOperationLog – Журнал Наблюдение на елемент
Атрибути	ElementOperationLogId – Идентификатор на журнал Наблюдение на елемент NumberOfRecords – Брой записи ElementId – Идентификатор на елемент EOperationId – Идентификатор на намесата на изпълнител за коригиране на параметрите на елемент при установяване на отклонение.
Бележки	ElementOperationLogId, NumberOfRecords и ElementId се четат и попълват от мениджъра. EOperationId се чете и попълва от мениджъра и се чете от изпълнителя.

Обект	CPStatusReport – Рапорт за статус на катодна защита
Атрибути	CPStatusReportId – Идентификатор на рапорт за статус на катодна защита QoSInfoId – Идентификатор на параметрите, които трябва да следи един изпълнител MediaOfReport – Медия на рапорта: хартия или електронен вид SecurityRights – Права за достъп LastDate – Дата ListOfParameters – Списък на параметрите, които следят всички изпълнители NumberOfRecords – Брой записи Employee – Име на служител, който следи даден параметър Functions – Допълнителни функции за мениджъра
Бележки	Атрибутите се четат и попълват от мениджъра. MediaOfReport може да се избира от мениджъра. ListOfParameters – Мениджърът може да избира кои параметри да се включат в рапорта.

Обект	EPerformance – Параметри на техническите характеристики, които трябва да следи един изпълнител
Атрибути	EPerformanceId – Идентификатор на параметрите на техническите характеристики, които трябва да следи един изпълнител
Бележки	Атрибутът се попълва от мениджъра, а се чете от изпълнителя.

Обект	QoSInfo – Информация за параметрите, които трябва да следи един изпълнител
Атрибути	EPot1 – Потенциална разлика „съоръжение-земя“ EPot2 – Потенциал електрод-земята ECur – Защитен ток EPot3 _{1-n} – Защитен потенциал по дължина на съоръжението за конкретната точка на измерване
Бележки	Атрибутите се четат от изпълнителя, стойностите им се записват в системата. Граничните стойности се определят от мениджъра.

Обект	ESetting – Намеса на изпълнител за коригиране на параметрите на системата за катодна защита
Атрибути	ESen1 – Пробив в стената на тръбопровода ESen2 – Прехвърляне на електрода ELev – Ниво на корозия на тръбопровода ESen2 – Състоянието на изолацията
Бележки	Атрибутите се четат от изпълнителя, стойностите им се записват в системата. Граничните стойности се определят от мениджъра. При достигане на стойностите на атрибутите близки до граничните или при сигнал ESen2 и ESen1 се изпълнява алгоритъм от процедура AlarmRecord.

Обект	EOperation – Намеса на изпълнител за коригиране на параметрите на елемент
Атрибути	EOperationId – Идентификатор на намеса на изпълнител за коригиране на параметрите на елемент QoSOfOperation – Вид намеса, начин на извършване
Бележки	Атрибутите се попълват от мениджъра, а се четат от изпълнителя.

Таблица3.

Класове управлявани обекти за Достъп на изпълнител до системата за управление на катодна защита

Обект	EmployeeProfile – Профил на служител, съдържа цялата информация за наблюдение и контрол, свързана с един потребител
Атрибути	EmployeeProfileId – Идентификатор на профил на служител SecurityAccess – Достъп до защита
Бележки	Мениджърът попълва атрибутите, изпълнителят чете и пише в тях.

Обект	EConfiguration – Информация за конфигурация на системата
Атрибути	EConfigurationId – Идентификатор на информация за конфигурация на системата
Бележки	Мениджърът попълва атрибута, изпълнителят го чете.

Обект	RoutInfo – Географска карта с местоположението на датчиците
Атрибути	RoutInfoId – Идентификатор на географската карта с местоположението на датчиците
Бележки	Мениджърът попълва атрибута, изпълнителят го чете.

Обект	NPlan – План, съдържа данни за датчиците
Атрибути	NPlanId – Идентификатор на план с данни за датчиците Number – Номер Name – Име Location - Местоположение
Бележки	Мениджърът попълва атрибутите, изпълнителят ги чете.

Обект	EMaintenance – Поддържане от изпълнител, съдържа задълженията на изпълнителя
Атрибути	EMaintenanceld – Идентификатор на поддържане от изпълнител
Бележки	Атрибутът се попълва от мениджъра, а се чете от изпълнителя

Обект	FaultInfo – Информация за повреда за специфичен изпълнител
Атрибути	FaultInfold – Идентификатор на информация за повреда ErrorCount – Брояч на повреди Date - Дата OperationUnits – Видове използвани елементи за отстраняване на повредата VolumeOfUsedUnits – брой използвани елементи
Бележки	FaultInfold, OperationUnits, VolumeOfUsedUnits се попълват от мениджъра, а се четат от изпълнителя. ErrorCount и Date се попълват автоматично от системата и се четат от изпълнителя.

Обект	Fault – Повреда, открита ръчно
Атрибути	FaultId – Идентификатор на повреда, открита ръчно FaultList – Списък на повреди, които могат да се откриват ръчно
Бележки	FaultId се попълва от мениджъра, а се попълва от изпълнителя. FaultList се попълва от мениджъра, а се чете от изпълнителя. Могат да се добавят нови повреди или да се изтриват такива, които вече се откриват автоматизирано.

Обект	EPerformance – Параметри на техническите характеристики, които трябва да следи един изпълнител
Атрибути	EPerformanceld – Идентификатор на параметрите на техническите характеристики, които трябва да следи един изпълнител
Бележки	Атрибутът се попълва от мениджъра, а се чете от изпълнителя.

Обект	QoSInfo – Информация за параметрите, които трябва да следи един изпълнител
Атрибути	EPot1 – Потенциална разлика „съоръжение-земя“ EPot2 – Потенциал електрод-земя ESur – Защитен ток EPot3 _{1-n} – Защитен потенциал по дължина на съоръжението за конкретната точка на измерване
Бележки	Атрибутите се четат от изпълнителя, стойностите им се записват в системата. Граничните стойности се определят от мениджъра.

Обект	ESetting – Намеса на изпълнител за коригиране на параметрите на системата за катодна защита
Атрибути	ESen1 – Пробив в стената на тръбопровода ESen2 – Прехвърляне на електрода ELev – Ниво на корозия на тръбопровода ESen2 – Състоянието на изолацията
Бележки	Атрибутите се четат от изпълнителя, стойностите им се записват в системата. Граничните стойности се определят от мениджъра. При достигане на стойностите на атрибутите близки до граничните или при сигнал ESen2 и ESen1 се изпълнява алгоритъм от процедура AlarmRecord.

Обект	EOperation – Намеса на изпълнител за коригиране на параметрите на елемент
Атрибути	EOperationId – Идентификатор на намеса на изпълнител за коригиране на параметрите на елемент QoSOfOperation – Вид намеса, начин на извършване
Бележки	Атрибутите се попълват от мениджъра, а се четат от изпълнителя.

Обект	ESecurity – Защита на системата, задължения на изпълнителя за защита на системата
Атрибути	ESecurityId – Идентификатор на защита на системата
Бележки	Атрибутът се попълва от мениджъра, а се чете от изпълнителя.

Обект	EAuthoriz – Оторизация на изпълнител, задължения
Атрибути	EAuthorizId – Идентификатор на оторизация на изпълнител EGroup – Група изпълнители AuthorizationType – Вид оторизация (задължения) Access – Достъп (парола) ObjectAttributeList – Списък с атрибути на обекта
Бележки	Атрибутите се попълват от мениджъра, а се четат от изпълнителя.

Обект	EAccounting – Осчетоводяване на изпълнител
Атрибути	EAccountingId – Идентификатор на осчетоводяване на изпълнител
Бележки	Атрибутът се попълва от мениджъра, а се чете от изпълнителя.

Обект	Remind – Напомняне за извършване на плащане
Атрибути	RemindId – Идентификатор на напомняне DateDue – Падеж Credit – Кредит Options – Опции за наказание SuggestedPayment – Предложение за плащане (краен срок) MediaOfReminder – Медия на напомнянето: хартия или електронен вид Notarisation - Заверка
Бележки	Атрибутите се попълват от мениджъра, а се четат от изпълнителя.

Обект	FinInfo – Съдържа данни за създаване на различни справки
Атрибути	FinInfoId – Идентификатор на справка MediaOfReport – Медия на справка: хартия или електронен вид DateDue – Дата ReportAttributeList – Списък с атрибути на рапорта
Бележки	Справките най-често се правят за изпълнител, за да се види неговото финансово състояние. За мениджъра е полезно да види финансовото състояние и натоварването на всеки от изпълнителите, участващи в обслужването. Мениджърът и изпълнителят могат да четат и попълват атрибутите. MediaOfReport, DateDue, ReportAttributeList могат да се избират от заявителя.

Обект	Invoice - Фактура
Атрибути	InvoiceId – Идентификатор на фактура MediaOfInvoice – Медия на фактура: хартия или електронен вид TypeOfBill – Вид фактура (данъчна или обикновена) Notarisation - Заверка
Бележки	Атрибутите се попълват от мениджъра, а се четат от изпълнителя.

Обект	Payment - Плащане
Атрибути	PaymentId – Идентификатор на плащане PaymentOptions – Опции на плащане PaymentVerification – Заверка PaymentFrequency – Честота на плащане
Бележки	Атрибутите се попълват от мениджъра, а се четат от изпълнителя.

Обект	Salary - Заплата
Атрибути	SalaryId – Идентификатор на заплата Currency – Валута, брой PaymentOptions – Опции на плащане CreditLimit – Лимит на кредит Taxes - Данъци
Бележки	Атрибутите се попълват от мениджъра, а се четат от изпълнителя. PaymentOptions, CreditLimit могат да се избират от мениджъра.

Обект	Limit – Лимит по финансови средства
Атрибути	LimitId – Идентификатор на лимит TotalLimit – Общ лимит (бюджет) PeriodLimit – Лимит за даден период ProductLimit – Лимит за даден продукт
Бележки	Атрибутите се попълват от мениджъра, а се четат от изпълнителя.

Обект	PayMode – Начин на плащане (в случай на съфинансиране и гъвкаво финансиране)
Атрибути	PayModeId – Идентификатор на начин на плащане CoFin – Съфинансиране FlexibleFin – Гъвкаво финансиране
Бележки	Атрибутите се попълват от мениджъра, а се четат от изпълнителя.

Изводи

- На базата на синтезираните изисквания за управление на катодна защита са описани класове управлявани обекти, които поддържат информационните модели, съответстващи на функционалните области. В статията са представени обектите за области: Поддържане на катодна защита, Технически характеристики на катодна защита и Достъп на изпълнител до системата за управление.
- Класовете обекти представят управляваните ресурси на катодната защита и действащите лица: Мениджър катодна защита и Изпълнител
- Класовете управлявани обекти са описани чрез атрибути и операции. Атрибутите характеризират обектите и отчитат корелацията между тях. Операциите представят правата за достъп на потребителите.
- На следващия етап се предвижда реализация на информационната система за управление на катодна защита.

Литература

- Ернандес, Майкъл Х., "Проектиране на бази от данни", СофтПрес ООД, 2007
- Илиева-Обретенова, М., Христова, Т., "Проектиране на информационен модел за управление на катодна защита", под печат
- Baggiolini, Vitto, Jurgen Harms, "Towards Automatic Run-time Fault Management for Component-Based Applications", <http://www.springerlink.com/content/k238r03325284t25/>, 2008
- Edwards, Anthony V., Robert J. Whitaker, "Fault Management: A Functional View of Root Cause Analysis and Correlation", http://www.tavve.com/EV_White_paper.pdf, 2008, pp. 90-101
- Ford, David, Lars Erik-Gadde, Hakan Hakansson, Ivan Snehota, "Managing Business Relationships", 2nd ed., Wiley, Chichester, 2003, pp.15-30

Препоръчана за публикуване от катедра
„Електротехника“, МЕМФ