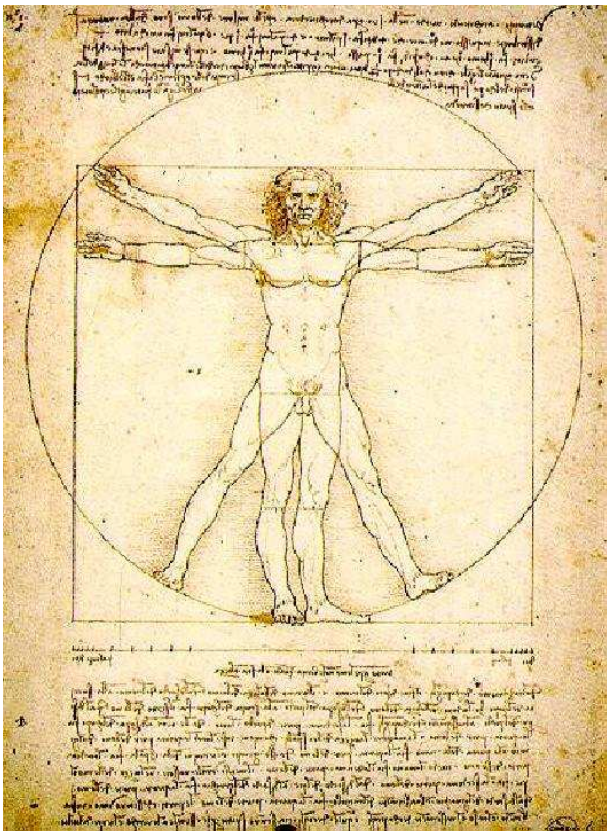
Overview of the DAVINCI Series

H.A. Proper and Th.P. van der Weide



The DAVINCI Series – The Art & Craft of System Alignment

Overview of the DAVINCI Series

H.A. Proper and Th.P. van der Weide

Contents

1 Introduction **7**

2 History **9**

3 Structure of the series **11**

3.1 Domain modeling 12

3.2 Basic design 13

3.3 Advanced design 15

3.4 Definition of Information Systems 20

3.5 Deployment & Maintenance 21

3.6 Architecting 22

4 Implementation at the ICIS **25**

Bibliography **27**

Author Index **31**

Dictionary **33**

Subject Index **35**

Chapter 1

Introduction

The subtitle of the DAVINCI series of lecture notes is *The Art & Craft of Information Systems Engineering*. On the one hand, this series of lecture notes takes a fundamental view (the *craft*) on the field information systems engineering. At the same time, it does so with an open eye to practical experiences (the *art*) gained from information system engineering in industry.

The kinds of information systems we are interested in range from personal information appliances to enterprise-wide information processing (a system of information systems). Even more, we regard an information system as a system that "handles" information, where "handling" should be interpreted in a broad fashion. The actors that do this "handling" can be computers, but can equally well be other "symbol wielding machines" [Hop03], or even human beings. The mix of humans and computers/machines in information systems makes the field of information system engineering particularly challenging.

The concept of "information" itself is very much related to the concepts of *data*, *knowledge* and *communication*. Based on [FVSV+98], we will (throughout the DAVINCI series) use the following definitions:

Data – Any representation in some language. Data is therefore simply a collection of symbols that may, or may not, have some meaning to some actor.

Information – The knowledge increment brought about when a human actor receives a message. In other words, it is the difference between the conceptions held by a human actor *after* interpreting a received message and the conceptions held beforehand.

Knowledge – A relatively stable, and usually mostly consistent, set of conceptions posessed by a single (possibly composed) actor.

In more popular terms: "an actor's picture of the world".

Communication – An exchange of messages, i.e. a sequence of mutual and alternating message transfers between at least two human actors, called communication partners, whereby these messages represent some knowledge and are expressed in languages understood by all communication partners, and whereby some amount of knowledge about the domain of communication and about the action context and the goal of the communication is made present in all communication partners.

When referring to an information system, we therefore really refer to systems that enable the communication/sharing of knowledge by means of the representation (by human actors), storage, processing, retrieval, and presentation (to human actors) of the underlying representations (data). This also implies that we will treat *information retrieval systems*, *knowledge-based systems*, *groupware systems*, *decision support systems*, etcetera, as special classes of information systems.

The lecture notes in the DAVINCI series have been organized around six key aspects of an information system's life-cycle:

Definition – The description of the requirements that should be met by both the desired information system as well as the documents documenting this information system. In literature this is also referred to as requirements engineering.

With regards to the information system, the resulting descriptions should identify: *what* it should do, *how well* it should do this, and *why* it should do so. With regards to the documentation of the information system, the descriptions should identify *what* should be documented, *how well* it should be documented, and *why/what-for* these documents are needed.

Design – The description of the design of an information system. These descriptions should identify *how* an information system will meet the requirements set out in its definition. The resulting design may (depending on the design goals) range from high-level designs to the detailed level of programming statements or specific worker tasks.

Deployment – The processes of delevering/implementating an information system to/in its usage context. The design of an information system is not enough to arrive at an operational system. It needs to be implemented-in/delivered-to a usage context.

Maintenance – An information system which is operational in its usage context, does not remain operational by itself. Both technical and non-technical elements of the system need active maintenance to keep the information system operational as is.

Architecting – The processes which tie definition, design, deployment and maintenance to the explicit and implicit needs, desires and requirements of the usage context. Issues such as: business/IT alignment, stakeholders, limiting design freedom, negotiation between stakeholders, enterprise architectures, stakeholder communication, and outsourcing, typify these processes.

Domain modeling – Modeling of the domains that are relevant to the information system being developed. The resulting models will typically correspond to *ontologies* of the domains. These domains can pertain to the information that will be processed by the information system, the processes in which the information system will play a role, the processing as it will occur inside the information system, etc. Understanding (and modeling) these domains is fundamental to the other activities in information system engineering.

For each these aspects, attention will be paid to relevant theories, methods and techniques to execute the tasks involved. When put together, these aspects can be related as depicted in Figure 1.1.

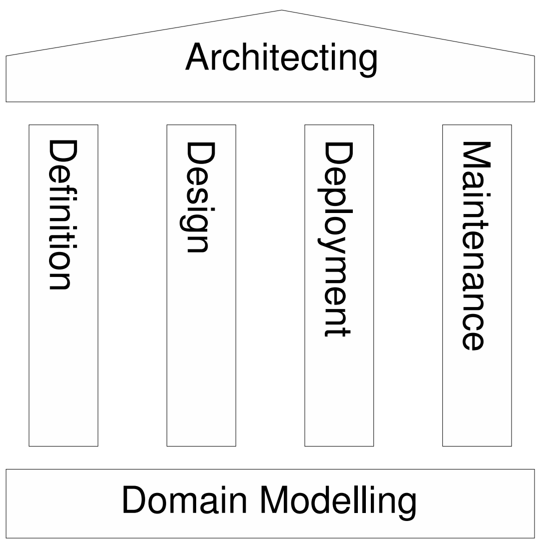


Figure 1.1: Aspects of Information Systems Engineering

Chapter 2

History

The use of the name DAVINCI originates from earlier work [Pro98] done on architecture-driven information systems engineering. The work reported in [Pro98] was the result of a confrontation between industrial practice and a theoretical perspective on information systems and their evolution [Pro94]. The result was a shared vision on *the architecture-driven development of information systems* as it was held by architects at a Dutch IT consultancy firm. In this shared vision, a foundation was laid for an integrated view on information system engineering. At that stage, the name "DAVINCI" was also selected. Not as some artificial acronym, but rather to honour an inspiring artist, scientist, inventor and architect. To us he personifies a balance between art and engineering, between human and technology.

After the development of the first DAVINCI version, a more elaborate version [Pro04] was developed at the Radboud University in the form of lecture notes associated to a course on *Architecture & Alignment*. In this second version, a more fundamental outlook on information system development was added to complement the practical orientation of the first version.

As a third step, we have now taken on the underlying philosophy of the first two DAVINCI documents, and used this as the source of inspiration to shape an entire line of lecture notes for a number of mutually related courses on different aspects of information systems engineering. In making this step we have also been able to anchor some of the fundamental research results from the co-authors, on subjects such as information modeling [BHW91, BW92a, HW93, HPW93, PW94, BBMP95, CHP96, CP96, PW95, BFW96, HPW97, Pro97, HVH97, FW02, FW04], information retrieval [BW90, BW91, BW92b, BB97, WBW00, SFG+00, PB99, PPY01, WBW01], (enterprise) information architecture [JLB+04] and information systems engineering [Pro01, VHP04] into the core of the DAVINCI series.

Chapter 3

Structure of the series

The series is structured into six tiers, which relate to the increasing maturity of students during their Bachelor and Master's studies. The courses in each tier can be attended/rostered in parallel. In Figure 3.1 we have depicted the dependencies between the tiers. The arrows between the tiers should be interpreted as "if $x \to y$, then tier x is a prerequisite to tier y".

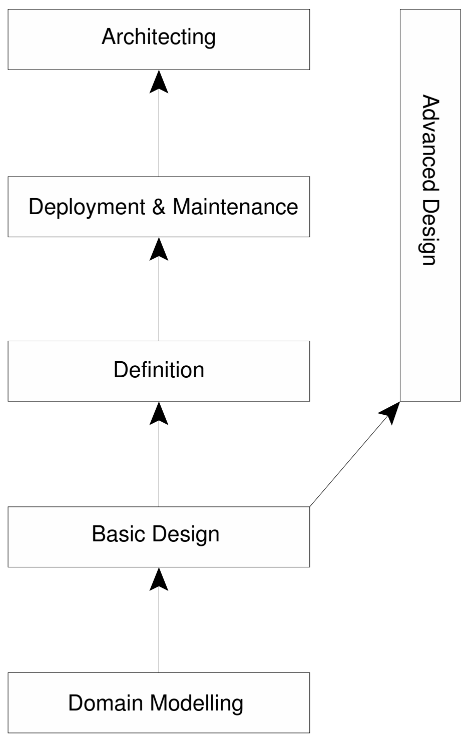


Figure 3.1: Tiers in the DAVINCI series

Note: the next section discusses the mapping of these courses to the Information & Computing Science curricula at the Radboud University

3.1 Domain modeling

The series starts with tier dedicated to domain modeling. This first tier consists of only one course.

Domain Modelling

Short description

This course covers an important prerequisite in the information system life-cycle where the intention is to obtain an (agreed) understanding of a given domain. In this course, students will be taught to demarkate a domain and identify the ontology of the domain, i.e. a specification of its conceptualization comprising the core concepts in the domain, their mutual relations and the laws (constraints) governing their behaviour.

Learning goals

1. Het verkrijgen van modelleervaardigheden en inzicht in de redenen om te modelleren.
2. Kennismaking met UML, SQL, elementen uit de systeemtheorie
3. Leren een onderscheid te maken tussen: het gemodelleerde domein, de waarnemer, het beeld van het domein zoals dit leeft bij de waarnemer en de beschrijving van het domein in een taal (natuurlijke taal of een formele taal).
4. Bewust zijn van de invloed die de gebruikte beschrijvingstaal kan hebben op het beeld van een waarnemer van het domein. Met een hamer in de hand is alles een spijker ...
5. Begrip van de plaats van modelleren binnen de information system engineering.

Topics

1. backgrounds of modelling
2. the role of communication
3. deriving a conceptual model
4. the formal meaning of a model
5. quality aspects: validation ad verification
6. transformation to other kind of models

Outline

In dit vak zul je kennis maken met het modelleren van domeinen (een stuk van "de werkelijkheid die we waarnemen"). Dit zullen we in eerste instantie doen aan de hand van concrete voorbeelden waarbij studenten middels de UML technieken aspecten van organisaties, informatiesystemen, etc, dienen te modelleren. SQL zal hierbij gebruikt worden om een UML klassediagrammen te concretiseren in termen van een implementatie, en Prolog zal gebruikt worden om kennis te maken met de semantiek van logische eigenschappen van domeinen (bijvoorbeeld de semantiek van constraints in UML). In dit vak wordt ook beoogd om de studenten kennis te laten maken met het feit dat eenzelfde domein, vanuit verschillende waarnemers en perspectieven gemodelleerd kan worden. Gaandeweg het semester zullen we de overstap maken naar een wat abstracter denkniveau en overgaan naar het denken in termen van systemen obv systeemtheorie.

3.2 Basic design

This second tier introduces students to some basic design issues of information systems. It consists of two courses which consider information systems from two complementary angles: the *Information Intensive Organizations* course considers information systems from an organizational perspective, while *Computerized Information Systems* zooms in on fully computerized information systems. Both courses build further on the foundations laid in the *Domain Modeling* course.

Computerised Information Systems

Short description

For the design of a computerised information system, it is important to consider the mapping of the conceptual structure, its processes and its data, onto the underlying storage devices.

Learning goals

1. Na afloop van dit vak kan een student verschillende technieken voor de opslag en ontsluiting van gegevens duiden.
2. Na afloop is een student tevens in staat om voor een gegeven situatie een beargumenteerde keuze maken.
3. Verder zijn studenten in staat om voor enkele technieken programma's te schrijven waarin deze worden toegepast.

Topics

Aan bod komen, in toenemende volgorde van verfijndheid, een aantal methoden voor opslag van informatie, met de daarbij behorende algoritmen voor opslaan en opvragen. Achtereenvolgens bespreken we:

1. Storage devices and their parameters
2. Design issues (reliability, availability, mtbf, etc.)
3. Data organization as a file system (unix, fat, ntfs, winfs, longhorn)
4. Internal file organization techniques (record file, isam, hashing, b-tree, text file, xml, efficiency)
5. Relational algebra as a calculation schema and base for query optimization (transformation sql to relational algebra)
6. Why do we store information: essentials of information retrieval (search engine, sql)
7. elementaire gegevensopslag met behulp van fundamentele opslagtechnieken (B-bomen, hashing),
8. meerdere toegangen: het indexed-sequential bestand,
9. de tabel en de relatie, de hierop gebaseerde relationele algebra en de relatie met SQL databases,
10. de geneste tabel en de relatie met Object-Oriented databases,
11. XML gebaseerde opslag en de relatie met het Internet,
12. opslag van ongestructureerde informatie en de relatie met Information Retrieval en methoden om exploratief te zoeken in grote hoeveelheden informatie, etc.
13. recursieve uitbreidingen van de relationele algebra (Datalog)

Outline

Dit vak beoogt studenten kennis te laten kennis met een breed scala aan technieken met hun achterliggende theorie, voor het efficient opslaan en ontsluiten van informatie. This course discusses storage media, their parameters and the essentials of storage strategies. The students learn how an information model can be mapped onto a storage strategy.

Naast een praktische kennismaking met opslagtechnieken komt ook het redeneren in termen van de tijd/ruimte uitruil aan bod, zodanig dat de deelnemers na afloop voldoende inzicht hebben in een concrete situatie een beargumenteerde keuze te maken voor de te kiezen methode van opslag.

Information Intensive Organizations

Short description

Organizations can by found anywhere. A University is an organization, a sports club is an organization, a bank is one, government departments are, etc. Organizations are everywhere. In our modern western society, most organizations use some information systems to support the activities of the organization. These organizations, and their information processing activities, will be referred to as *information intensive organizations*.

Learning goals

After this course, students are able to:

1. given a case-description of an organization:
   (a) produce models for different aspects of this organization,
   (b) understand and evaluate given models of that organization,
2. argue about, and prove, properties regarding the syntax & semantics of models of organizations,
3. reason about the link between an organization's strategy, and the services & processes it uses to realize this strategy,
4. model the evolution of an organization's structures,
5. reason about the position of modeling (information intensive) organizations in the context of information systems engineering.

Topics

1. Organizations.
2. Temporal ordering, actors, actions, actands.
3. Decomposition of actors, actions and actands.
4. Organizational processes.
5. Organizational services.
6. Organizational strategy.
7. Describing evolution.

Outline

The focus of this course is on the modeling of different aspects of information intensive organizations. It does so, by building on top of the general modeling foundation laid in the *Domain Modeling* course. We will follow a bottom up process in discussing different aspects of organizations, where we will first look at concepts needed to describe the structures of an information intensive organizations, then look at the relationship to their strategies, and finish by discussing a way to describe their evolution.

The course starts with a brief discussion of the concept of organization, and links this to the more general notion of *system*. We regard organizations essentially as *open & active systems*.

We then continue by using the modeling approach from *Domain Modeling*, and applying this to the modeling of organizations and their information processing. This requires the immediate introduction of some new core concepts such as: temporal ordering, action, actor, actand. These concepts allow us to reason about such things as: *when* does something happens (temporal ordering), *what* happens (action), *who/what* makes it happen (actor), and *who/what* does it happen *to* (actand).

With these new concepts, we can take the ORM domain models that result from the *Domain Modeling* course, and "annotate" them in terms of the new concepts. We will base this process of annotation on linguistic foundations, much in the same vein as the modeling approach from the *Domain Modeling* course. Based on these annotated ORM models, we are able to mechanically derive process models in a modeling notation (Testbed & ArchiMate) that is particularly suited for the modeling of business processes. For these process models, we will study some formal properties covering both syntactical and semantical issues.

Using the Testbed & ArchiMate models, we can study workflows that underly an organization's (business) processes. Even more, we can identify data flows (which hopefully result in information/knowledge flow between the actors in an organization).

Once we have studied processes and flows, we can move on to a more abstract level of viewing organizations. We can view organizational units essentially as being composed actors that provide *services* to their outside world. We will therefore discuss techniques (mostly from ArchiMate) to model these services in more detail.

With the introduction of services we have completed our discussion of the key concepts needed to describe the structures of information intensive organizations. This allows us to move on to the relationship between an organization's structures and its strategy and mission.

Thus far, we have focused on organizations where we implicitly presumed the structures of the organization to remain stable. This is, for obvious reasons, not a realistic assumption. To finalize this course, we will therefore discuss a way to describe the past/planned evolution of an organization and its structures.

3.3 Advanced design

In this tier, the design techniques from previous parts of this series are the topic of a more in-depth study. This leads to further reflections on the essentials of these techniques, and enables the student to reason in-depth about such models, their properties and their qualities, and to make comparisons between models. To allow for a proper in depth study, this tier is divided into four courses.

This tier is optional before progressing to the final tier. The courses in this tier can also be attended in parallel to the courses of the next two tiers. However, students are advised to attend at least one of the courses from this tier during their studies. Note that some of these courses are allocated to the Bachelor courses, while others are to the Master courses. This allocation may evolve over time.

Foundations of Information Systems

Short description

In this course, the fundamental techniques from previous parts of this series are the topic of an in-depth study. This leads to a reflection of the essentials of such techniques, and enables the student to reason about such models, their properties and their qualities, and to make comparisons between models.

The nature of this course is research-oriented, the students are challenged to do research to advance the state-of-the-arts of both the craftsmanship and the art of modelling.

Learning goals

1. De syntax en semantiek van informatiestructuren kennen.
2. De syntax en semantiek van veel voorkomende constrainttypen. Bovendien: tegenspraken in constraints bewijzen.
3. De formele definitie van identificeerbaarheid kennen en toepassen in concrete situaties en daarvan de theoretische consequenties overzien.
4. Het formuleren van theoretische uitspraken voor wat betreft complexiteit van verificatie en expressieve kracht van gegevensmodellen.
5. De syntax en semantiek van padexpressies kunnen begrijpen en toepassen

Topics

1. Foundations of information modelling techniques. Data models and process models.
2. Comparison of fact-oriented modelling and object-oriented modelling.
3. Abstraction mechanisms.
4. Semantics of complex integrity constraints. Static constraints versus dynamic constraints.
5. Meta data models of data models and process models.
6. Meta process models of data models and process models.
7. Artificial identification and real identification.
8. Validation of information models.
9. Verification and enforcement.
10. Path expressions through information structures.
11. Transformation of data models. Structure transformations. Guided transformations. Preprocessing and postprocessing.
12. Complexity analysis.
13. Transformation of database contents and integrity constraints.
14. Transformation of database operations (retrieval and updates). Proof schema for structural induction.
15. Correctness and completeness of transformations.
16. Repairment and penalization for incorrect transformations.
17. Equivalence of data models.
18. Distance between data models. Comparison of distance measures.
19. Modelling of web sites and semistructured data.
20. Object graphs, object coordination graphs, path definition and evaluation.
21. Representation language for data views.
22. Encoding of data views. Augmentations of basic encoding.

23. Database design and optimization. Hill climbing. Evolutionary algormithms. Genetic algorithms.
24. Solution spaces. Size, structure, and compactness of solution spaces.
25. Properties of solutaion space (e.g. local and global optima) versus properties of transformation algorithms (e.g. living and dead database structures).
26. Markov Chain analysis.
27. Relationship with ER, NIAM, UML, XML, OO, RDBMS, ORDBMS, SQL

Outline

Dit vak behandelt informatiemodellering, een essentieel onderdeel van systeemanalyse. Er wordt ook gewerkt aan de modellering van websites. Centraal staat de vraag hoe we op een nette manier de semantiek van informatie- modellen kunnen vastleggen.

Deze cursus behandelt informatiemodellering, een essentieel onderdeel van systeemanalyse ("Requirements Analysis"). Er wordt een modelleringstechniek besproken, die voldoet aan de volgende eisen. De techniek heeft een formele onderbouwing, is op een conceptueel nivo, heeft een ruime expressieve kracht, is executeerbaar, leidt tot begrijpelijke en communiceerbare modellen, en is geschikt voor het modelleren van gegevensintensieve domeinen. Communiceerbaarheid is een belangrijke eis, omdat conceptuele modellen een cruciale rol spelen in de communicatie met domeinexperts. De modelleringstechniek bevat naast de gebruikelijke gegevensmodellering ook aspecten van Object Oriëntatie, bijvoorbeeld Object Life Cycle modellen. Er wordt ook gewerkt aan de modellering van websites, waarbij elke student een eigen website modelleert volgens de principes van datamodellering. Wat je gaat leren: (1) Het belang van formele methoden kunnen beargumenteren. (2) De syntax en semantiek van gegevensschema's zodanig begrijpen dat je complexe domeinen kunt modelleren.

Foundations of Information Retrieval

Short description

IR (Een constructieve visie op Information Retrieval) behandelt de achtergronden van Information Retrieval: (1) hoe zoeken mensen informatie, hoe formaliseren we dat, (2) hoe beschrijven mensen wat ze bedoelen, hoe formaliseren we 'betekenis', en (3) hoe brengen we (1) en (2) met elkaar in verband? Een belangrijk toepassingsgebied is het Internet.

Learning goals

De bedoeling van IR (Een constructieve visie op Information Retrieval) is dat deelnemers na afloop:

1. Bekend zijn met de basismodellen voor Information Retrieval.
2. Kennis hebben van query talen, zowel syntactisch als semantisch.
3. Elementaire kennis hebben van het indexeren van teksten
4. Inzicht en vaardigheid in ontwerp en bouw van zoekmachines
5. Bekend zijn met document-representatie methoden, hun achterliggende motivatie en begrenzingen.
6. Kennis hebben automatische documentclassicatie: de probleemstelling en algoritmen.
7. bekend zijn met de basisbegrippen en -technieken van datamining.
8. Vaardigheid bij het toepassen van tools op zoek naar begrijpelijke patronen en kennis.

9. Kennis van visualisatietechnieken voor kennisstructuren.
10. Vaardigheid in het toepassen hiervan bij de ontsluiting van een kennisbron.

Topics

Na een algemene formulering van de probleemstelling worden de bekende retrieval modellen besproken (zoals vector space model en boolean retrieval) en gekoppeld aan een computationele (i.h.b. computationele) visie. Voor de evaluatie van een retrieval systeem worden een aantal maten ingevoerd.

Hierna worden query talen besproken, en manieren om de gebruiker te helpen bij de formulering van de query door verrijkings- en feedback methoden.

Kennisextractie uit teksten wordt inleidend besproken. Hierna wordt een keuze gemaakt uit de volgende onderwerpen: multimedia IR, zoekmachines en digital libraries.

Outline

Het opslaan van grote hoeveelheden zwak gestructureerde informatie stelt extra eisen aan de methoden om hieruit gericht informatie op te vragen. Voorbeelden zijn bibliotheken en het internet. Op het internet wordt veel gebruik gemaakt van zoekmachines. Deze machines proberen zoveel informatie te vergaren over wat op het internet aangeboden wordt, en stellen gebruikers in staat hierover vragen te stellen. Naast deze elementaire retrieval functie kunnen complexere retrieval taken beschouwd worden. Bijvoorbeeld, het filteren van berichten die via een nieuwsgroep verspreid worden. Of het bouwen van een systeem waarin complexe vragen gesteld worden (question-answering systemen), zoals: wat is het totaal aan studiepunten van vakken die over gegevensmodellering gaan?

Omdat de toegankelijkheid van de informatie groot moet zijn, dient de vraagtaal van een zoekmachine erg krachtig en eenvoudig bruikbaar te zijn. Die laatste eis maakt het probleem extra lastig. Een mooi gemaakt retrieval systeem scoort slecht wanneer het (brede) publiek er onvoldoende mee uit de voeten kan. De interactie tussen de gebruiker van zo'n systeem vindt op 2 momenten plaats: tijdens de probleemformulering en tijdens de presentatie van de gevonden documenten. Daartussen is het systeem bezig de geformuleerde vraag te koppelen aan documenten. Hoe beter de zoekmachine een beeld kan vormen over de inhoud van documenten, hoe beter deze matching zal zijn.

Agent-based Information Systems

Short description

The purpose of this course is to introduce the paradigm of intelligent agency. Intelligent agents are software programs that can autonomously perform tasks for users. The ideal agent can perceive its environment, communicate with other agents, and take a series of actions to achieve a complex goal. The potential applications of agents are numerous: including web search assistants, travel advisors, electronic secretaries, and bidders in on-line auctions. The course will cover the underlying theory of agents, the common agent architectures, the potential applications for agents, and will let students construct their own agents.

Learning goals

1. design and construct intelligent agents,
2. apply formal models and logic of rational agency,
3. determine in which situations agent architectures should be used.

Topics

Planning, plan recognition, logic of beliefs, desires and intentions, game theory, autonomy, adaptation, reactivity and proactivity, knowledge representation (tasks, environment), bounded rationality, Sense-Plan-Act cycle, agent architectures, uncertainty reasoning, communication and coordination, multi-agent systems Leerdoelen

Outline

To be done.

Knowledge-based Information Systems

Short description

Dit vak heeft tot doel de cursist kennis te laten maken met de belangrijkste onderwerpen uit de Kunstmatige Intelligentie: kennisrepresentatie, redeneren en leren. Daarnaast wordt aandacht besteed aan de geschiedenis van kennissystemen, de ontwikkeling van kennissystemen en de formele basis ervan. Het omgaan met onzekerheid in kennissystemen is ook een onderwerp dat in een zekere mate van detail bestudeerd zal worden. Tenslotte zullen cursisten enkele talen en tools die gebruikt worden in de Kunstmatige Intelligentie leren kennen, en ook zelf een kennissysteem hiermee ontwikkelen.

Learning goals

1. Kennismaken met de belangrijkste onderwerpen in de kunstmatige intelligentie (AI), zoals probleemoplossen, toestandsruimte, heuristisch zoeken, kennisrepresentatie en automatisch redeneren, machinaal leren, intelligent agent, kennisacquisitie, knowledge engineering.
2. Inzicht opdoen in de toepassing van algoritmische, logische en wiskundige methoden in de kunstmatige intelligentie.
3. Ervaring opdoen met de ontwikkeling van een kennissysteem voor een concreet domein.
4. Ervaring opdoen in de ontwikkeling van AI-programma's.

Topics

1. Wat is kunstmatige intelligentie?
2. Geschiedenis: GPS, MYCIN, DENDRAL, ATP.
3. Probleemoplossen en zoeken in toestandsruimten: uitputtend zoeken, backtracking, heuristisch zoeken op basis van A*, hill climbing, tabu search, simulated annealing.
4. Kennisrepresentatie en automatisch redeneren: produktieregels en redeneren, logica en resolutie, Herbrand universum en basis, object representatie in de AI, redeneren met onzekere kennis, produktieregels en onzekerheid, Bayesiaanse netwerken.
5. Knowledge engineering: methodology van de ontwikkeling van kennissystemen, machinaal leren.
6. AI programmeren in een daarvoor geschikte taal.

Outline

In dit vak zullen onderwerpen uit de computationele intelligentie, zoals representaties bij automatisch leren (beslisbomen, classificatieregels, neurale netwerken) aan de orde komen, naast onderwerpen uit de information retrieval met betrekking tot kennisextractie uit concrete bronnen (text mining, concepttralies, extentionele semantiek).

Daarnaast zal aandacht besteed worden aan probleemklasse-specifieke redeneervormen, zoals case-based retrieval and reasoning, model-gebaseerd redeneren, spatieel redeneren, en redeneren met ontwerpspecificaties.

Dit vak biedt een inleiding op de kunstmatige intelligentie, in het bijzonder de onderwerpen probleemoplossers, kennisrepresentatie, automatisch redeneren, kennisacquisitie en knowledge engineering.

3.4 Definition of Information Systems

At this point in the series, students are able to produce an underlying ontology (conceptual model) for a given domain, produce models for different aspects of information intensive organizations, and is is aware of design parameters and design strategies of both information-intensive organizations and computerized information systems. This provides them with a background that allows them to start learning about the *requirements* that can be put on information systems. In other words, the definition of information systems. This tier comprises one course only: *Definition of Information Systems*.

Note: It would probably be natural to also involve a course like *Quality of Information Systems* in this tier. The course *Definition of Information Systems* provides insight into *what* qualities should be provided by an information system, while the latter course would focus on how to ensure that an information system being engineered will meet these qualities and/or to assess whether a fully designed (or even operational) information system meets these qualities.

Definition of Information Systems

Short description

Een inleiding in de kunst en wetenschap van "Requirements Engineering" (RE): het boven water krijgen en uitspecificeren van wat opdrachtgevers en gebruikers eisen en verwachten van een toekomstig informatiesysteem.

Naast het bestuderen (tekstboek, colleges) van en oefenen (casus) met een methode voor requirements gathering worden enige meer wetenschappelijke aspecten belicht. Tevens wordt stilgestaan bij praktijkissues.

Een belangrijk deelproces van systeemontwikkeling is het goed helder krijgen van de behoeften. Dit is het "requirements engineering" proces. Het belangrijkste doel van dit proces is er voor te zorgen dat een eventueel te ontwikkelen systeem ook het goede systeem is. Centraal in dit proces staat de vergaring, specificatie, validatie en de bewaking van de systeemeisen.

In dit vak leren de studenten hoe ze moeten komen tot een goede requirementsspecificatie, hoe ze die moeten valideren, en hoe ze deze bij de daadwerkelijke realisatie en invoering van een systeem kunnen bewaken.

Learning goals

1. De student is in staat in een gegeven organisatorische context een informatievoorzieningsprobleem te analyseren en samen met de belanghebbenden in die organisatie een programma van eisen voor een nieuw (in te voeren of te wijzigen) informatiesysteem(deel) opstellen.
2. De student is in staat om op basis van enerzijds de eisen mbt. een informatiesysteem en anderzijds een gereed produkt (elders gebouwd maatwerk informatiesysteem danwel een 'getuned' multipurpose softwarepakket) de bruikbaarheid van dat produkt aan te beoordelen en te beargumenteren.

3. Requirements specificaties onderscheiden van designspecificaties
4. Requirements specificaties beoordelen op hun kwaliteit
5. Requirements engineering projecten ruwweg plannen, en ruwe planningen (aanpakken) evalueren
6. Zelf heldere en zorgvuldig verzamelde en gespecificeerde requirements opleveren gegeven voldoende goede en toegankelijke informatiebronnen (stakeholders, documenten)
7. Uitleggen wat de plaats is van requirements documenten binnen verdere documentatie in organisaties en systeemontwikkelprojecten
8. Uitleggen hoe processen voor requirements engineering te plaatsen zijn binnen systeemontwikkelingsprocessen in het algemeen
9. Vanuit systeemontwikkelingstheorie reflecteren op het vakgebied requirements engineering en op het RE proces

Topics

1. Wat zijn requirements? (functional en non-functional)
2. Verzamelen van requirements
3. Specificeren (nauwkeurig opschrijven) van requirements
4. Fasering en planning in RE
5. Use Cases
6. Business rules
7. Risk analysis
8. Stakeholder Analysis
9. Requirements engineering in systeemontwikkelingstheorie
10. Requirements en Taal
11. Praktijk van RE

Outline

To be done.

3.5 Deployment & Maintenance

Once students have reached this tier, they should have obtained an appreciation for the issues involved in the definition and design of information systems. However, a fully designed information system, even if all of the underlying algorithms and work processes have been specified in full detail, is far from being an *operational* information system.

The focus of this tier is therefore on deploying (implementing/delivering) a designed information system in its usage context, and making it into an operational information system, and *maintaining* it in an operational state. The courses in this tier will address both technical and non-technical issues.

Deployment & Maintenance of Information Systems

Short description

This course focusses both on the deployment of a fully designed/constructed information system in its usage context, as well as the processes and measures needed to maintain such a system in an operational state.

In this course, attention will be paid to some public standards (ISPL, ITIL, ASL) which are relevant to the deployment, maintenance and servicing of information systems. Specific attention will also be paid to so-called service-level agreements (SLA's) with regards to the operational maintenance of systems, the risks that may threaten these service levels, and strategies to manage these risks.

Learning goals

After this course, students are able to:

1. reason about the role of deploytment and maintenance in information systems engineering.
2. identify, and discuss, the major elements in some relevant public standards for deployment and maintenance,
3. apply the theories underlying these public standards to some practical cases,
4. design and discuss organisational structures/processes for deployment and maintanance,
5. reason about service level agreements and their impact on both the using environment and the maintenance processes,
6. assess potential risks to these service level agreements.

Topics

1. Relevant public standards:

   ISPL – Information Services Procurement Library.
   ITIL – Information Technology Infrastructure Library.
   ASL – Application Services Library.
2. Deployment
3. Maintenance
4. Service Level Agreements (SLA),
5. Risk management

Outline

To be done

3.6 Architecting

In this last tier, students are presumed to have matured enough to be presented with issues such regarding the management of IT facilities and projects, outsourcing, a high level view on design processes, architecting of enterprise-wide information systems, etc. This tier consists of two courses. These courses focus on a future role of students as an architect (design lead), rather than a manager (project lead).

Information Systems Architecture

Short description

This course comprises three parts. It starts by discussing information systems and information systems engineering from an abstract point of view. As such, it ties the courses of the previous courses in the DAVINCI series together. This abstract view translates to taking a system theoretical perspective on information systems, their engineering and the role of architecture in bounding/guiding the engineering processes.

We then move on to the architecture level, by discussing both the definition of architecture in an information systems engineering context, its need, as well as its potential role as a means of negotiation and communication. Special attention will be paid to the mechanism of *architecture principles* as a mechanism to guide/bound the engineering processes.

In the third part, we zoom in on aspects of the process of architecting information systems, in particular on communication strategies and negotiation between different stakeholders. An information system involves organisational, human and technological issues, leading to a plethora of stakes and stakeholders. These stakes need to be balanced during the development of an information system's architecture.

Learning goals

After this course, students are able to:

1. argue about the different aspects of information system engineering and position the methods & techniques as presented in the other courses in the DAVINCI series relative to these aspects,
2. reason about organisations, information systems, and computerised information systems at an abstract (system theoretical) level,
3. argue the need, and potential role, for architecture in the context of information systems engineering,
4. discuss and relate different definitions of architecture,
5. reason about different levels of abstraction at which an information system, and its context, can be modelled/designed/architected, as well as identify situations for which these levels are relevant,
6. reason about the selection of modelling techniques, for given goals and situations during information systems engineering, that are most apt to the situation at hand,
7. develop, and reason about, communication and negotiation strategies relevant to information systems engineering.

Topics

1. System theory
2. Viewpoints
3. Architecture
4. Architecture principles
5. Communication & negotiation
6. Stakeholders

Outline

To be done

Enterprise Information Architecture

Short description

De kwaliteit van de informatievoorziening is voor moderne organisaties een succesfactor van belang. In veel gevallen is die voorziening het resultaat van ongeleide 'organische' groei. Vaak is dan tegelijk met de toename van het aantal technische en informatorische voorzieningen het geheel uitgegroeid tot een onontwarbare kluwen digitale onderdelen.

De kosten zijn toegenomen en de inflexibiliteit eveneens terwijl het reactievermogen van de organisatie juist groter moet worden en daardoor extra eisen stelt aan de souplesse van de informatievoorziening.

In dit vak wordt deze situatie nader geanalyseerd en ontleed, en worden concepten ontwikkeld om de structurele veroudering en verstarring tegen te gaan. Derhalve wordt veel aandacht besteed aan 'functies en constructies' van de informatievoorziening en aan beheersing van de complexiteit.

Architectuur en informatie-infrastructuur zijn daarbij belangrijke begrippen, evenals architectuurprincipes.

Note: Maybe we should also pay attention to outsourcing. Both IT services (ISPL!) as well as entire business processes (BPO). By aligning the *Information Systems Architecture* course and the *Enterprise Information Architecture* course, there may be more room in the latter course for oursourcing.

Learning goals

1. het concept architectuur en enkele van de definities die hieraan gegeven worden, met nadruk op de rol van architectuur als communicatie- en onderhandelingsmiddel.
2. de rol van digitale architectuur in ondernemingen en organisaties mbt het bereiken van een betere afstemming tussen de bedrijfsvoering en de ICT (business-ICT alignment).
3. adaptiviteitscriteria die organisaties in staat stellen om beter in te spelen op de immer veranderende, nauwelijks voorspelbare omgeving.
4. belangrijke raamwerken en patronen met betrekking tot architectuur, zoals service oriëntatie, etc.
5. in te schatten welke impact architectuurkeuzes hebben.
6. relevante wetenschappelijke/professionele literatuur op het gebied van architectuur te duiden en de relevantie ervan te bepalen voor een gegeven probleemsituatie.
7. de noodzaak te duiden van een goede afstemming tussen bedrijfsvoering en ICT.
8. te beredeneren welke adaptiviteitscriteria relevant zouden kunnen zijn.
9. een eerste inschatting te maken welke raamwerken en patronen met betrekking tot architectuur relevant zijn.

Topics

1. Digitale Wereld
2. Informatiearchitectuur
3. Architectuurprincipes

Outline

To be done.

Chapter 4

Implementation at the ICIS

The DAVINCI series forms a main line of the Information & Computing Science curricula at the ICIS, both on the bachelor and the master level.

The above identified courses map to the following courses in these curricula for 2004/2005:

DAVINCI	ICIS	IS	CS
Domain Modeling	Domeinmodelleren	$\times$	$\times$
Computerized Information Systems	Opslaan & terugvinden	$\times$	$[\times]$
Organizational Information Systems	Modelleren van organisaties	$\times$	$[\times]$
Definition of Information Systems	Requirements engineering	$\times$	$\times$
Foundations of Information Systems	Informatiesystemen	$\times$	$\times$
Foundations of Information Retrieval	Information retrieval	$[\times]$	$[\times]$
Agent-based Information Systems	Intelligente agenten	$[\times]$	$[\times]$
Knowledge-based Information Systems	Intelligente systemen	$[\times]$	$[\times]$
Deployment & Maintenance of Information Systems	ICT Management (3 ec)	$[\times]$	$[\times]$
	Informatieverzorging (3ec)	$[\times]$	$[\times]$
Information Systems Architecture	ST: Ontwerp en evolutie (3ec)	$\times$	$[\times]$
	O&V: Design rationale (3ec)	$\times$	$[\times]$
Enterprise Information Architecture	Informatiearchitectuur	$\times$	$[\times]$

Note: a $\times$ signifies that a course is mandatory for the specific curriculum, while a $[\times]$ signifies that the course is optional. All courses are 6ec, unless stated otherwise.

Bibliography

[BB97] F.C. Berger and P. van Bommel. Augmenting a characterization network with semantical information. *Information Processing & Management*, 33(4):453–479, 1997.

[BBMP95] G.H.W.M. Bronts, S.J. Brouwer, C.L.J. Martens, and H.A. Proper. A Unifying Object Role Modelling Approach. *Information Systems*, 20(3):213–235, 1995.

[BFW96] P. van Bommel, P.J.M. Frederiks, and Th.P. van der Weide. Object-Oriented Modeling based on Logbooks. *The Computer Journal*, 39(9):793–799, 1996.

[BHW91] P. van Bommel, A.H.M. ter Hofstede, and Th.P. van der Weide. Semantics and verification of object-role models. *Information Systems*, 16(5):471–495, October 1991.

[BW90] P.D. Bruza and Th.P. van der Weide. Assessing the Quality of Hypertext Views. *ACM SIGIR FORUM (Refereed Section)*, 24(3):6–25, 1990.

[BW91] P.D. Bruza and Th.P. van der Weide. The modelling and retrieval of documents using index expressions. *ACM SIGIR FORUM (Refereed Section)*, 25(2), 1991.

[BW92a] P. van Bommel and Th.P. van der Weide. Reducing the search space for conceptual schema transformation. *Data & Knowledge Engineering*, 8:269–292, 1992.

[BW92b] P.D. Bruza and Th.P. van der Weide. Stratified Hypermedia Structures for Information Disclosure. *The Computer Journal*, 35(3):208–220, 1992.

[CHP96] L.J. Campbell, T.A. Halpin, and H.A. Proper. Conceptual Schemas with Abstractions – Making flat conceptual schemas more comprehensible. *Data & Knowledge Engineering*, 20(1):39–85, 1996.

[CP96] P.N. Creasy and H.A. Proper. A Generic Model for 3-Dimensional Conceptual Modelling. *Data & Knowledge Engineering*, 20(2):119–162, 1996.

[FVSV+98] E.D. Falkenberg, A.A. Verrijn-Stuart, K. Voss, W. Hesse, P. Lindgreen, B.E. Nilsson, J.L.H. Oei, C. Rolland, and R.K. and Stamper, editors. *A Framework of Information Systems Concepts*. IFIP WG 8.1 Task Group FRISCO, IFIP, Laxenburg, Austria, EU, 1998. ISBN 3-901-88201-4

[FW02] P.J.M. Frederiks and Th.P. van der Weide. Deriving and paraphrasing information grammars using object-oriented analysis models. *Acta Informatica*, 38(7):437–88, June 2002.

[FW04] P.J.M. Frederiks and Th.P. van der Weide. Information modeling: the process and the required competencies of its participants. *Data & Knowledge Engineering*, 2004. To appear in a special issue on the NLDB 2004 conference.

[Hop03] S.J.B.A. Hoppenbrouwers. *Freezing Language; Conceptualisation processes in ICT supported organisations*. PhD thesis, University of Nijmegen, Nijmegen, The Netherlands, EU, 2003. ISBN 90-9017318-8

[HPW93] A.H.M. ter Hofstede, H.A. (Erik) Proper, and Th.P. van der Weide. Formal definition of a conceptual language for the description and manipulation of information models. *Information Systems*, 18(7):489–523, October 1993.

[HPW97] A.H.M. ter Hofstede, H.A. Proper, and Th.P. van der Weide. Exploiting Fact Verbalisation in Conceptual Information Modelling. *Information Systems*, 22(6/7):349–385, September 1997.

[HVH97] J.J.A.C. Hoppenbrouwers, B. van der Vos, and S.J.B.A. Hoppenbrouwers. Nl structures and conceptual modelling: Grammalizing for KISS. *Data & Knowledge Engineering*, 23(1):79–92, 1997.

[HW93] A.H.M. ter Hofstede and Th.P. van der Weide. Expressiveness in conceptual data modelling. *Data & Knowledge Engineering*, 10(1):65–100, February 1993.

[JLB+04] H. Jonkers, M.M Lankhorst, R. van Buuren, S.J.B.A. Hoppenbrouwers, M. Bonsangue, and L. van der Torre. Concepts for Modeling Enterprise Architectures. *International Journal of Cooperative Information Systems*, 13(3):257–288, 2004.

[PB99] H.A. Proper and P.D. Bruza. What is Information Discovery About? *Journal of the American Society for Information Science*, 50(9):737–750, July 1999.

[PPY01] M.P. Papazoglou, H.A. Proper, and J. Yang. Landscaping the information space of large multi-database networks. *Data & Knowledge Engineering*, 36(3):251–281, 2001.

[Pro94] H.A. Proper. *A Theory for Conceptual Modelling of Evolving Application Domains*. PhD thesis, University of Nijmegen, Nijmegen, The Netherlands, EU, 1994. ISBN 90-9006849-X

[Pro97] H.A. Proper. Data Schema Design as a Schema Evolution Process. *Data & Knowledge Engineering*, 22(2):159–189, 1997.

[Pro98] H.A. Proper. Da Vinci – Architecture-Driven Business Solutions. Technical report, Origin, Amsterdam, The Netherlands, EU, Summer 1998.

[Pro01] H.A. Proper, editor. *ISP for Large-scale Migrations*. Information Services Procurement Library. ten Hagen & Stam, Den Haag, The Netherlands, EU, 2001. ISBN 9076304882

[Pro04] H.A. Proper. *Da Vinci – Architecture-driven Information Systems Engineering*. Nijmegen Institute for Information and Computing Sciences, University of Nijmegen, Nijmegen, The Netherlands, EU, 2004.

[PW94] H.A. Proper and Th.P. van der Weide. EVORM - A Conceptual Modelling Technique for Evolving Application Domains. *Data & Knowledge Engineering*, 12:313–359, 1994.

[PW95] H.A. Proper and Th.P. van der Weide. A General Theory for the Evolution of Application Models. *IEEE Transactions on Knowledge and Data Engineering*, 7(6):984–996, December 1995.

[SFG+00] J.J. Sarbo, J.I. Farkas, F.A. Grootjen, P. van Bommel, and Th.P. van der Weide. Meaning Extraction from a Peircean Perspective. *International Journal of Computing Anticipatory Systems*, 6:209–227, 2000.

[VHP04] G.E. Veldhuijzen van Zanten, S.J.B.A. Hoppenbrouwers, and H.A. Proper. System Development as a Rational Communicative Process. *Journal of Systemics, Cybernetics and Informatics*, 2(4), 2004.
http://www.iiisci.org/Journal/sci/pdfs/P492036.pdf

[WBW00] B.C.M. Wondergem, P. van Bommel, and Th.P. van der Weide. Matching Index Expressions for Information Retrieval. *Information Retrieval Journal*, 2(4), 2000. To appear.

[WBW01] B.C.M. Wondergem, P. van Bommel, and Th.P. van der Weide. Combining Boolean Logic and Linguistic Structure. *Information & Software Technology*, (43):53–59, 2001.

Author Index

B

Berger, F.C., 9

Bommel, P. van, 9

Bonsangue, M., 9

Bronts, G.H.W.M., 9

Brouwer, S.J., 9

Bruza, P.D., 9

Buuren, R. van, 9

C

Campbell, L.J., 9

Creasy, P.N., 9

F

Falkenberg, E.D., 7

Farkas, J.I., 9

Frederiks, P.J.M., 9

G

Grootjen, F.A., 9

H

Halpin, T.A., 9

Hesse, W., 7

Hofstede, A.H.M. ter, 9

Hoppenbrouwers, J.J.A.C., 9

Hoppenbrouwers, S.J.B.A., 7, 9

J

Jonkers, H., 9

L

Lankhorst, M.M, 9

Lindgreen, P., 7

M

Martens, C.L.J., 9

N

Nilsson, B.E., 7

O

Oei, J.L.H., 7

P

Papazoglou, M.P., 9

Proper, H.A., 9

Proper, H.A. (Erik), 9

R

Rolland, C., 7

S

Sarbo, J.J., 9

Stamper, R.K. and, 7

T

Torre, L. van der, 9

V

Veldhuijzen van Zanten, G.E., 9

Verrijn-Stuart, A.A., 7

Vos, B. van der, 9

Voss, K., 7

W

Weide, Th.P. van der, 9

Wondergem, B.C.M., 9

Y

Yang, J., 9

Dictionary

Activity participation – A **system link** between a **system activity** and one of its **actor**.

Actor – A **system element** that is conceived of as having some involvement in a **system activity**. This involvement is a special kind of **system link**, referred to as an **activity participation**.

Conception – That what results, in the mind of a **viewer**, when they interpret a **perception** of a **domain**.

Concept – Any **element** from a **conception** that is not a **links**.

Data – Any representation in some language. Data is therefore simply a collection of symbols that may, or may not, have some meaning to some **actor**.

Definition – The description of the requirements that should be met by both the desired information system as well as the documents documenting this information system. In literature this is also referred to as requirements engineering.

With regards to the information system, the resulting descriptions should identify: *what* it should do, *how well* it should do this, and *why* it should do so. With regards to the documentation of the information system, the descriptions should identify *what* should be documented, *how well* it should be documented, and *why/what-for* these documents are needed.

Deployment – The processes of delevering/implementating an information system to/in its usage context. The **design** of an information system is not enough to arrive at an operational system. It needs to be implemented-in/delivered-to a usage context.

Design – The description of the design of an information system. These descriptions should identify *how* an information system will meet the requirements set out in its definition. The resulting design may (depending on the design goals) range from high-level designs to the detailed level of programming statements or specific worker tasks.

Domain – Any 'part' or 'aspect' of the **universe** a **viewer** may have an **interest** in.

Element – The elementary parts of a **viewer**'s **conception**.

Environment – The environment of a **domain** is that part of a **viewer**'s **conception** of a **universe**, which has a direct **link** to the **domain**.

Human actors – An **actor** which is a single human being, or essentially a set of human-beings, such as a team.

Interest – The specific reason(s) why a **viewer** observes a **domain**.

In the case of a **system**, this this is usually a confluence of the **systemic properties** of **interest** to the **system viewer** and the aspects of the **system** that are considered relevant (by the **system viewer** to these **systemic properties**).

Knowledge – A relatively stable, and usually mostly consistent, set of **conceptions** posessed by a single (possibly composed) **actor**.

In more popular terms: "an actor's picture of the world".

Link – Any **element** from a **conception** that relates two **concepts**.

Maintenance – An information system which is operational in its usage context, does not remain operational by itself. Both technical and non-technical elements of the system need active maintenance to keep the information system operational as is.

Message – **Data** that is transmitted from one **actor** (the sender) to another **actor** (the receiver).

A message may actually be 'routed' via several **actors** before reaching its actual receiver. For example, when **human actor** exchange messages, they usually need to make use of some other **actor** playing the role of a medium (for example, vibrations in the air, or an e-mail system).

Model – A purposely abstracted **domain** (possibly in conjunction with its **environment**) of some 'part' or 'aspect' of the **universe** a **viewer** may have an **interest** in.

For practical reasons, a **model** will typically be consistent and unambiguous with regards to some underlying semantical domain, such as logic.

Perception – That what results, in the mind of a **viewer**, when they observe a **domain** with their senses, and forms a specific pattern of visual, auditory or other sensations in their minds.

System activity – A **system concept** that is conceived of as changing parts of the **universe**.

System concept – Any element from a **system** that is a **concept**.

System domain – A **domain** that is conceived to be a **system**, by some **viewer**, by the distinction from its **environment**, by its coherence, and because of its **systemic property**.

System element – Any element from a **system**.

System link – Any element from a **system** that is a **link**.

System viewer – A **viewer** of a **system domain**.

Systemic property – A meaningful relationship that exists between the **domain** of elements considered as a whole, the **system domain** and its **environment**.

System – A special **model** of a **system domain**, whereby all the things contained in that model are transitively coherent, i.e. all of them are directly or indirectly related to each other and form a coherent whole.

A system is conceived as having assigned to it, as a whole, a specific characterisation (a non-empty set of **systemic properties**) which, in general, cannot be attributed exclusively to any of its components.

Universe – The 'world' under consideration.

Viewer – An **actor** perceiving and conceiving (part of) a **domain**.

Subject Index

The following conventions are used in this index:

- A page where a concept is defined: *35*.
- A page where a concept is discussed or mentioned: 35.
- The page in the dictionary where a concept is defined: **35**.

A

activity participation, **31**, 31

actor, 7, **31**, 31, 32

architecting, *8*

C

communication, *7*

concept, **31**, 31, 32

conception, 7, **31**, 31

D

data, *7*, **31**, 32

definition, *8*, 8, 21, **31**

deployment, *8*, 8, **31**

design, *8*, 8, 21, **31**, 31

domain, **31**, 31, 32

domain modeling, *8*

E

element, **31**, 31

environment, **31**, 32

H

human actor, 7, **31**, 32

I

information, *7*

interest, **31**, 31, 32

K

knowledge, *7*, 7, **31**

L

link, **31**, 31, 32

M

maintenance, *8*, 8, **32**

message, 7, **32**

model, **32**, 32

P

perception, 31, **32**

S

system, 31, **32**, 32

system activity, 31, **32**

system concept, **32**, 32

system domain, **32**, 32

system element, 31, **32**

system link, 31, **32**

system viewer, 31, **32**

systemic property, 31, **32**, 32

U

universe, 31, **32**, 32

V

viewer, 31, **32**, 32

The DaVinci Lecture Notes Series:

The DaVinci series of lecture notes is concerned with *The Art & Craft of Information Systems Engineering*. On the one hand, this series of lecture notes takes a fundamental view (*craft*) on the field information systems engineering. At the same time, it does so with an open eye to practical experiences (the *art*) gained from information system engineering in industry.

Main contributors:

P. (Patrick) van Bommel

S.J.B.A. (Stijn) Hoppenbrouwers

H.A. (Erik) Proper

Th.P. (Theo) van der Weide