

CAMPUS
DE NAYER

Empowered by *Lessius*

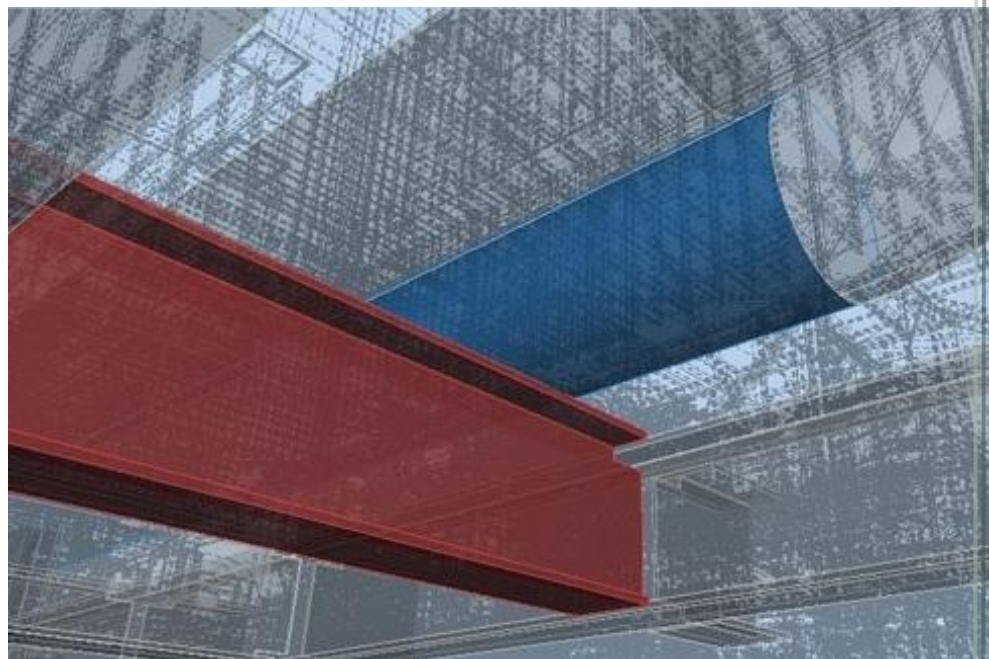
iNFRANEA
3D-ENGINEERING & VISUALISATION

Chaim Barbier
Evert Vanderheyden

Master Industriële Wetenschappen - Bouwkunde

MASTERPROEF

STUDIE NAAR EEN OPTIMAAL BEHEER VAN
GEOMETRISCHE RAAKVLAKKEN



Academiejaar
2010-2011

Interne Promotor
ing. Marc Meeus

Externe Promotor
ing. Peter Imbrechts

© Alle rechten voorbehouden. Behoudens uitzonderingen door de wet bepaald, mag zonder schriftelijke toestemming van de rechthebbenden niets uit dit document worden vermenigvuldigd en/of openbaar worden gemaakt door middel van druk, fotokopie, digitale reproductie of anderszins.

STUDIE NAAR EEN OPTIMAAL BEHEER VAN GEOMETRISCHE RAAKVLAKKEN

MASTERPROEF

Juni 2011

Katholieke Hogeschool Mechelen
Campus De Nayer



Ingenieurs- en visualisatiebureau
iNFRANEA



VOORWOORD

Beste lezer,

De afstudeerscriptie die voor u ligt, werd geschreven in het kader van de opleiding Industriële Wetenschappen Bouwkunde aan de Lessius Hogeschool Mechelen - campus De Nayer. Het hele traject dat doorlopen is om te komen tot de onderzoeksresultaten die beschreven zijn in deze tekst, werd begeleid door onze beide promotoren: ing. Peter Imbrechts, project manager bij het ingenieursbureau iNFRANEA, en ing. Marc Meeus, docent aan de Lessius Hogeschool Mechelen.

Wij wensen hier iedereen te bedanken die ons geholpen heeft bij dit onderzoek. In het bijzonder onze beide promotoren, en de mensen van de verschillende bedrijven die we mochten interviewen voor ons marktonderzoek. Ook naar de volgende personen gaat onze dank uit: ing. Wouter Bulens en ing. Michel Dekker voor het verlenen van opbouwende commentaren op het onderzoek en het nalezen van de uiteindelijke tekst, ing. Joris Laureys voor zijn deskundige input omtrent het praktisch project, en ing. Johan Kuppens die ons als managing director van iNFRANEA de gelegenheid heeft geboden dit onderwerp uit te diepen en er een artikel over te schrijven. Wij willen ook de andere mensen niet vergeten die de definitieve eindwerktekst hebben nagelezen en becommentarieerd, met name ing. Angus Noakes, Herta Van den Eynde en Walter Vanderheyden.

Wij wensen de lezer alvast een aangename en leerrijke tijd bij het doornemen van onze scriptie. Hopelijk biedt zij nieuwe en verhelderende inzichten.

Veel leesplezier!

Chaïm Barbier
Evert Vanderheyden

ABSTRACT

Grootschalige bouw- en infrastructuurprojecten zijn het resultaat van een intense en complexe samenwerking tussen verschillende partijen. De zgn. PPS- en DBFM-projecten zorgen er bovendien voor dat een groot deel van de projectrisico's bij de opdrachtnemers terecht komt. Om deze evolutie op te vangen, ervaren steeds meer opdrachtnemers de noodzaak hun huidige projectmanagement bij te sturen. Vertrekkend vanuit de principes van 'Systems Engineering' wordt het mogelijk niet alleen het volledige beheerproces, maar ook het huidige raakvlakkenbeheer te optimaliseren.

Het ultieme doel van deze masterproef bestaat dan ook uit het ontwikkelen van een systematiek om geometrische raakvlakken efficiënt te beheren. Deze geometrische raakvlakken ontstaan wanneer fysieke objecten met elkaar interfereren. Slecht raakvlakkenbeheer leidt vaak tot grote faalkosten, het oplopen van een achterstand in de planning en/of een afname van de opgeleverde kwaliteit.

De problematiek is in kaart gebracht door middel van een marktonderzoek bij verschillende partijen uit de bouwsector. De resultaten van deze interviews vormen de basis voor het vervolg van de studie die uitmondt in de ontwikkeling van een efficiënte en concrete beheermethode.

Trefwoorden:

- Systems Engineering
- Building Information Model (BIM)
- Clash Detection software
- Relatics
- N²-chart

ABSTRACT

Large-scale building and infrastructure projects are the result of intense and complex collaborations between various parties. Due to the so-called PPS- and DBFM-projects, the major part of the risk is placed with the contractors. In answer to this development, contractors feel compelled to review their project management methodology. Using 'Systems Engineering' principles allows them to optimize the entire management process, including the management of the interfaces.

The purpose of this thesis is to develop a system to manage geometric interfaces efficiently. These geometric interfaces occur when physical objects interact with each other. Inefficient management often results in high failure costs, an accumulation of the backlog, and/or a decrease in delivered quality.

To explore the problem, various parties in the construction industry were surveyed. The outcome of these interviews was used as the starting point of this study, which results in the development of an efficient and practical method to manage interfaces.

Keywords

- Systems Engineering
- Building Information Model (BIM)
- Clash Detection software
- Relatics
- N²-chart

SUMMARY –**A STUDY INTO THE OPTIMAL MANAGEMENT OF GEOMETRIC INTERFACES**

Large construction projects are the result of an intense and complex cooperation between architects, contractors and engineers. New projects require ever increasing levels of expertise, existing contracts need to be reevaluate and project deadlines keep getting tighter. Design, Build, Finance and Maintain (DBFM) contracts and Public-Private Corporation (Publiek Private Samenwerking - PPS) projects put a large emphasis of the responsibility with the contractor.

In view of these evolutions, there is a call to rethink the current way of managing projects and make project management more explicit. One way of accomplishing this is using 'Systems Engineering' principles, a project management methodology that increases the odds of completing projects successfully.

The new DBFM contracts offer a number of advantages to the contractor. First of all, the costs are spread over a longer period of time, which simplifies financing and therefore the realization of large projects. Secondly, because the contractor is also responsible for maintaining the construction, he will be more inclined to examine the entire life span of a project instead of focusing only on the construction phase.

Systems Engineering considers the whole environment as a single system: a collection of interacting components, which may themselves be subsystems. For practical purposes, however, most of the time a system will be limited to a specific part the environment that is of interest to the observer. Within the defined system, several different decompositions can be made. E.g., when all subsystems are listed in a tree, a System Breakdown Structure (SBS) is formed. It provides a convenient way to overview the different components in a system.

When decomposing systems it is important to not only pay attention to the components and subsystems, but also to the transverse cohesion of the entire project. By listing all the components in a design, it becomes easier to systematically determine which objects interface with each other. When

interfaces are detected during the early stages of a project, it becomes much easier to manage them properly.

A system will always have interfaces with its surroundings. Added to these, interfaces between systems and objects are created in the decomposition process. Because they are often overlooked, explicit management of these interfaces is crucial to the success of a project, as it will prevent failure costs, delays and a decrease in the overall quality of the project as a whole. Yet explicitly naming each and every interface and attaching requirements to them should be avoided as well, as it would cause the system to quickly become overburdened with information about mundane clashes that don't require explicit management. Several major construction companies attest that interfaces are usually defined only when they effectively pose a non-negligible risk to the project.

In most projects, plans are drawn in two dimensions only with minimal cross references. Especially in complex projects, this may give rise to interpretation errors and certain design errors, such as low headroom clearances that may go undetected.

Designs modelled in 3D, on the contrary, are understood more intuitively, even to those without a background in construction. They can help enlarge the social support for a project, and make it easier to detect and visualize clashes between construction elements. In addition, when all designs are made in 3D, a Building Information Model (BIM) can be created, which not only integrates planning and cost management, but clash detection options as well.

In theory, a BIM not only centralizes all information on a project in one location (thus creating a *Single Point of Information* or *SPI*), it also integrates all the data in a single model. However, keeping such a BIM up-to-date is far from evident, even if only speaking purely logistically. On top of that, compatibility of BIM software is still an issue: partial loss of data and model intelligence is commonplace. In practice, periodically joining the various model parts to check the interfaces would already be a huge improvement.

A new software file format, called IFC, has been introduced to tackle the compatibility issue. This format can be seen as a step towards standardizing how model information should be organized in model files. Most clash detection software support this new file format, and already a

number of programs like Solibri Model Checker, Autodesk Navisworks and Bentley Navigator are designed specifically with this new format in mind. Even though most clash detection software supports importing model files created in file formats from other producers, compatibility remains an issue. On import, the original model tends to get stripped of its intelligence and reduced to the most basic mesh form. This may introduce imperfections in the model, create new clashes between model elements or cause existing clashes to be overlooked during clash detection.

Using a Geographical Information System as a graphic Single Point of Information can facilitate project management. In addition to the advantages offered by other SPI products, a GIS enables the visual representation of data as well as the relations between them. A GIS could therefore be used as a form of BIM. The fact that information in a GIS system can easily be made accessible to all team members is a major advantage. With advanced GIS software, it may even be possible to review design interfaces without the aid of CAD software.

However, at the time of writing, GIS software has not yet developed enough of an edge over specialised CAD software to play a significant role in the management of (geometrical) interfaces. A major drawback is that most GIS systems are not capable of managing sufficient levels of detail. This is problematic because the three dimensional images shown are not yet 'true' 3D, but, e.g., surfaces with a height attached to them. Furthermore, linking CAD- and/or database data to Geographical Information Systems is far from evident.

In spite of its current shortcomings, GIS may prove to be a valid alternative for conventional BIM software in the future.

It goes without saying that software can only support the management of interfaces. The management *processes* are more important than the applied software. Without the foundation of a well-structured process, projects simply cannot be managed effectively. Software may *assist* in detecting design interfaces or manage data in a database, but there must be a methodical process behind it. This Master's paper proposes a process for managing interfaces in the form of a flowchart. The database environment that plays an important supporting role in this process, is also explained.

Systems Engineering methodology often uses N^2 -charts to detect interfaces between elements. These charts are matrices with dimensions $N \times N$, where N represents the number of objects that may have interfaces on the main diagonal. The other cells are used to mark the interfaces. An empty cell means that there is no (or a negligible) interface between the corresponding elements. The following figure shows an example of an N^2 -chart.

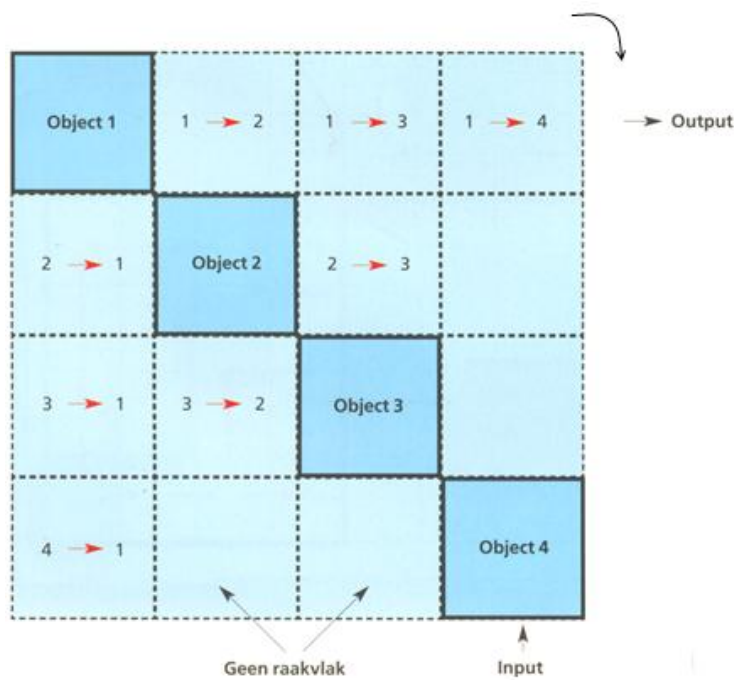


Figure 1: N^2 -chart [4]

The main advantage of a N^2 -chart is that it allows a systematic approach to discovering which objects in a system have interfaces. An 'interface manager' can then be assigned to follow up on the interfaces and assume the responsibility for them.

Once the interfaces have been located, it is possible to determine the 'critical objects' and which objects are 'leading' and which are 'intersecting'. An intersecting object must adjust to design iterations of the leading object in an interface. The two types of objects are distinguished mainly to reach clear and unequivocal agreements.

Creating N²-charts is highly labour-intensive, and therefore will be mostly used in cases of doubt or conflict. In most cases, determining which objects are 'leading' is self-evident. When it is not, the choice will need to be made based on experience.

When design interfaces have been detected, it is crucial that they are managed throughout the life-cycle of a project. This can be accomplished by entering the interfaces in a database, where they can be linked to the actions required to construct and manage them.

3D geometry and supporting software are of invaluable assistance when managing interfaces, but it is important to realize that they are not in and of themselves the solution to the problem. Indeed, the use of software only makes sense as an integrated part of project management as a whole.

Although there is room for improvement at various levels, many problems related to interfaces can be avoided through open communication processes within the project.

Both the problems highlighted in this thesis, and the proposed solutions aim to contribute towards a structured and transparent management of interfaces.

INHOUDSOPGAVE

	Hoofdstuk 1: Situering	1
1.1	Aanleiding en situering	1
1.2	Probleem- en doelstelling	2
1.3	Onderzoeksvraag	3
1.4	Structuur van de tekst	3
	Hoofdstuk 2: Literatuurstudie	5
2.1	DBFM	5
2.2	Systems Engineering	8
2.3	Raakvlakken	15
2.4	Overdracht van informatie	19
2.4.1	Databeheersystemen	20
2.4.2	Processen	26
2.5	Virtueel bouwen	28
2.6	Building Information Model (BIM)	30
2.7	Marktonderzoek	31
2.8	Besluit	36
	Hoofdstuk 3: 3D-modellering	39
3.1	3D-modellering en raakvlakkenbeheer	39
3.2	Bouw Informatiemodel	41
3.2.1	IFC	43
3.2.2	BIM-software	44
3.2.3	Samenvatting	61
3.3	Geografische Informatiesystemen	63
3.3.1	Werking van een GIS	64
3.3.2	Koppeling met CAD-tekeningen en databankgegevens	66
3.3.3	GIS-producenten	70
3.3.4	Samenvatting	71
3.4	Besluit	72
	Hoofdstuk 4: Raakvlakkenbeheer	73
4.1	N ² -chart	73
4.1.1	Bepaling van de kritische objecten	77
4.2	Proces voor het beheer van geometrische raakvlakken	82

4.2.1	Systems Engineering – Breakdown Structures	83
4.2.2	Raakvlakdetectie	85
4.2.3	Raakvlakkenbeheer	87
4.2.4	Opvolging & verificatie	89
4.2.5	Samengevat	92
4.3	Implementatie van het proces in Relatics	95
4.4	Besluit	113
Hoofdstuk 5: Praktisch project		115
5.1	Het project	115
5.2	Case 1 – Gewapende grondkerende wanden	118
5.3	Case 2 - Rioleringsdoorsteek	120
5.3.1	Autodesk Navisworks	123
5.3.2	Projectwise Navigator	125
5.4	Case 3 - Randbalk	128
5.5	Case 4 - Landhoofd	133
5.6	Besluit	136
Hoofdstuk 6: Besluit		137
6.1	Conclusies	137
6.1.1	Verkennd marktonderzoek	138
6.1.2	Beheersystematiek	138
6.1.3	Softwareondersteuning voor raakvlakkenbeheer	139
6.2	Aanbevelingen voor verder onderzoek	139
Bibliografie		137

INDEX

AGIV: Agentschap voor Geografische Informatie Vlaanderen	64
BAFO: Best And Final Offer	35
BCF: open BIM Collaboration Format	48
BIM: Building Information Model(ing)	30
CAD: Computer Aided Design	21
DBFM: Design, Build, Finance & Maintain	5
DEM: Digital Elevation Model	65
DGN: DesiGN - bestandsformaat voor o.m. MicroStation (Bentley)	46
DTM: Digital Terrain Model	65
DWG: DraWinG - bestandsformaat voor verschillende CAD pakketten	46
FBS: Functional Breakdown Structure	12
GWV: Grond-, Weg- en Waterbouw	14
HTML: HyperText Markup Language	24
IAI: International Alliance for Interoperability	43
IFC: Industry Foundation Classes	43
INCOSE: International Council on Systems Engineering	8
LBS: Location Breakdown Structure	13
MER: Milieu Effecten Rapport	33
OBS: Organizationel Breakdown Structure	13
PPS: Publiek Private Samenwerking	6
PvE: Programma van Eisen	10
RBS: Requirements Breakdown Structure	12
SBS: System Breakdown Structure	12
SE: Systems Engineering	8
SMART: Specifiek, Meetbaar, Acceptabel, Realistisch & Tijdsgebonden	12
SPI: Single Point of Informaion	30
SQL: Structured Query Language	69
WBS: Work Breakdown Structure	13
XML: eXtensible Markup Language	24

FIGURENLIJST

Figuur 1: Decompositie van het systeem [4]	9
Figuur 2: Aspecten doorheen verschillende disciplines	10
Figuur 3: Iteratief specificatieproces [4]	11
Figuur 4: Specificatieproces en oplossingsruimte [4]	13
Figuur 5: V-model [4]	14
Figuur 6: Interdisciplinair raakvlak	17
Figuur 7: Flowchart voor raakvlakbeheer opgesteld door BAM infra [6]	18
Figuur 8: Praktische vs. theoretische informatieoverdracht [7]	20
Figuur 9: Semantische structuur Relatics [9]	22
Figuur 10: Relatics-interface [10]	23
Figuur 11: Status van items in Relatics [10]	25
Figuur 12: Projectduur en taakverdeling (traditioneel)	27
Figuur 13: Projectduur en taakverdeling (heden)	27
Figuur 14: Baselining [10]	29
Figuur 15: Informatieoverdracht: traditionele werkwijze vs. SPI [12]	30
Figuur 16: Werkingsprincipe van een BIM	42
Figuur 17: De interface van Solibri	47
Figuur 18: De interface van Navisworks [18]	50
Figuur 19: De interface van Navigator [21]	55
Figuur 20: 3D-solids in Navisworks	58
Figuur 21: Geneste kolom en clash detection	60
Figuur 22: Lagenprincipe van een GIS-omgeving [24]	65
Figuur 23: N ² -chart [4]	74
Figuur 24: N ² -chart met aanvullende informatie bij de pijlen [25]	75
Figuur 25: N ² -chart met feedbacklus	76
Figuur 26: Voorbeeld van een N ² -chart	78
Figuur 27: Voorbeeld van een N ² -chart met controlelussen	80
Figuur 28: Raakvlakkenbeheer - start van het proces	84

Figuur 29: Raakvlakkenbeheer - detectie van raakvlakken aan de hand van een N ² -chart	85
Figuur 30: Raakvlakkenbeheer - implementatie in de databankomgeving	88
Figuur 31: Raakvlakkenbeheer tijdens de uitvoering	90
Figuur 32: Raakvlakkenbeheer - opvolging	91
Figuur 33: Raakvlakkenbeheer - verificatie	91
Figuur 34: Raakvlakkenbeheer - flowchart	93
Figuur 35: Screenshot van de Relaticsomgeving voor raakvlakkenbeheer	95
Figuur 36: Relaties tussen 'Object' en 'Activiteit'	96
Figuur 37: Relaties vanuit 'Object'	97
Figuur 38: Modellering van een geometrisch raakvlak in Relatics	98
Figuur 39: Leidend en rakend object in Relatics	98
Figuur 40: Schets randbalk	99
Figuur 41: 'Objecten' gerealiseerd door 'activiteiten'	100
Figuur 42: Achterliggende object-eisrelatie in Relatics	102
Figuur 43: Objectenboom bij figuur 44	103
Figuur 44: Eisen bij objecten	103
Figuur 45: Raakvlakken, gekoppeld aan een activiteit	104
Figuur 46: Raakvlakeisen in Relatics	106
Figuur 47: Detailvenster van een raakvlak in Relatics – tabblad "Bijkomende informatie"	109
Figuur 48: Detailvenster van een raakvlak in Relatics - tabblad "Algemeen"	109
Figuur 49: Linkse venster van de raakvlakpagina - tabblad "Lijst"	110
Figuur 50: Linkse venster van de raakvlakpagina - tabblad "Filter"	110
Figuur 51: Semantische structuur voor raakvlakkenbeheer in Relatics	112
Figuur 52: Situering project Noorderlaanbruggen [28]	116
Figuur 53: De Noorderlaanbruggen in afgewerkte toestand	117
Figuur 54: Landhoofd van de Noorderlaanbruggen	118
Figuur 55: Trap met bordes	119
Figuur 56: 3D-model van de trap met de bordessen	120

Figuur 57: Situatieschets rioleringsdoorsteek	121
Figuur 58: 3D-model van de rioleringsdoorsteek	122
Figuur 59: Raakvlak tussen de rioleringsdoorsteek en de trambedding	122
Figuur 60: Instellingen clash detection test in Navisworks	124
Figuur 61: Resultaat van de clash detection test in Navisworks	125
Figuur 62: Clash detection test rapportage in HTML-formaat	126
Figuur 63: Instellingen voor de clash detection job in Navigator	127
Figuur 64: Resultaat van de clash detection test in Navigator	128
Figuur 65: Foto randbalk	129
Figuur 66: Plan randbalk	130
Figuur 67: N ² -chart voor de randbalk	132
Figuur 68: Objectenboom in Relatics	134
Figuur 69: Detailscherm object "Keermuur toegangshelling"	134
Figuur 70: Detailscherm raakvlak "Landhoofd - Keermuur toegangshelling"	135

HOOFDSTUK

1

Situering

In dit hoofdstuk wordt het onderzoekskader voor deze masterproef aangereikt. Naast het beschrijven van de aanleiding tot de studie, wordt ook vastgelegd wat de probleem- en doelstelling zijn. Dit alles leidt tot de formulering van één centrale onderzoeksvraag.

1.1 Aanleiding en situering

Deze masterproef situeert zich voornamelijk in de sector van de infrastructuurprojecten. De Vlaamse bouwsector wordt recentelijk geconfronteerd met het feit dat de overheid bijna alle grootschalige infrastructuurprojecten voor wegen, spoorwegen, tramlijnen ... aanbesteedt als PPS-projecten (Publiek Private Samenwerking). Een groot deel van deze projecten zijn gebaseerd op volledig nieuwe DBFM-contractvormen (Design, Build, Finance and Maintain). Hierbij beperkt de overheid zich tot het opstellen en controleren van een 'Programma Van Eisen' voor alle fasen van de projectlevenscyclus.

Dit veroorzaakt een grondige verschuiving van taken en verantwoordelijkheden, zowel in de relatie opdrachtgever - opdrachtnemer, als binnen de groeperingen van aannemers en hun raadgevende ingenieursbureaus en architecten. Omdat de overheid nu een groot deel van de projectrisico's bij de aannemers(combinaties) legt, worden zij verplicht hun projectbeheersing bij te sturen en meer expliciet te maken. Daarbij kan gebruik gemaakt worden van de principes van 'Systems Engineering'. Deze methode geeft immers goede garanties voor het succesvolle verloop van projecten met een groot maatschappelijk belang. Voor de zgn. 'Missing Link'-projecten, waarmee de Vlaamse Overheid het wegennet wil optimaliseren, is de opdrachtnemer vanuit het bestek zelfs verplicht tot het toepassen van de Systems Engineering principes.

In de praktijk blijkt echter dat de verhoogde risico's voor de opdrachtnemer pijnlijke confrontaties kunnen veroorzaken tijdens de uitvoering van de projecten. Een slecht beheer van de projectrisico's, die vaak samenhangen met raakvlakken, leidt dan ook tot grote faalkosten en het oplopen van achterstand in de planning.

1.2 **Probleem- en doelstelling**

De verhoogde risico's die samenhangen met de nieuwe contractvormen dwingen opdrachtnemers na te denken over knelpunten in hun processen. Eén van deze knelpunten wordt gevormd door een gebrekkig raakvlakkenbeheer. Wanneer raakvlakken bij het ontwerp onvoldoende aandacht krijgen, zullen er onvermijdelijk uit het oog verloren worden. Vaak is echter niet eenduidig gedefinieerd wanneer van een raakvlak kan gesproken worden. De afgelopen jaren is een wollige discussie gevoerd over het begrip 'raakvlak', maar in de praktijk laat het beheer ervan vaak te wensen over. Zoals ook blijkt uit onderhavige studie, kan het begrip 'raakvlak' heel breed geïnterpreteerd worden. Deze masterproef probeert aan te geven wanneer kan en moet gesproken worden over een 'raakvlak'. Voor het beheer van raakvlakken beperkt de studie zich tot geometrische raakvlakken, die ontstaan door de interactie tussen fysieke objecten. Naast het ontbreken van een duidelijke definitie van het begrip 'raakvlak', wordt in de praktijk vaak op een weinig gestructureerde manier omgegaan met raakvlakkenbeheer. De noodzaak een systematiek te ontwikkelen voor het vastleggen en beheren van raakvlakken, is dan ook bijzonder groot.

In het algemeen kan de volgende **probleemstelling** geformuleerd worden:

Naast het feit dat het begrip 'raakvlak' niet eenduidig gedefinieerd is, blijkt dat in de praktijk ook te weinig aandacht wordt besteed aan het beheer ervan. Dit geeft onvermijdelijk aanleiding tot faalkosten, het oplopen van een achterstand op de planning en/of een mogelijke daling van de opgeleverde kwaliteit.

Zoals reeds aangegeven, is deze masterproef een studie naar een optimalisering van het huidige raakvlakkenbeheer. Het ultieme doel vertaalt zich in een systematiek om raakvlakken efficiënt te beheren, waarbij deze

raakvlakken eenduidig in kaart worden gebracht aan de hand van de geometrie.

Concreet kan de **doelstelling** als volgt ontleed worden:

Optimalisering van het raakvlakkenbeheer door middel van:

- 1) het uitvoeren van een marktonderzoek, waarbij de raakvlakkenproblematiek in kaart wordt gebracht;
- 2) het zoeken naar een systematiek om raakvlakken vast te leggen en te beheren;
- 3) het onderzoeken van de verschillende software-mogelijkheden om geometrische raakvlakcontroles mogelijk te maken.

1.3 **Onderzoeksvraag**

Uit §1.2 – **Probleem- en doelstelling** blijkt dat de kern van het onderzoek herleid kan worden tot de onderstaande **onderzoeksvraag**:

Is het mogelijk een systematiek vast te leggen waarmee het beheer van geometrische raakvlakken efficiënter kan gebeuren?

1.4 **Structuur van de tekst**

De opbouw van deze masterproef volgt de weg die het onderzoek heeft doorlopen. Eerst is een brede studie van de beschikbare literatuur uitgevoerd, om zo een algemeen beeld te kunnen vormen van de problematiek omtrent geometrische raakvlakken. Om de verderop uitgewerkte beheersystematiek te implementeren in een daarvoor geschikte databank, is een afweging gemaakt van verschillende databeheersystemen.

Gezien het praktijkgerichte karakter van dit onderwerp, is de literatuurstudie uitgebreid met een marktonderzoek. Dit marktonderzoek is gebaseerd op een aantal interviews met voornamelijk partijen uit de bouwsector. Hierbij is getracht zowel de opinie van de aannemers en de studiebureaus, als van de overheid weer te geven.

Op basis van het algemene beeld dat in hoofdstuk 2 is opgebouwd, wordt in een volgende sectie bestudeerd op welke manier 3D-geometrie kan helpen bij raakvlakmanagement. In het kader hiervan worden ook een aantal softwarepakketten kort besproken en met elkaar vergeleken.

Vermits software alleen geen soelaas kan bieden om problemen omtrent raakvlakken te vermijden, is in hoofdstuk 4 een volledige beheersystematiek uitgewerkt. Deze systematiek is praktisch geïmplementeerd in een relationeel databeheersysteem.

In het daaropvolgende hoofdstuk worden de verkregen inzichten toegepast op een praktisch project. De scriptie sluit af met een kort overzicht van de belangrijkste conclusies uit het onderzoek.

Als opmerking kan nog worden meegegeven dat de literatuurverwijzingen in deze tekst als volgt zijn aangegeven:

- Verwijzingen die bij een hele paragraaf of alinea horen, zijn als afzonderlijk nummer weergegeven op het einde ervan.
- Referenties die betrekking hebben op één welbepaalde zin worden op het einde van deze zin vermeld.
- De literatuurverwijzingen zijn in volgorde van voorkomen genummerd en opgenomen in de literatuurlijst achteraan.

HOOFDSTUK

2

Literatuurstudie

Dit hoofdstuk verkent de context waarbinnen de raakvlakkenproblematiek speelt. In de eerste plaats worden op basis van de beschikbare literatuur de recente contractuele veranderingen in de bouwsector geschetst. Vervolgens wordt dieper ingegaan op de methode van Systems Engineering en hoe deze kan helpen met de eerder vermelde veranderingen om te gaan. Daarbij is de nodige aandacht besteed aan de rol van raakvlakken binnen het concept van Systems Engineering.

Paragrafen 2.4 en 2.6 halen het probleem van informatiedegradatie aan en leggen uit hoe daarmee kan worden omgegaan, zowel procesmatig als op gebied van softwareondersteuning.

Aangezien raakvlakkenbeheer een praktisch probleem is, mag een inleidende studie zich niet beperken tot het doornemen van naslagwerken. Daarom wordt in [§2.7 - Marktonderzoek](#) een beeld geschetst van de actuele visie en problemen omtrent raakvlakkenbeheer en Systems Engineering. Deze paragraaf is tot stand gekomen op basis van interviews met een aantal voorname partijen op de Belgische en Nederlandse markt.

2.1**DBFM**

Sinds kort maakt de overheid gebruik van een nieuwe contractvorm voor aanbestedingen, de DBFM-contracten, waarbij een samenwerking tussen publieke en private partijen tot stand wordt gebracht. De opdrachtgever beperkt zich in deze werkvorm tot het specificeren van het uiteindelijk gewenste resultaat. Alle andere verantwoordelijkheden en bijbehorende risico's inzake ontwerp (Design), uitvoering (Build), financiering (Finance) en onderhoud (Maintain) liggen bij de opdrachtnemer. De opdrachtgever

legt het gewenste resultaat vast in een zogenaamde outputspecificatie, een ondubbelzinnige beschrijving van het gewenste resultaat [1]. Een outputspecificatie omschrijft niet enkel de technische aspecten van een project, maar ook juridische procedures, gewenste functionaliteiten ... De outputspecificaties maken een integraal deel uit van het DBFM-contract. [2]

Een DBFM-contract kan normaliter worden beschouwd als een specifieke vorm van een PPS-contract, waarbij PPS staat voor Publiek Private Samenwerking. Het *Vlaams kenniscentrum PPS* maakt een onderscheid tussen een contractuele en een participatieve PPS:

- Participatieve PPS: Overheid en opdrachtnemer richten een nieuwe juridische entiteit op om samen het project te realiseren.
- Contractuele PPS: Overheid en opdrachtnemer leggen hun samenwerking vast in een contract, maar behouden hun eigen identiteit. Het is onder deze vorm dat de meeste DBFM-contracten worden afgesloten. [2]

Naast een contract dat volledig volgens het DBFM-principe wordt uitgevoerd, zijn nog verscheidene andere contractvormen mogelijk. Zo kan de overheid ervoor opteren het werk aan te besteden als een Design&Build-opdracht, waarbij zowel het ontwerp als de realisatie onder de verantwoordelijkheid van de opdrachtnemer vallen. Een andere variant is een Engineering&Build-contract waarin de opdrachtnemer de realisatie en alle nodige stabiliteitsberekeningen voor zijn rekening neemt. De verantwoordelijkheid voor het ontwerp blijft echter in handen van de overheid.

De financiële vergoeding van de opdrachtnemer kan bij PPS-contract verschillende vormen aannemen. Zo kan de opdrachtgever het product gedurende een vooraf vastgestelde periode 'huren' van de opdrachtnemer. Deze laatste zal hiervoor van de opdrachtgever een beschikbaarheidsvergoeding ontvangen. Een alternatieve vorm geeft de opdrachtnemer gedurende een bepaalde tijd exploitatierechten over het gerealiseerde project (bv. tolheffing). De financiering gebeurt bij een DBFM-project door de private partij, waardoor zij wel duurder uitvalt dan bij een klassieke aanbesteding.

Aan een DBFM-contractvorm zijn een aantal belangrijke voordelen verbonden: Vooreerst wordt de kost voor de opdrachtgever gespreid over een langere periode, wat de realisatie van grote projecten vergemakkelijkt. Daarnaast zal al in de ontwerpfase van het project de volledige levenscyclus ervan onder de loep genomen worden. Dit voorkomt dat opdrachtnemers enkel gericht zijn op het drukken van de bouwkosten, door bijvoorbeeld goedkopere maar minder duurzame materialen of werkmethoden te gebruiken. Bovendien zal de opdrachtgever zich niet meer bezighouden met het volledige ontwerp van het te realiseren project, maar laat hij dit over aan de opdrachtnemers. Met het oog op de concurrentie binnen de markt zullen deze laatsten gedwongen worden een zo creatief mogelijke invulling van de klantwens naar voren te brengen. In het algemeen kan dus gesteld worden dat PPS-contracten bijdragen tot innovatie en het aanwakkeren van de concurrentieposities binnen de markt. [1]

Zoals reeds aangehaald, is één van de voordelen van deze participatieve werkvorm dat een opdrachtgever, die vaak niet vertrouwd is met de bouwtechnische aspecten van een project, geen ontwerp meer hoeft vast te leggen.

De opdrachtgever heeft immers behoefte aan het realiseren van een nieuw systeem (constructie, infrastructuur ...) en zal daarbij een marktonderzoek uitvoeren om na te gaan of er inderdaad nood is aan zijn beoogde systeem [3]. De realisatie van het systeem zelf en de bijbehorende werkzaamheden vallen echter buiten zijn primaire proces. De opdrachtgever zal dus specificeren aan welke functies zijn project moet voldoen en zal controleren of de oplossingen die worden aangedragen door de opdrachtnemer voldoen aan zijn wensen. De concrete invulling van deze specificaties wordt echter overgelaten aan de hiervoor beter gekwalificeerde opdrachtnemer. Zo brengt een PPS-project het beste van twee werelden samen. Hierbij dient te worden opgemerkt dat dergelijke PPS-contracten wel een complexere voorbereiding met zich mee brengen. [4]

2.2 **Systems Engineering**

Een recente evolutie binnen de PPS-projecten is het toepassen van Systems Engineering (SE). The International Council on Systems Engineering (INCOSE) geeft de volgende definitie:

An interdisciplinary approach [...] to enable the realization of successful systems. Systems Engineering considers both the business and technical needs of all customers with the goal of providing a quality product that meets the user needs [5].

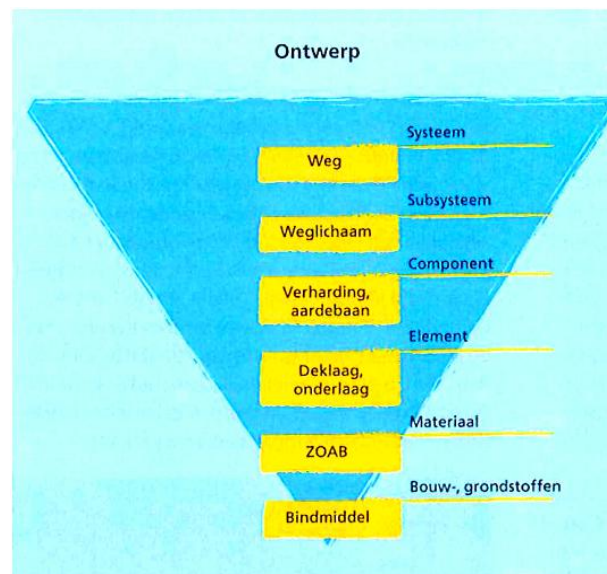
Aan de basis van Systems Engineering ligt het systeemdenken: een denkwijze waarbij de werkelijkheid wordt beschouwd als één systeem dat kan worden opgesplitst in deelsystemen. Hierbij is de opsplitsing in systemen afhankelijk van de scope van de waarnemer. Zo kan vanuit het standpunt van de overheid het globale woonbeleid als een systeem beschouwd worden, met het sociale woonbeleid als subsysteem, terwijl een OCMW het sociale woonbeleid als een systeem kan zien met de individuele verkavelingen als subsysteem. Een systeem kan dus de hele realiteit omvatten, maar een waarnemer kan ook de voor hem relevante omgeving afbakenen als een systeem. Belangrijk daarbij is dat elk systeem steeds op zichzelf kan functioneren, hoewel het altijd deel uitmaakt van een groter geheel. Deze functionele autonomie impliceert dat het systeem onafhankelijk van zijn omgeving moet kunnen beschouwd en gerealiseerd worden. Uiteraard zal het systeem altijd relaties hebben met de omgeving, maar het uitgangspunt is dat de afstemming op deze omgeving tot een minimum beperkt moet blijven. [4]

De SE-wijzer die is uitgegeven door de Koninklijke BAM-Groep geeft de volgende, praktische definitie van een systeem vanuit een bouwkundig standpunt:

Een systeem omvat alle onderdelen die fysiek achterblijven en/of proceswijzigingen die worden beschouwd als het opgeleverde werk [6].

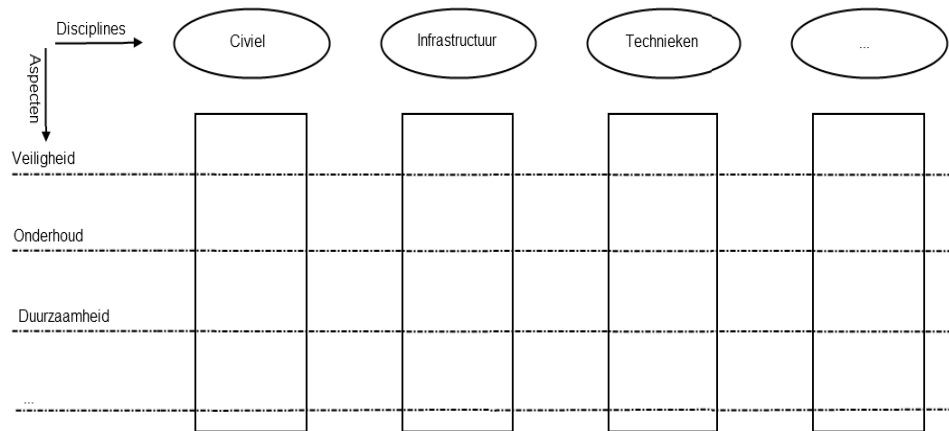
Met proceswijzigingen worden bijvoorbeeld tijdelijke constructiedelen bedoeld. Deze voorlopige elementen blijven weliswaar niet fysiek achter, maar kunnen wel opgenomen worden in het systeem.

Eens een waarnemer zijn systeem heeft afgebakend zal hij een beschrijving maken van de gewenste functies daarvan. In deze fase wordt het hele systeem nog beschouwd als een 'black box'. Het stellen van eisen aan functies en niet aan objecten heeft als belangrijk voordeel dat op deze manier het systeem zelf nog niet wordt vastgelegd. Zo bepaalt de functionele eis aan de capaciteit van een verbinding tussen de punten A en B (het aantal voertuigen dat binnen een bepaalde tijdspanne moet kunnen passeren) niet of die verbinding door een weg, een tunnel, een fly-over ... gerealiseerd moet worden. In een volgende stap zal de 'black box' geopend worden en wordt het volledige systeem opgesplitst in een aantal subsystemen. Elk van deze subsystemen is in deze fase opnieuw een 'black box' waarvan de functies worden beschreven. Dit proces gaat steeds verder tot op het gewenste detailniveau en wordt omschreven als het 'decomponeren van het systeem'. De onderstaande figuur geeft deze decompositie aanschouwelijk weer voor een concreet voorbeeld. [4]



Figuur 1: Decompositie van het systeem [4]

Bij de decompositie van systemen is het belangrijk om niet alleen aandacht te schenken aan de opsplitsing in deelsystemen en systeemelementen (objecten), maar ook aan de overdwarse samenhang van het geheel (zie figuur 2). Zo kan het aspect veiligheid over verschillende deelsystemen heen bekeken worden. Het beschouwen van de belangrijke aspecten van het systeem is meteen ook een mogelijkheid om raakvlakken over de grenzen van de subsystemen te gaan beheersen. [3]



Figuur 2: Aspecten doorheen verschillende disciplines

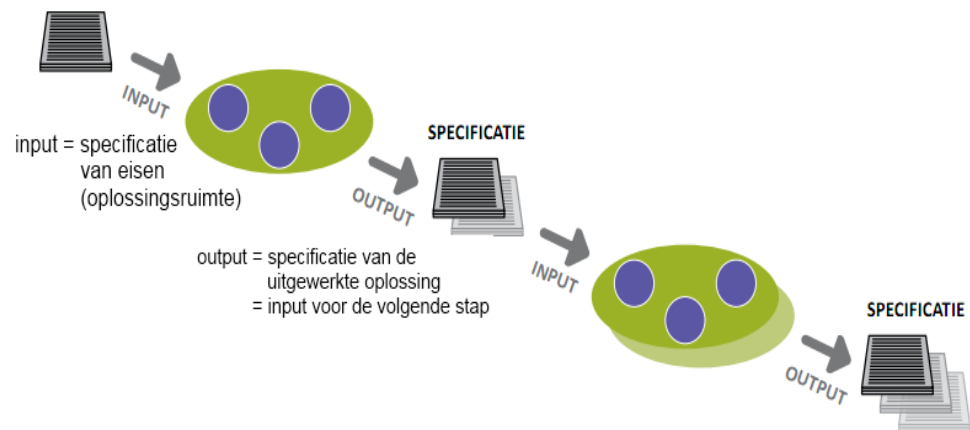
Zoals hierboven al werd aangehaald zal een klant, eens hij zijn 'system of interest' heeft afgebakend, samen met de andere stakeholders bepalen aan welke functies het systeem moet voldoen. Deze *functionele eisen* worden samen met de zogenaamde *aspecteisen* opgenomen in een Klant Eis Specificatie (Programma van Eisen). Binnen het hele proces van Systems Engineering is elke stap gericht op het vervullen van de behoeften van de klant, en staan zijn wensen steeds centraal.

Omdat de klant tevens een vrij goed beeld heeft van de raakvlakken tussen het systeem en zijn omgeving, kunnen naast de functionele eisen en de aspecteisen ook al enkele (externe) raakvlakeisen worden opgenomen in het eisenpakket. [4]

Het specificeren van de behoeften van de diverse stakeholders leidt meteen tot een afbakening van de mogelijke oplossingen om aan die behoeften te voldoen. Binnen de gedefinieerde oplossingsruimte maakt de opdrachtnemer vervolgens een gegronde keuze voor één mogelijke oplossing.

Deze ontwerpkeuze geeft op haar beurt weer aanleiding tot nieuwe afgeleide functies en dus tot nieuwe eisen. Dit iteratieve proces (zie [figuur 3](#)) wordt doorlopen tot op het gewenste detailniveau. Gedurende het hele proces van Systems Engineering is open communicatie tussen de verschillende partijen cruciaal. Daarnaast wordt er gestreefd naar zoveel mogelijk transparantie. Dit betekent dat steeds wordt bijgehouden welke

eisen, beslissingen ... gedurende het proces vastgelegd worden en om welke redenen deze gemaakt zijn. Zo kan worden aangetoond dat een oplossing voldoet aan de gestelde vraagspecificatie van de klant, en kan later nog worden achterhaald waarom bepaalde (ontwerp)keuzes gemaakt werden.



Figuur 3: Iteratief specificatieproces [4]

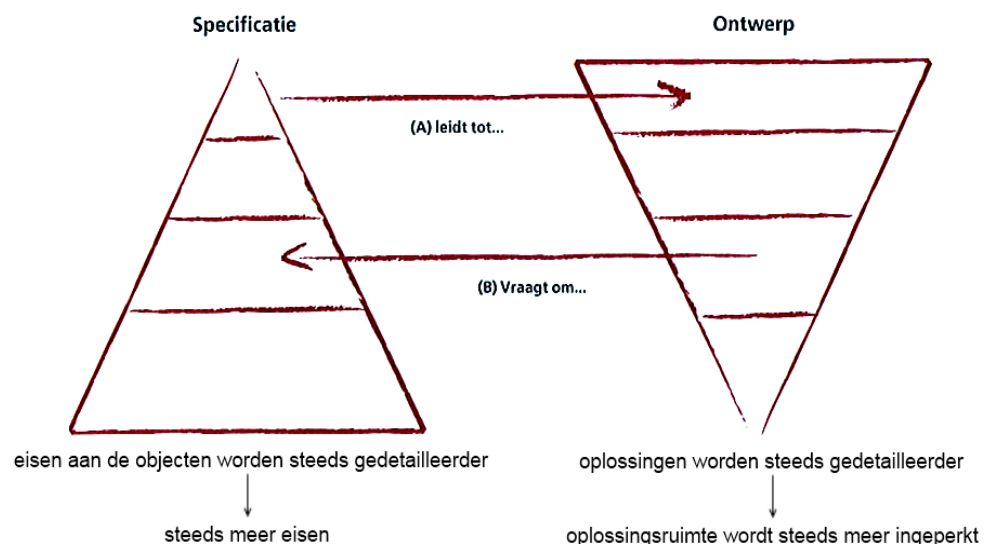
De behoefte van de opdrachtgever (klant) wordt vastgelegd in een Programma van Eisen (PvE) waarmee de opdrachtnemer aan de slag kan. In het ideale geval legt het PvE louter de behoefte van de klant vast zonder te specificeren *hoe* daaraan kan voldaan worden. De eisen moeten dus voldoende duidelijk geformuleerd zijn, maar toch genoeg ruimte laten aan de opdrachtnemer om er invulling aan te geven. Het opstellen van dergelijke eisen is allerminst evident. Het *Handboek Oplossingsvrij Specificeren* kan een houvast bieden om dit op een correcte en systematische manier te doen. [3] [4]

Binnen de methode van Systems Engineering kunnen verschillende decomposities worden doorgevoerd, hieronder volgt een korte beschrijving [6]:

1. System Breakdown Structure (SBS), ook wel objectenboom genoemd. Zoals **figuur 1** aangeeft, zal het volledige systeem worden gedecomposeerd in subsystemen, componenten, elementen ... Door de verschillende objecten van een systeem weer te geven in een boomstructuur, kan gemakkelijk een overzicht behouden worden over het hele systeem. Objecten zijn daarbij fysieke en tastbare elementen van het systeem [4]. Zo zijn vergunningen, normen, bestekteksten ... nodig om een constructie te kunnen bouwen, maar deze worden binnen de methode van SE niet benoemd als objecten. Indien nodig kunnen ook tijdelijke objecten die geen tastbaar onderdeel vormen van de uiteindelijke constructie, maar wel een belangrijk deel uitmaken van de werkzaamheden (zoals een tijdelijke omleiding), eveneens opgenomen worden in de objectenboom [6].
2. Functional Breakdown Structure (FBS) of functieboom zal de eerder gedefinieerde functies (zie hoger) vastleggen in een duidelijke structuur.
3. Requirements Breakdown Structure (RBS) of eisenboom, waarin de eisen en hun onderlinge relaties worden weergegeven. Het eisenpakket van de opdrachtgever wordt na ontvangst eerst geanalyseerd. Dit houdt in dat wordt nagegaan of alle informatie beschikbaar is om een oplossing voor de desbetreffende eisen te definiëren. Daarnaast moeten de eisen ook voldoende SMART geformuleerd zijn. SMART staat voor Specifiek, Meetbaar, Acceptabel, Realistisch en Tijdsgebonden. Het is nodig dat eisen op een dergelijke manier worden gedefinieerd om er een oplossing voor te kunnen formuleren die eenduidig, goed te verantwoorden en gemakkelijk verifieerbaar is. Zo kan een eis aan de stroefheid van de weg meetbaar en verifieerbaar gemaakt worden door de exacte waarde waaraan die stroefheid moet voldoen op te zoeken in de desbetreffende norm. Naast het SMART maken van de eisen kunnen deze ook onderverdeeld worden als objecteis, aspecteis, raakvlakeis ...

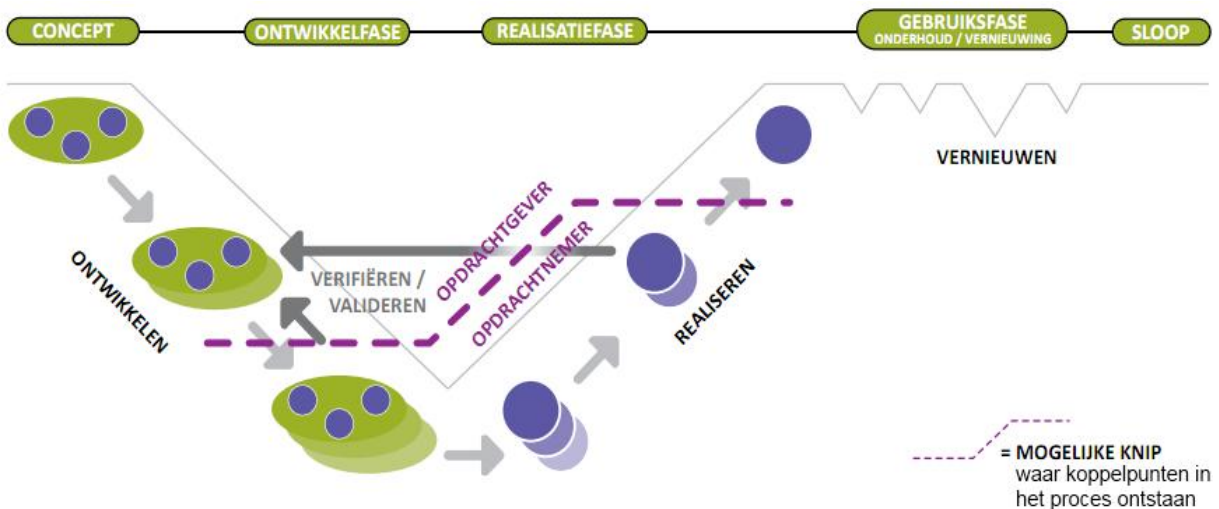
4. Work Breakdown Structure (WBS) of de activiteitenboom geeft een gestructureerd overzicht van de uit te voeren activiteiten.
5. Organization Breakdown Structure (OBS) deelt de organisatie van een project op in verschillende segmenten, zoals de grond- en wegeniswerken, kunstwerken ...
6. Location Breakdown Structure (LBS) linkt de verschillende objecten binnen het project aan hun fysieke locatie.

Na de analyse van het eisenpakket en het opstellen van de eisenboom zal worden overgegaan tot het kiezen van een gepaste oplossing. Binnen de gedefinieerde oplossingsruimte zal, op voorwaarde dat de eisen voldoende oplossingsvrij gehouden zijn, steeds ruimte zijn voor meerdere varianten. Het maken van een zo correct mogelijke keuze houdt in dat deze verschillende varianten worden geïdentificeerd om vervolgens de optimale oplossing te kiezen en verder uit te werken. Zoals al vermeld zal elke oplossing nieuwe eisen genereren. De onderstaande **figuur** geeft een grafische voorstelling van deze procedure.



Figuur 4: Specificatieproces en oplossingsruimte [4]

Door het decomponeren van het systeem zullen de deelsystemen en de bijbehorende eisen steeds gedetailleerder worden; dit wordt in de *Leidraad voor Systems Engineering binnen de GWW-sector* omschreven als 'top-down ontwikkelen'. Bij het realiseren van het uiteindelijke systeem moeten deze verschillende deelsystemen uiteraard terug samengevoegd worden tot het globale systeem; dit kan gezien worden als 'bottom-up realiseren'. Het 'top-down' ontwikkelen en 'bottom-up' realiseren kan aanschouwelijk worden voorgesteld als een V-vormig model, zoals de onderstaande **figuur** toont. Zowel in het specificatie- als in het realisatieproces moet bovendien continu gecheckt worden of aan de gedefinieerde oplossingsruimte wordt voldaan. Vaak wordt daarbij een onderscheid gemaakt tussen verificatie en validatie. Verificatie controleert of iets *juist* gerealiseerd is, valideren gebeurt meestal op het einde van het integratieproces en gaat na of het *juiste* gebouwd is [6]. [3]



Figuur 5: V-model [4]

2.3 Raakvlakken

Zoals hierboven al aangegeven werd, zal een systeem steeds raakvlakken hebben met zijn omgeving. Daarnaast zullen bij de decompositie van het systeem altijd relaties ontstaan tussen de verschillende subsystemen en objecten. Deze relaties worden benoemd als 'raakvlakken'. Hierbij kan een onderscheid gemaakt worden tussen externe en interne raakvlakken. Het expliciet beheren van raakvlakken is cruciaal omdat deze vaak uit het oog worden verloren, wat aanleiding geeft tot het ontstaan van faalkosten. [4]

Externe raakvlakken ontstaan doordat een systeem moet worden ingepast in een bestaande omgeving. Eerder is al uitgelegd dat bij de afbakening van systemen de interactie met de omgeving tot een minimum beperkt moet blijven. Dit voorkomt dat een overvloed aan externe raakvlakken ontstaat die nauwelijks nog beheersbaar is. Toch ontstaat altijd een wisselwerking met de omgeving, waarbij enerzijds het systeem zijn omgeving beïnvloedt en anderzijds de bestaande omgeving beperkingen oplegt of mogelijkheden schept voor het systeem. Zo zal de afwatering van een weg moeten aansluiten op het bestaande rioleringsstelsel. Dit rioleringsstelsel valt buiten het systeem van het wegontwerp, maar er moet wel degelijk rekening mee gehouden worden.

Net zoals moet worden vermeden dat onnodig veel externe raakvlakken ontstaan, is het aangewezen de decompositie van het systeem niet verder door te drijven dan nodig. Interne raakvlakken ontstaan immers tussen de verschillende subsystemen en objecten. Hoe meer het systeem wordt gedecomposeerd, hoe meer objecten en hoe meer onderlinge relaties tussen deze objecten tevoorschijn komen. Volgens het *Handboek Oplossingsvrij Specificeren* moet, telkens er nieuwe objecten gedefinieerd worden, nagegaan worden welke raakvlakken hierbij zijn ontstaan.

Toch is het vanuit praktisch oogpunt weinig zinvol elke relatie tussen objecten te benoemen als raakvlak en daar eisen aan te koppelen. Zo zou het hele systeem al snel overladen en log worden. **Uit marktonderzoek (zie verder) blijkt dat raakvlakken expliciet als eis worden vastgelegd op het moment dat er ook effectief een risico aan vast hangt dat beheerd moet worden.** Risico's kunnen in beeld gebracht worden door het uitvoeren van een risicoanalyse. Sommige bedrijven maken echter gebruik van generieke raakvlakken. Dit zijn raakvlakken die vaak terug komen bij

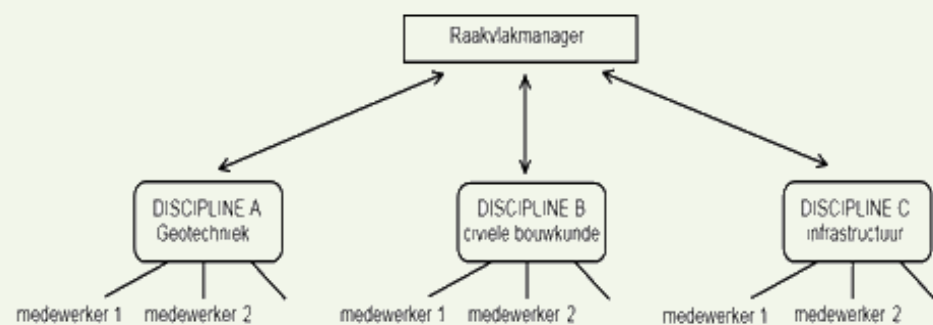
verschillende projecten en waarvan bekend is dat er een risico aan verbonden is. Een voorbeeld van een raakvlak met risico is de doorrijhoogte onder een kunstwerk; daar zal dan ook expliciet een raakvlakeis aan worden gesteld. In het *Handboek Oplossingsvrij Specificeren* wordt bovendien gewag gemaakt van een methode waarbij kritische objecten kunnen worden gedetecteerd. Hierbij wordt gebruik gemaakt van een N²-chart om na te gaan of een object veel raakvlakken heeft met andere objecten. Dit eerste object wordt benoemd als kritisch en er moet dan ook extra aandacht aan worden besteed. In [§4.1 – N²-chart](#) wordt een werkbare en gedetailleerde methode aangereikt voor het opstellen van een N²-chart.

Het decomponeren van het systeem en het correct specificeren en managen van eisen is een eerste cruciale stap voor een efficiënt raakvlakkenbeheer. Daarbij komt nog dat projecten vaak worden opgesplitst volgens de verschillende disciplines die aan bod komen binnen dat project. Zeker bij grote projecten is het immers onmogelijk dat één enkele persoon het volledige overzicht bewaart over het gehele systeem. Dit impliceert dat tussen objecten niet alleen ‘zuivere’ raakvlakken, maar ook interdisciplinaire raakvlakken ontstaan. Hierbij kan worden opgemerkt dat het beschouwen van de aspecten van het systeem (veiligheid, duurzaamheid, onderhoud ...) al een deel van die interdisciplinaire raakvlakken in kaart kan brengen. [3]

Voorbeeld 1

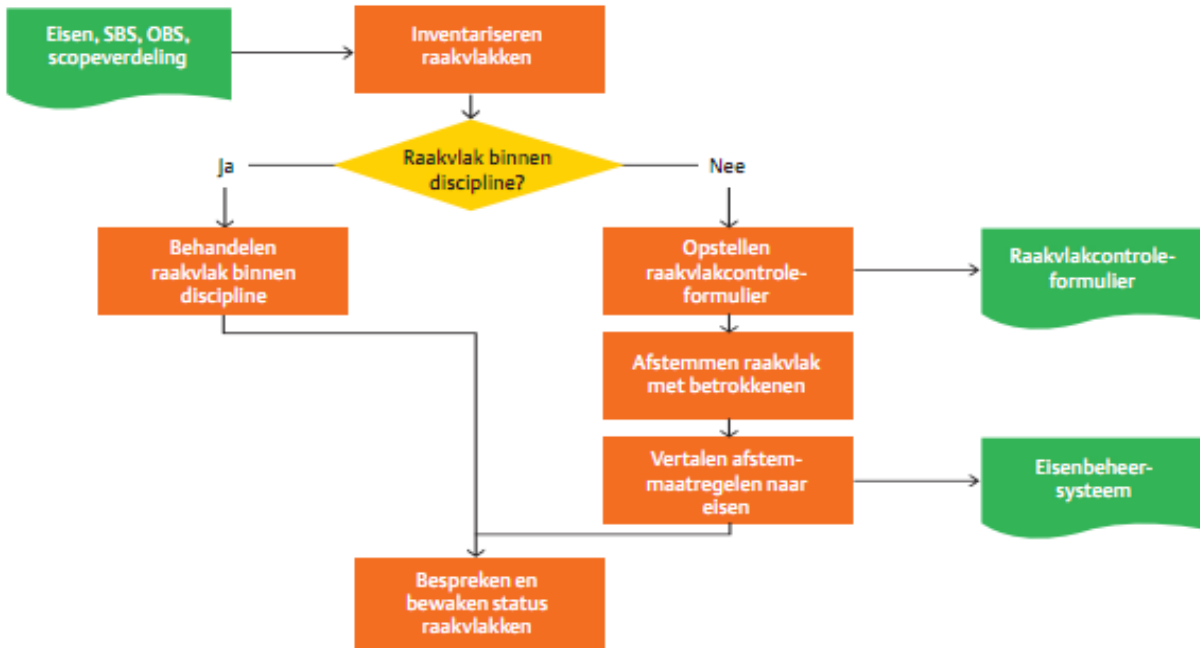
Neem als voorbeeld de ontdekking van een nieuw raakvlak dat mogelijk de ontwerpplannen van meerdere disciplines beïnvloedt. Zonder vooraf duidelijk vastgelegde processen kan het heel wat voeten in de aarde hebben om dit raakvlak met de juiste betrokkenen correct te beheren.

De onderstaande **figuur** toont een (sterk vereenvoudigd) beeld van een structuur waarmee een dergelijke situatie relatief eenvoudig beheerd kan worden. Het project is hierbij opgesplitst volgens verschillende disciplines. Als een medewerker die verantwoordelijk is voor het geotechnisch ontwerp een raakvlak ontdekt, kan hij dit voorleggen aan de verantwoordelijke van zijn discipline. Deze disciplinemanager bekijkt dan of het raakvlak binnen zijn eigen discipline beheerd kan worden. Indien het raakvlak echter consequenties heeft voor het ontwerp van andere disciplines zal hij contact opnemen met de raakvlakmanager. Deze zal ervoor zorgen dat alle relevante partijen betrokken worden bij het beheersen van het raakvlak. Het voordeel hierbij is dat de raakvlakmanager zelf geen inhoudelijke kennis hoeft te hebben van de verschillende disciplines.



Figuur 6: Interdisciplinair raakvlak

Indien een raakvlak binnen dezelfde discipline blijft, is het veel gemakkelijker om dit raakvlak te beheersen. De *SE-wijzer*, uitgegeven door BAM Infra, geeft onderstaande flowchart voor het detecteren en managen van raakvlakken [6]:



Figuur 7: Flowchart voor raakvlakbeheer opgesteld door BAM infra [6]

Om raakvlakken correct te beheren is het noodzakelijk deze te inventariseren. Eens het raakvlak benoemd is, wordt een raakvlakeigenaar aangeduid. Het is belangrijk één enkele persoon te benoemen voor het beheer van een raakvlak om te vermijden dat de verantwoordelijkheid wordt doorgeschoven naar iemand anders. Hier komt ook één van de grote voordelen van een raakvlakkenanalyse naar voren: het is een manier om expliciet vast te leggen dat er een raakvlak is, wie de betrokken partijen zijn, wat de relaties zijn tussen verschillende objecten ... [4]. Uiteraard volstaat het niet om het raakvlak te benoemen, maar moet ook worden vastgelegd met welke maatregelen dit raakvlak beheerst zal worden. De raakvlakeigenaar neemt vervolgens de taak op zich om op te volgen of deze maatregelen effectief in acht worden genomen.

In de flowchart wordt het opstellen van een raakvlakcontroleformulier aangehaald. Dit formulier is opgesteld door de Koninklijke BAM Groep en

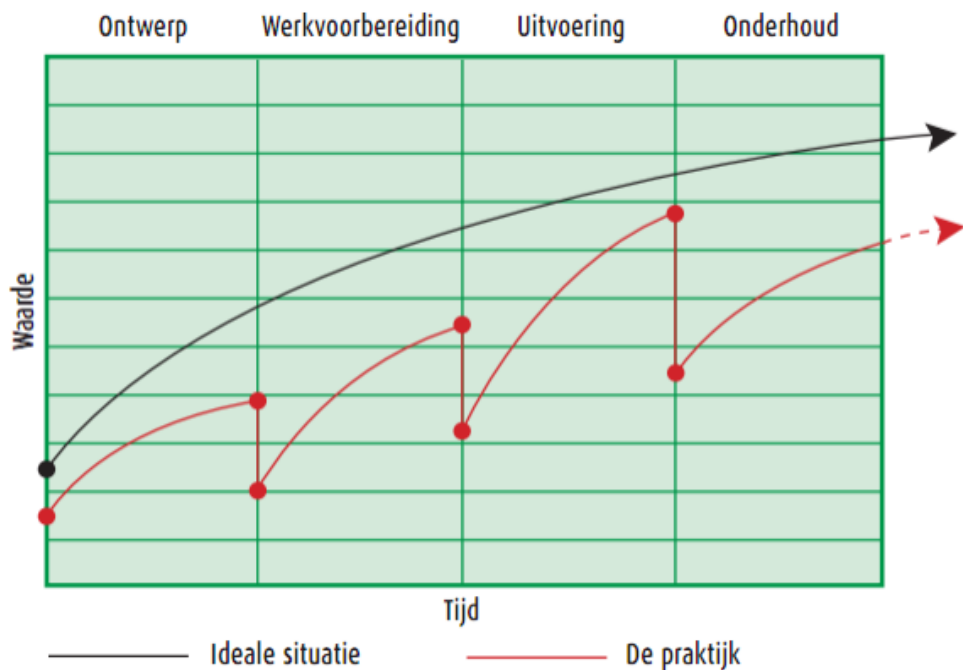
dient als hulp bij het beheersen van een raakvlak. Het is terug te vinden in de *SE-wijzer voor BAM Infra*. [6]

2.4 Overdracht van informatie

Bij het managen van raakvlakken is het tevens cruciaal dat de verschillende partijen steeds met de juiste informatie werken. Enerzijds houdt dit in dat de laatste versie van een ontwerp, uitvoeringsprocedure ... ter beschikking moet staan van de betrokken personen. Daarnaast is het ook belangrijk dat de algemene overdracht van informatie op een efficiënte manier verloopt. Informatieoverdracht vindt niet alleen plaats tussen de verschillende disciplines en fasen binnen het project, maar ook tussen opdrachtgever en opdrachtnemer. Gedurende het hele proces is open communicatie tussen de betrokken partijen dus onontbeerlijk. De *Leidraad voor SE binnen de GWW-sector* benoemt de overdrachtmomenten tijdens het proces als 'koppelpunten'. **Figuur 5** geeft de scheiding van de informatieoverdracht tussen opdrachtnemer en opdrachtgever op een visuele manier weer. Het is onder meer op dergelijke plaatsen in het proces dat koppelpunten zullen ontstaan.

In **voorbeeld 1** is al aangegeven hoe de communicatie binnen een project kan verlopen bij het ontdekken van nieuwe raakvlakken. Natuurlijk bestaan hiervoor meerdere mogelijkheden en is het vaak een kwestie van correcte afspraken te maken binnen een bedrijf. Toch zal bij de koppelpunten vaak informatie verloren gaan, zoals ook blijkt uit de onderstaande **figuur** [7].

Hulpmiddelen om de informatiestroom van een project te kanaliseren zijn dus allerminst overbodig.



Figuur 8: Praktische vs. theoretische informatieoverdracht [7]

2.4.1 Databeheersystemen

Om de projectinformatie te beheren maken sommige bedrijven gebruik van een Document Management System. Daarbij staan alle documenten op een gemeenschappelijke server. Indien een medewerker bepaalde informatie nodig heeft, kan hij deze gemakkelijk terugvinden. Bovendien kan de beheerder van het systeem bepalen wie toegang heeft tot de informatie. Na het ophalen van het gewenste document, mag de gebruiker er zeker van zijn dat niemand anders tegelijkertijd wijzigingen kan aanbrengen in het bestand. Dit systeem laat echter niet toe een duidelijke structuur in de beschikbare informatie te brengen. Zo kan moeilijk een verband gelegd worden tussen het ontwerpplan van een brug, het document met de bijbehorende eisen, de vergunningen voor de realisatie ervan enz.

Deze tekortkoming kan worden opgevangen door 'relationele' document management systemen te gebruiken. Verschillende softwareproducenten bieden dergelijke databanken aan, waarbij het tevens mogelijk is grafische en tekstuele bestanden aan elkaar te koppelen. Dit kadert binnen een

recente evolutie in de bouwsector, waarbij gestreefd wordt naar één globaal informatiemodel. Idealiter wordt voor een project een 3D-model ontworpen waarin alle nuttige informatie (zoals afspraken, (raakvlak)eisen, materiaalhoeveelheden ...) is terug te vinden. Dit model wordt aangeduid met de term Bouw Informatie Model, afgekort BIM. Hierop wordt verder dieper ingegaan.

Eén van de producentafhankelijke data management systemen is Autodesk Vault. Dit is een client-server applicatie waarmee zowel grafische CAD-bestanden als niet-grafische documenten beheerd kunnen worden. Op de server worden alle documenten en tekeningen bewaard zodat ze op één centrale locatie geraadpleegd kunnen worden. Bovendien kan, net zoals bij een document management system, bepaald worden welke gebruikers toegang hebben tot de informatie. Het 'ontlenen' van bestanden gebeurt op exact dezelfde wijze als bij het document management system. Daarnaast kan de server alle vorige versies van gewijzigde documenten bijhouden, wat aansluit bij de visie van SE om zo veel mogelijk alle projectinformatie te bewaren (zie hoger). De server bevat ook een relationele databank waarin eigenschappen van en relaties tussen de bestanden worden beheerd. Deze databank kan echter enkel relaties leggen tussen bestanden op de server, en niet tussen de eigenlijke projectinformatie (fysieke objecten, eisen, activiteiten ...).

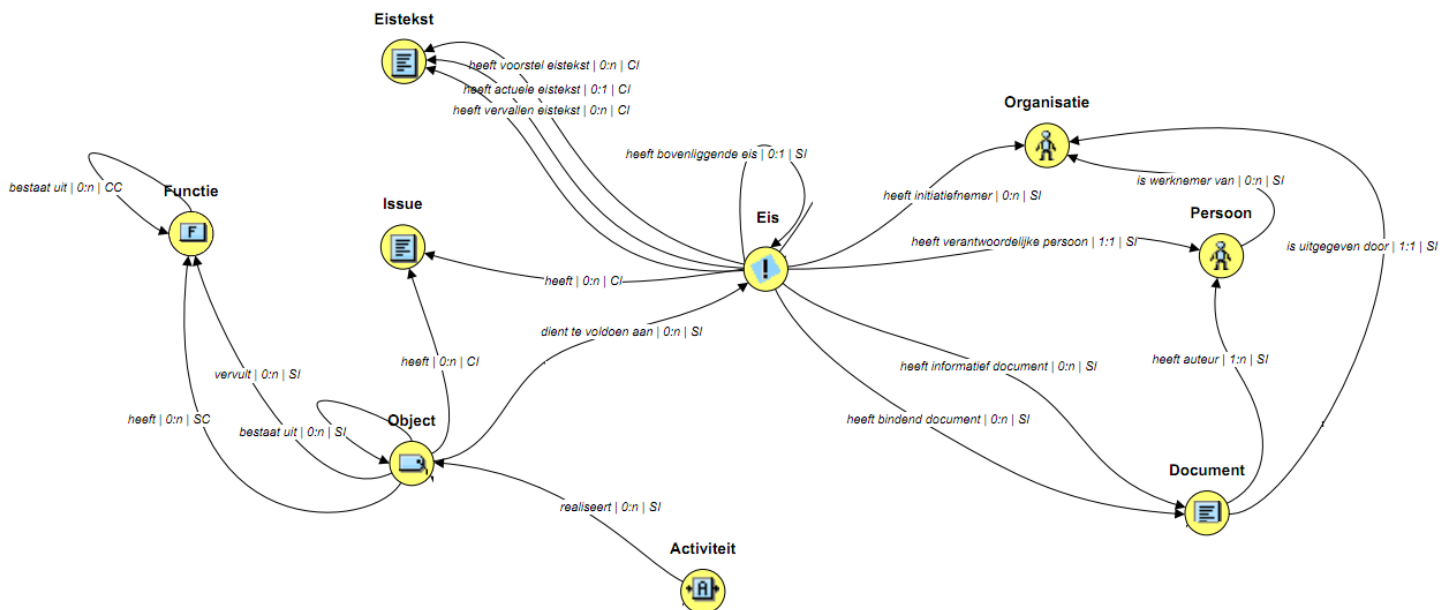
De client applicatie biedt toegang tot de Vault-server. Bovendien is het als ontwerper mogelijk via een plug-in rechtstreeks Vault-informatie op te halen vanuit een AutoCAD-applicatie. Deze plug-ins werken echter alleen voor bepaalde Autodesk-programma's en Microsoft Office. Dit impliceert dat nagenoeg het hele ontwerpproces met Autodesk-software uitgevoerd moet worden. Omdat de verschillende bedrijven in de praktijk al gekozen hebben voor een bepaald CAD-softwarepakket, is het dan ook weinig realistisch van elke partij te eisen met dezelfde software te werken.[8]

Om niet enkel informatie over de bestanden te beheren, maar ook over het project zelf, kan een document management systeem (bijvoorbeeld van een CAD-softwareproducent) gecombineerd worden met een relationeel informatiebeheersysteem. Eén van de mogelijkheden is het gebruik van Relatics, een web-based software applicatie uitgegeven door PKM Solutions. De databank is voor alle gebruikers gemakkelijk toegankelijk vanuit hun webbrowser. Een aangepast paswoord bepaalt dan de

functionaliteiten waarvan een end user gebruik kan maken. Hierbij dient te worden vermeld dat ook andere producenten relationele databanken hebben ontwikkeld. Relatics wordt echter frequent gebruikt op de Belgische en Nederlandse markt, daarom wordt dit beheersysteem verder uitgewerkt in deze tekst.

Het grote voordeel van Relatics is dat alle informatie slechts één maal vastgelegd moet worden. Hierdoor wordt vermeden dat een kunstwerk in de ene tabel of databank als KW_01 en in een andere als KW1 wordt benoemd. Tussen deze unieke objecten worden vervolgens relaties getrokken, waardoor een semantisch netwerk bekomen wordt.

De onderstaande **figuur** toont een (sterk vereenvoudigd) overzicht van een semantische structuur die in Relatics gebruikt kan worden. Het is natuurlijk mogelijk deze structuur naar wens aan te passen. In §4.3 – Implementatie van het proces in Relatics is beschreven hoe dit het best gebeurt voor het beheer van geometrische raakvlakken

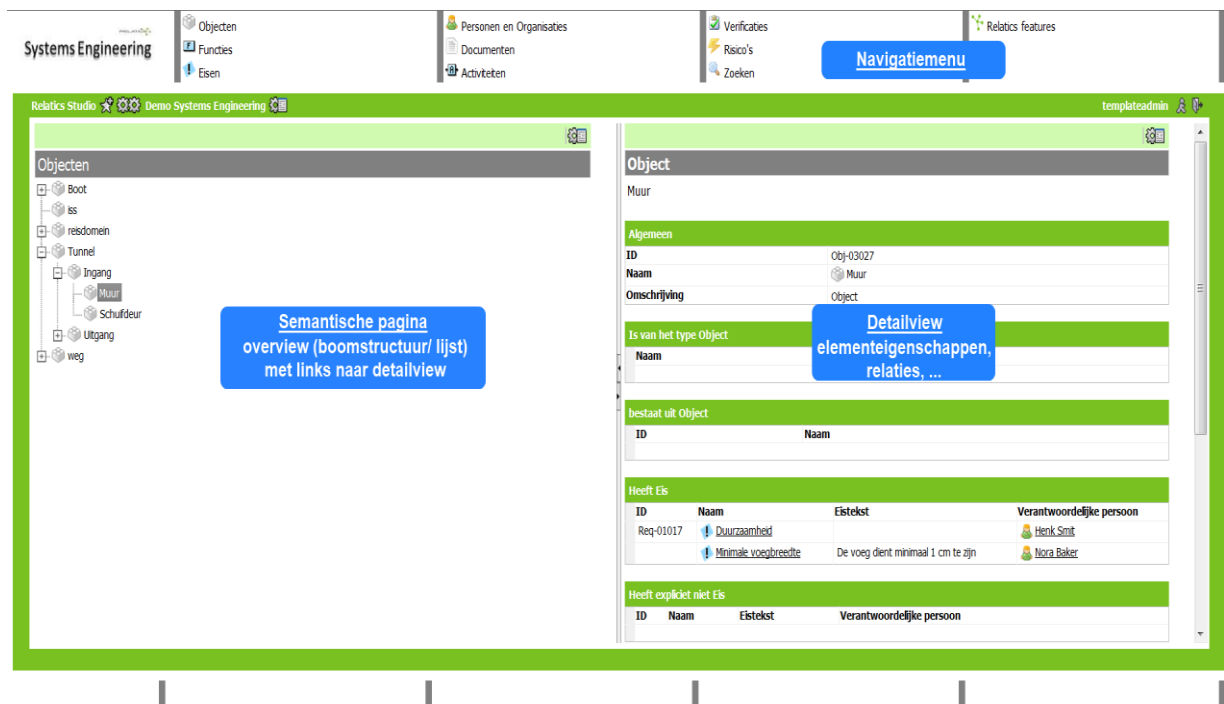


Figuur 9: Semantische structuur Relatics [9]

De gele cirkels in de bovenstaande **figuur** stellen entiteiten in Relatics voor: zo zijn er eisen, objecten, personen, activiteiten ... Hierbij dient te worden opgemerkt dat deze entiteiten 'objecten' uit de realiteit kunnen zijn, maar ze kunnen ook abstracte begrippen als functies voorstellen.

Zoals eerder aangehaald worden entiteiten in Relatics met elkaar verbonden door 'relaties', op de **figuur** voorgesteld door pijlen. Zo wordt het verband tussen een object en een eis weergegeven door een relatie met als benaming "dient te voldoen aan". Met behulp van deze relatie is het mogelijk een object (bv. 'brug') te verbinden met een eis (bv. 'doorrijhoogte').

Doordat Relatics de informatie ordent in verschillende boomstructuren (objectenboom, functieboom ...) kan dit programma als hulpmiddel bij het toepassen van SE aangewend worden. Praktisch ziet de Relatics-interface er als volgt uit [10]:



Figuur 10: Relatics-interface [10]

Door relaties te trekken tussen de verschillende entiteiten is het mogelijk op een heel flexibele en gemakkelijke manier alle gewenste informatie te ontsluiten. Bij de implementatie is het wel belangrijk dat niet te veel onzinnige relaties getrokken worden om te vermijden dat het systeem log en onoverzichtelijk wordt. Hoewel het bij de start van een nieuw project verleidelijk is onmiddellijk aan de slag te gaan met het toegeleverde Programma van Eisen, kan het toch tijdwinst opleveren om dit programma eerst te analyseren. Deze geanalyseerde eisen worden samen met alle

objecten, activiteiten, verantwoordelijken ... vastgelegd in een Relaticsomgeving voor dat project. Door voor de start van het ontwerp de moeite te nemen een databankomgeving op te zetten, werkt elke betrokken partij al van bij het begin met de correcte informatie. Bovendien bevindt die informatie zich op één vaste plaats, een zogenaamd 'Single Point of Information' (zie verder). De huidige versie van Relatics is echter nog niet helemaal geschikt als uitwisselingsplatform tussen verschillende betrokkenen. Een end user moet immers, telkens hij een bestand uit de databank bewerkt, het gewijzigde bestand opnieuw manueel uploaden. In Relatics kan bovendien enkel tekstuele informatie ontsloten worden, het koppelen van grafische entiteiten is niet mogelijk. Dit is in de bouwwereld, waar tekeningen nagenoeg het belangrijkste communicatiemiddel vormen, een belangrijke tekortkoming.

Relatics biedt wel de mogelijkheid informatie te exporteren als XML- of HTML-bestand, als Excelsheet of als grafische informatie in de vorm van tabellen, grafieken ... Hiervoor kan een statisch rapport aangemaakt worden, maar het is via webservices ook mogelijk de informatie op een dynamische wijze te koppelen. Hierbij dient te worden opgemerkt dat de conversie van XML- en HTML-bestanden naar Microsoft Word nog niet helemaal op punt staat. Wanneer een aannemer moet aantonen dat hij aan de door de opdrachtgever gestelde eisen voldoet, moet dit gebeuren door middel van tekstuele documenten. Het zou voor hem dus handig zijn uit Relatics onmiddellijk Word-documenten te kunnen genereren.

Toch is Relatics een krachtig hulpmiddel bij het toepassen van SE. Zo is het mogelijk de status van documenten, eisen ... weer te geven. Op deze manier gaat geen informatie verloren, maar zal de status van het item weergegeven of het nog actueel is (zie [figuur 11](#)). Zoals al aangehaald, biedt Relatics de mogelijkheid een flexibel en semantisch netwerk op te zetten. Een bijkomend voordeel is dat het systeem relatief eenvoudig geconfigureerd kan worden.

heeft actuele eistekst		
Omschrijving	Status	Reden vervallen
Aanplanting moet jaarlijks onderhouden worden	Actueel	

heeft vervallen eistekst		
Omschrijving	Reden vervallen	Vervallen door

heeft toelichting
Toelichting

Search

Actueel

Vervallen

Voorstel

close

Figuur 11: Status van items in Relatics [10]

In [hoofdstuk 4 – Raakvlakkenbeheer](#) is een systematiek voor een optimaal raakvlakkenmanagement uitgewerkt, en vervolgens geïmplementeerd in Relatics.

Er zijn natuurlijk nog alternatieven voor de bovenstaande informatiebeheersystemen beschikbaar. Een grote speler op de markt in Amerika is BIWTechnologies, een bedrijf dat naast projectondersteuning ook online applicaties aanbiedt voor het beheer van informatie in verschillende sectoren, gaande van infrastructuur tot verkoop. Net als bij Relatics is het mogelijk op elk moment de databank te raadplegen vanuit een webbrowser. Wanneer een gebruiker inlogt, komt hij terecht op een gepersonaliseerde pagina waar alle voor hem relevante documenten in het systeem getoond worden. Ook is het mogelijk toegangsrestricties op de informatie in het systeem te leggen, zodat de systeembeheerder kan bepalen wie toegang heeft tot welke documenten. Het beheer van de databank blijft wel in handen van BIWTechnologies. Een groot voordeel in vergelijking met Relatics is dat deze applicatie gebruik maakt van een 'geïntegreerde viewing technologie', waardoor het mogelijk is CAD-tekeningen en andere grafische informatie onmiddellijk vanuit de databank te bekijken en opmerkingen in te voegen. Het is dus niet nodig de gepaste CAD-software ter beschikking te hebben om deze grafische bestanden te kunnen bekijken. Dit opent perspectieven voor het beheer van raakvlakken. Wanneer een betrokkene een raakvlak of probleem wil melden, kan hij dit doen op basis van de grafische informatie, wat in de bouwwereld toch het eerste en belangrijkste communicatiemiddel is. Een opmerking op een

tekening, een verslag ... wordt in het systeem door een 'alert' onder de aandacht van alle betrokkenen gebracht. [11]

Algemeen kan worden gesteld dat het gebruik van databeheersystemen onontbeerlijk is geworden om alle informatie bij (grote) bouwprojecten te managen. De waarde van een databank wordt echter volledig bepaald door de inhoud en het beheer ervan. Het is dus belangrijk erop toe te zien dat de databank up-to-date blijft. Bovendien moet de op voorhand afgesproken structuur van de databank gerespecteerd worden. Indien dit niet gebeurt, komen bestanden op de verkeerde plaats in het systeem terecht, waardoor gebruikers na verloop van tijd niet meer weten waar ze bepaalde informatie kunnen vinden. Daarbij aansluitend hoort een correct versiebeheer zodat iedereen steeds met de meest recente informatie werkt. **Figuur 11** toont aan hoe dit versiebeheer in Relatics handmatig kan gebeuren.

2.4.2 Processen

Een grote valkuil bij het opzetten van een project is dat al snel te veel aandacht gaat naar de ondersteunende software (relationele databanken, MS-Project ...). Dit kan er vooral bij grotere projecten toe leiden dat iedereen de ondersteunende software op een andere manier gebruikt. Hierdoor zal het overzicht over het project gemakkelijk verloren gaan, met een verhoogd risico op fouten en faalkosten tot gevolg.

Om dit te vermijden is het belangrijk dat vooraf voldoende tijd wordt besteed aan het stroomlijnen van de *processen* achter het project.

Neem bijvoorbeeld een firma die voor een nieuw project een databank opzet. Als niet op voorhand eenduidig is vastgelegd op welke plaats in het systeem bepaalde informatie ingegeven moet worden, zal het voor een projectmanager al snel onmogelijk worden dit project op te volgen.

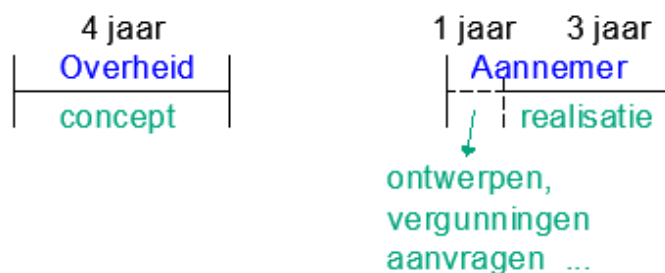
In het licht van de steeds toenemende tijdsdruk bij bouwprojecten, lijkt het opzetten van werkprocessen op het eerste gezicht een behoorlijke hoeveelheid bijkomend werk. De verleiding is immers bijzonder groot om zo snel mogelijk te beginnen aan het 'echte werk'.

Tot een paar jaar geleden stond de opdrachtgever zelf nog in voor het uitwerken van het concept en de ontwikkeling, waarna de opdrachtnemer de realisatie op zich nam. Zoals de **figuur** hieronder weergeeft, kon bij grote projecten in overheidsopdracht de fase van concept en ontwikkeling gemakkelijk tien jaar of langer duren, waarna de realisatie in een drietal jaar gebeurde.



Figuur 12: Projectduur en taakverdeling (traditioneel)

Sinds de opkomst van de DBFM-contractvorm is dit tijdsverloop behoorlijk omgegooid. De opdrachtgever/overheid zorgt enkel nog voor de uitwerking van het concept; de opdrachtnemer is verantwoordelijk voor het ontwerp, de aanvraag van vergunningen, de realisatie ... Dit geeft aanleiding tot de onderstaande **tijdslijn**.



Figuur 13: Projectduur en taakverdeling (heden)

Deze evolutie impliceert dat de opdrachtnemer in de (kortere) periode voor de bouwstart meer werk moet verzetten. Het spreekt voor zich dat het dan van cruciaal belang is om de bijkomende werkprocessen zo gestroomlijnd mogelijk te laten verlopen. Er gaan dan ook stemmen op om voor de start van het ontwerp de tijd te nemen deze processen vast te leggen en een databeheersysteem op te zetten. Met een relationele databank (zoals Relatics) kan het werk gestructureerd verlopen, is duidelijk wie verantwoordelijk is voor de realisatie van een object, het managen van een

raakvlak etc. Deze SE-tool is meteen een krachtig middel om de eisen te koppelen aan objecten en de relaties tussen deze objecten bloot te leggen.

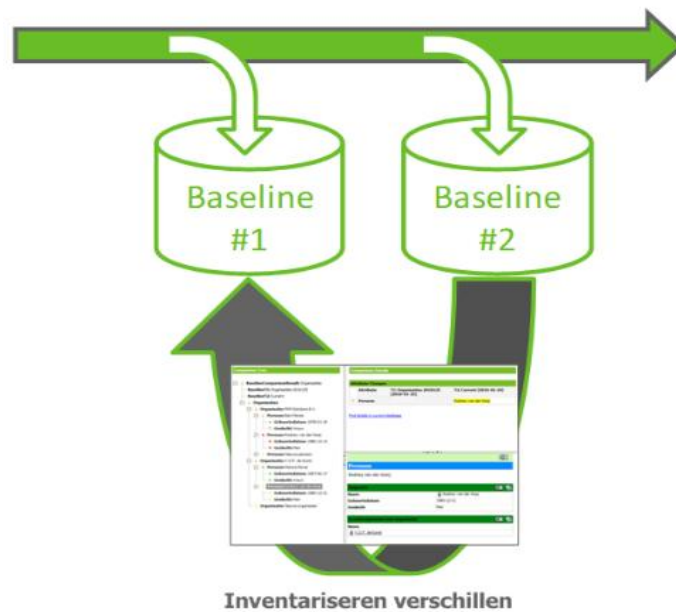
Geformaliseerde procedures zorgen ervoor dat alle betrokkenen een duidelijk beeld hebben van hun rol in het geheel en van de stappen die gevolgd moeten worden in bepaalde situaties. Het vastleggen van de procedures en hun output in een databeheersysteem (bijvoorbeeld Relatics) zorgt voor transparantie. Dit leidt tot meer gestructureerde processen en de zekerheid dat die processen correct afgehandeld worden.

2.5 **Virtueel bouwen**

Zoals hierboven werd uitgelegd, zullen steeds meerdere varianten aan de door de opdrachtgever gedefinieerde oplossingsruimte voldoen. Het virtuele bouwproces start zodra uit deze verschillende varianten de meest optimale is gekozen. Dit betekent meteen de start van een iteratief decompositieproces, waarbij wordt gekeken naar de functies die vereist zijn voor de gekozen oplossingsrichting. Vervolgens wordt gezocht naar bouwdelen (objecten) die deze functies kunnen realiseren.

Op deze manier doorloopt het virtuele bouwproces verschillende fasen, gaande van een voorontwerp tot een definitief model. Na het afsluiten van een fase wordt de vergaarde informatie vastgelegd in een zgn. 'baseline'. Deze informatie omvat het gemaakte ontwerp, maar ook alle beslissingen en afspraken die tot dan toe rond het ontwerp gemaakt zijn. Ook in de relationele databank Relatics is het mogelijk dergelijke baselines te maken. De onderstaande **figuur** geeft het principe van baselining visueel weer [10]. Op bepaalde momenten in het proces wordt de informatie in de databank als het ware 'bevroren', waardoor het eveneens mogelijk wordt na verloop van tijd de evolutie van het project in kaart te brengen. [7]

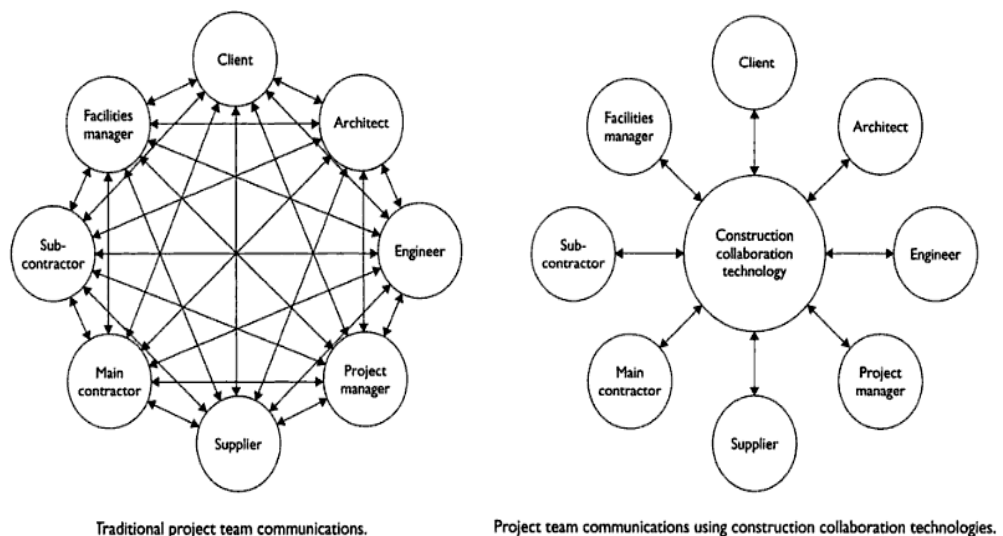
In de praktijk komt het, zeker bij grote infrastructuurprojecten, vaak voor dat al begonnen moet worden aan de uitvoeringsfase voordat alles in detail is uitgewerkt. Virtueel Bouwen is dus eerder een theoretisch, idealistisch concept dan een praktisch werkbaar principe.



Figuur 14: Baselining [10]

2.6 Building Information Model (BIM)

Zoals eerder al is aangegeven, wordt het beheren van alle projectinformatie een hele opgave, vooral wanneer het project een zekere omvang heeft. Omdat zoveel verschillende partijen behoefte hebben aan dezelfde gegevens is het wenselijk deze op eenzelfde locatie samen te brengen. Dit principe kan worden omschreven als *Single Point of Information (SPI)*. Dergelijk SPI kan verschillende vormen aannemen, zowel grafisch als tekstueel. In het eenvoudigste geval bestaat het SPI uit een gemeenschappelijke databank waarin de gegevens alleen in tekstvorm terug te vinden zijn. Aangezien in bouwprojecten ruimtelijke structuren worden verwezenlijkt, ligt het voor de hand een grafisch SPI op te bouwen. De onderstaande **figuur** geeft op een vereenvoudigde manier de meerwaarde van een SPI weer: de communicatie tussen de verschillende betrokkenen verloopt minder omslachtig en is beter gestructureerd.



Figuur 15: Informatieoverdracht: traditionele werkwijze vs. SPI [12]

Het hele project wordt bij een grafisch SPI weergegeven als een model waar de nodige informatie aan gekoppeld kan worden. Zo'n geïntegreerd informatiemodel is gekend onder de noemer *Building Information Model (BIM)*. Deze afkorting duidt niet enkel op een BIM als 3D-model, maar kan ook staan voor *Building Information ModellING*. Dit verwijst naar het

proces dat doorlopen wordt om te komen tot een Building Information Model [14].

Het BIM-principe wordt in andere sectoren al langer toegepast. Zo worden in de chemische industrie reactoren voorgesteld als gedetailleerde 3D-modellen. Deze modellen zijn niet beperkt tot het tonen van informatie; met de juiste softwaretools kunnen bijvoorbeeld ook leidingberekeningen uitgevoerd worden. Daarnaast is het eveneens mogelijk materiaalhoeveelheden snel en correct op te stellen aan de hand van het geïntegreerde model.

In de meest ideale vorm komen in een BIM alle ontwerpdelen van de verschillende disciplines samen in één model. Hierbij moet 'model' in de breedst mogelijke betekenis geïnterpreteerd worden: het gaat immers niet enkel om grafische gegevens, maar ook om tekstuele data die aan de relevante modelonderdelen gekoppeld zijn. Dit betekent bijvoorbeeld dat een raakvlakeis over de doorrijhoogte *in het model* gekoppeld wordt aan de objecten waar het raakvlak tussen geldt. [7]

Door het samenbrengen van de verschillende ontwerpen in één omgeving wordt het dus al mogelijk om een aantal geometrische raakvlakken te controleren. Denk hierbij bijvoorbeeld aan de aansluiting van een weg op een kunstwerk.

Een BIM-omgeving is met andere woorden niet alleen handig als Single Point of Information, zij kan ook een belangrijke bijdrage leveren aan het opsporen en beheren van raakvlakken, zowel proces- als softwarematig.

In §3.2 – [Bouw Informatie Model](#) wordt dieper ingegaan op het BIM-principe en de ondersteunde software.

2.7 **Marktonderzoek**

Gezien het praktijkgerichte karakter van raakvlakkenbeheer is een marktonderzoek uitgevoerd bij verschillende partijen. Zo hebben zowel opdrachtnemers als de overheid hun mening kunnen geven over deze problematiek. In wat volgt, worden de belangrijkste conclusies uit de gevoerde gesprekken aangegeven. De verkregen informatie omtrent het

gebruik van Geografische Informatiesystemen en de werking van een BIM is volledig opgenomen onder [hoofdstuk 3 – 3D-modellering](#).

Wanneer de term 'raakvlakken' valt, blijkt dat het beheer ervan vaak impliciet gebeurt. In het projectproces zijn 70 à 80 % van de afspraken logisch; de projecten worden dus voornamelijk gestuurd op basis van ervaring. Voor de overige 20 % is het belangrijk goede afspraken te maken, en deze vast te leggen in de vorm van voorgeschreven processen.

Uiteraard is het nuttig de elementen die door ervaring aangestuurd worden, ook expliciet vast te leggen. Op die manier wordt verzekerd dat deze kostbare ervaring niet verloren gaat. Hoewel processen niet overschat mogen worden, kan het in dit opzicht eveneens handig zijn een lijst met 'standaard' raakvlakken te definiëren.

Bij het beheer van een proces spelen een aantal kritieke succesfactoren. Zo moeten de processen zeer duidelijk worden vastgelegd en is het belangrijk dat de medewerkers een gepaste opleiding krijgen. Ook moeten de softwareprogramma's die het proces ondersteunen voldoende gebruiksvriendelijk zijn. In de loop van deze tekst zijn verschillende programma's aangehaald die kunnen helpen bij het beheer van raakvlakken. Software mag echter niet gezien worden als de ultieme tool voor projectbeheersing, de processen achter het gebruik van deze software zijn minstens even belangrijk.

In het algemeen kan worden gesteld dat een raakvlak ontstaat bij interactie tussen verschillende componenten. Aangezien een raakvlak geen afgeleerd begrip is, bestaat ook geen eenduidige definitie voor het vastleggen ervan. Er heerst in de praktijk dan ook veel onduidelijkheid over welke interacties als 'raakvlak' gezien moeten worden. Eerst en vooral is het belangrijk op te merken dat niet elk raakvlak als dusdanig beschouwd moet worden. Zo is het bijvoorbeeld vanzelfsprekend dat de betonplaten in een berlinerwand elkaar raken. Het is ook niet nodig alle raakvlakken expliciet vast te leggen en te beheren. In het algemeen blijkt dat het belangrijk wordt een raakvlak te beheersen als er een risico aan verbonden is. Op dat moment zal het geometrische raakvlak vastgelegd worden onder de vorm van een eis aan de relevante objecten. Zo geeft een procesverantwoordelijke aan dat beheersmaatregelen moeten getroffen worden om problemen omtrent het raakvlak te vermijden. Toch is het risicogehalte van een raakvlak niet het

enige criterium op basis waarvan deze als eis gedefinieerd zal worden. Wanneer het beheer ervan neerkomt op het volgen van standaardprocedures en normen, zou beslist kunnen worden dit raakvlak niet expliciet in een eistekst te gieten. Een voorbeeld hiervan is het respecteren van een betondekking. In [hoofdstuk 4 – Raakvlakkenbeheer](#) wordt echter uitgelegd dat bij elk raakvlak exact één eis gedefinieerd wordt om inconsistenties in het beheersysteem te vermijden. Daarnaast kunnen verschillende raakvlakken op basis van ervaring gemakkelijk beheerst worden. Zo zullen met het oog op het rijcomfort riooldeksels niet in een fietsstrook worden aangelegd; met dit ontwerpaspect zijn ontwerpers vertrouwd en ze zullen het ook automatisch respecteren.

De resultaten uit het marktonderzoek geven aan dat het beheer van raakvlakken vooral werkbaar moet blijven. Het is niet nodig eindeloos raakvlakken en eisen te definiëren, maar veeleer op een duidelijke manier te communiceren met elkaar.

Net zoals het niet wenselijk is te veel raakvlakken te definiëren, zal ook geprobeerd worden het aantal eisen in de eisenboom van het project te beperken. Deze worden in eerste instantie door de overheid geformuleerd op basis van een referentieontwerp. Dit ontwerp gaat niet tot op detailniveau, maar is bedoeld om er zeker van te zijn dat alle nodige aspecten opgenomen worden in het Programma van Eisen. De eisen worden op een functionele manier omschreven, en dit op een zo hoog mogelijk niveau. Omdat ze op basis van de gewenste functies van het bouwwerk worden vastgelegd, zullen raakvlakeisen niet als een aparte categorie voorkomen. In de praktijk blijkt dat de aangeleverde eisen door de aannemers vaak niet als voldoende oplossingsvrij worden ervaren. Daarbij komt nog dat vele eisen heel algemeen gespecificeerd zijn, bijvoorbeeld “Het ontwerp van een fietspad moet gebeuren volgens de bepalingen van het fietsvademecum”. De overheid kan met deze eis al zijn wensen dekken, maar de aannemer zal deze verder concretiseren. De tekstuele bepalingen moeten dus omgezet worden in SMART-eisen zodat ze verifieerbaar zijn. Zo zal de opdrachtnemer een eis die stelt dat “de levensduur van een voeg 40 jaar moet zijn”, vertalen naar de benodigde hoeveelheid wapening, rubber ...

Het oplossingsvrij specificeren van eisen is uiteraard niet eenvoudig. De overheid moet ook bepalingen vanuit bijvoorbeeld een MER (Milieu Effecten Rapport) respecteren. Deze eisen zijn van nature niet

oplossingsvrij; zo kan om eventuele geluidshinder te vermijden de aanleg van een asfaltverharding verplicht worden. Aangezien de eisen uit het MER strikt nageleefd moeten worden, kunnen deze niet-functionele eisen moeilijk veralgemeend worden tot op functioneel niveau. Het is wenselijk dergelijke bepalingen in de toekomst te definiëren onder de vorm van functionele eisen.

De functionele eisen kunnen bovendien opgeslagen worden in een databank, waardoor een soort 'eisenbibliotheek' verkregen wordt. Deze werkwijze op consequente en systematische manier aanhouden, is echter economisch moeilijk haalbaar. De enorme bibliotheek van eisen moet immers ook up-to-date gehouden worden. Alleen al het beheren van de verschillende versies is nauwelijks doenbaar voor dergelijke grote hoeveelheden informatie. Een aannemer zou echter wel een database van eisen kunnen opzetten voor ontwerpplannen die hij vaak moet maken. Zo heeft hij vaak terugkerende bepalingen uit het Programma van Eisen al op voorhand ter beschikking in zijn databank. Ook voor raakvlakken zou op deze manier een lijst opgesteld kunnen worden met elementen die zeker gecheckt moeten worden bij het ontwerp van bijvoorbeeld een weg. Uiteraard mag het opstellen van generieke raakvlakken niet betekenen dat raakvlakkenbeheer herleid wordt tot het doornemen van een checklist.

In het Programma van Eisen wordt voor elke bepaling aangegeven of deze al dan niet bindend is. Bij een bindende eis ligt zowel het resultaat vast als de wijze waarop het moet bereikt worden. Een niet-bindende eis bepaalt daarentegen enkel het gewenste resultaat. Zo kan gespecificeerd zijn dat een rotonde moet gerealiseerd worden, maar mag de aannemer zelf kiezen of hij deze in doorgaand gewapend beton of in asfalt uitvoert. Daarnaast wordt per eis aangegeven of deze aantoonbaar moet zijn en in welke fase van het ontwerp dit moet gebeuren. Ook bij een niet-aantoonbare eis kan de overheid achteraf nog vragen aan de aannemer om te bewijzen dat zijn ontwerp voldoet.

Het opgemaakte referentieontwerp is uiteraard niet bindend; geheel volgens de regels van DBFM mogen potentiële opdrachtnemers zelf voorstellen indienen. De overheid probeert daarbij de studiekosten van de aannemers te drukken door het aanbestedingsproces op te splitsen in verschillende fasen. De 1^e offerte zorgt voor een schifting van de potentiële

opdrachtnemers en pas bij de 2^e offerte wordt aan de resterende kandidaten gevraagd een volledig ontwerp op te maken. Deze 2^e offerte kan eventueel gevolgd worden door een Best And Final Offer (BAFO). De ontwerpen van potentiële opdrachtnemers worden bij DBFM-projecten met behulp van globale beoordelingscriteria geëvalueerd. Het intrinsieke karakter van deze projecten impliceert immers dat de opdrachtgever op voorhand niet kan weten welke ontwerpen zullen ingediend worden. In vergelijking met klassieke aanbestedingen wordt de beoordelingsmarge dus ruimer gehouden door de criteria minder gedetailleerd te formuleren. Momenteel is door de Vlaamse Overheid bepaald welke werken als DBFM-project aanbesteed zullen worden. Het Vlaams Kenniscentrum PPS heeft ondertussen een scan ontwikkeld, waarmee in de toekomst beslist kan worden of een opdracht in aanmerking komt om als DBFM-project aanbesteed te worden.

De Vlaamse Overheid legt in het kader van de realisatie van de zgn. 'Missing Link' projecten, het verplicht gebruik van SE op. Dit doet ze door expliciet de naleving van de norm ISO 15288 te eisen. Deze norm, getiteld 'Systems and software engineering - System life cycle processes', beschrijft de fasen in de levenscyclus van systemen, en geeft verscheidene processen om elke fase efficiënt te beheersen [13].

De meeste opdrachtnemers maken daarbij dan ook gebruik van een softwareondersteuning (bv. Relatics). Vanuit de overheid wordt echter geen initiatief genomen om dergelijk databeheersysteem op te zetten. Het studie bureau, dat zich in opdracht van de overheid bezighoudt met de opmaak van het referentieontwerp en het bestek, ontwikkelt meestal voor zichzelf wel een digitale omgeving. De opdrachtnemer werkt met deze omgeving echter (nog) niet verder, en wel om de volgende redenen:

- Eigenlijk zou via marktonderzoek bepaald moeten worden welke SE-tool de beste is. Het probleem blijft dan nog dat de opdrachtgever aan de aannemer niet kan opleggen om SE met behulp van bijvoorbeeld Relatics te implementeren. Zo wordt immers een concurrentieel voordeel gegeven aan bepaalde partijen die reeds vertrouwd zijn met dit databeheersysteem. Bovendien druist het opleggen van één welbepaalde IT-toepassing in tegen de vrijheid van SE. De overheid kan er wel voor opteren de informatie omtrent de bestekbepalingen via een omgevingsonafhankelijke standaard (zoals XML) vrij te geven. Dit

zou de opdrachtnemer toelaten de informatie op eenvoudige wijze in het systeem van zijn keuze in te laden.

Door de SE-informatie vast te leggen in een gestandaardiseerde databankomgeving, zou het bovendien mogelijk worden op een geautomatiseerde manier omgevingen van verschillende projecten aan elkaar te linken. Een voorbeeld hiervan is het recentelijk ontwikkelde Gellish, waarbij de implementatie van een databank gebeurt via het toekennen van standaardrelaties.

- Aangezien de projectspecifieke softwareomgeving niet wordt doorgegeven aan potentiële opdrachtnemers, kunnen zij dus zelf kiezen met welk systeem ze willen werken.

Om het werk van de opdrachtnemer te kunnen verifiëren, heeft de overheid toegang tot het databeheersysteem van het project. De overheid heeft ondertussen zelf ook een 'e-room' geïnstalleerd waarlangs de documentenstroom passeert. Documenten uit de SE-tool van de aannemer worden door de opdrachtgever geüpload en in deze e-room opgeslagen. Zo heeft de overheid alle documenten in eigen beheer, wat belangrijk kan zijn in geval van discussie. Daarnaast zal de aannemer ook officiële documenten op papier moeten voorleggen ter acceptatie.

Dit is natuurlijk geen ideale manier van werken. Door een gemeenschappelijke database te creëren tussen opdrachtgever en -nemer zou een uitwisselingsplatform kunnen ontstaan, waarmee de overheid onmiddellijk alle gewenste documenten ter beschikking heeft. Zo kan een aannemer, wanneer hij een eis gecontroleerd heeft, de aantoondocumenten onmiddellijk in het systeem koppelen aan de desbetreffende eis. Om praktische redenen kiest de overheid er momenteel niet voor deze werkwijze te hanteren.

2.8

Besluit

De invoering van de DBFM-contractvorm heeft een cultuuromslag teweeggebracht in de manier waarop zowel opdrachtgever als -nemer een project benaderen. Opdrachtnemers worden immers verantwoordelijk gesteld voor het ontwerpen, uitvoeren, onderhouden en financieren van de

structuur. Aannemers zullen dus moeten proberen het hele project te optimaliseren en elk van deze domeinen tegenover elkaar af te wegen. De toename in verantwoordelijkheden brengt bijkomende risico's met zich mee, waardoor de potentiële faalkosten stijgen. Opdrachtnemers worden in dit opzicht gedwongen een project met de nodige omzichtigheid te beheren. De methodiek van Systems Engineering kan hierbij een houvast bieden. Door een systeem gestructureerd te decomponeren en te analyseren, kan bij elke iteratiestap gekozen worden voor een oplossing die enerzijds werkbaar is voor de opdrachtnemer en anderzijds aan de klantvraag voldoet. Het toepassen van de SE-filosofie betekent bovendien dat projecten op een open en transparante manier gerealiseerd worden.

Bij de decompositie van een systeem volgens de SE-methodiek ontstaan steeds interne raakvlakken tussen de verschillende subsystemen en objecten. Externe raakvlakken ontstaan doordat het systeem in een bestaande omgeving ingepast moet worden.

Een goed beheer van deze raakvlakken is cruciaal om zoveel mogelijk faalkosten te vermijden. Dit kan gebeuren op basis van een databeheersysteem als Relatics. Om te vermijden dat het systeem onwerkbaar en complex wordt, zullen enkel raakvlakken waar voldoende risico aan verbonden is, expliciet vastgelegd worden.

Omdat raakvlakkenbeheer onlosmakelijk verbonden is met een correcte en efficiënte informatiestroom, is het belangrijk dat deze informatie zo volledig mogelijk wordt doorgegeven aan alle betrokken partijen. Daarvoor zijn verschillende hulpmiddelen beschikbaar, gaande van document management systemen tot geavanceerde BIM-software.

Hierbij mag echter niet uit het oog verloren worden dat softwaretoepassingen slechts een hulpmiddel zijn, zonder ondersteunende processen is een correct projectbeheer onmogelijk.

In [hoofdstuk 3 – 3D-modellering](#) wordt aangegeven hoe 3D-geometrie kan helpen bij het opsporen en vermijden van raakvlakproblemen. Daarbij worden verschillende softwaretoepassingen die een zgn. *clash detection* toelaten, verder uitgewerkt.

HOOFDSTUK 3

3D-modellering

Het vorige hoofdstuk geeft de noodzaak en het referentiekader aan om te komen tot een efficiënt raakvlakkenbeheer. Raakvlakken zijn een complexe materie, en het beheer ervan moet in de eerste plaats praktisch werkbaar zijn. Daarom neemt dit hoofdstuk enkele softwarematige hulpmiddelen onder de loep. Uiteraard wordt daarbij nagegaan of deze voldoende ondersteuning kunnen bieden bij het controleren van raakvlakken, maar ook de mogelijkheden met betrekking tot het algemene projectmanagement worden daarbij niet uit het oog verloren.

De meeste softwareomgevingen maken gebruik van 3D-modellen. Aangezien het ontwerp tegenwoordig meestal nog in 2D gebeurt, wordt in [§3.1 – 3D-Modellering en raakvlakkenbeheer](#) aangegeven waarom 3D-modellen toch aangewezen zijn voor het beheer van raakvlakken.

[§3.2 – Bouw Informatie Model](#) gaat dieper in op het efficiënt beheer van informatie en de bijbehorende ICT-tools. Eén van de mogelijke softwareomgevingen om de projectgegevens samen te brengen is een Geografisch Informatiesysteem. De werking van een dergelijke omgeving en de mogelijkheden inzake raakvlakkenbeheer worden uitgewerkt in [§3.3 – Geografische Informatiesystemen](#).

3.1 3D-modellering en raakvlakkenbeheer

In de praktijk wordt meestal met tweedimensionale plannen gewerkt, in combinatie met doorsnedetekeningen of hoogteaanduidingen (2.5D). Uitvoeringsteams op de werf zijn gewend dergelijke 2D-tekeningen te lezen en deze om te zetten naar de eigenlijke uitwerking. Toch geeft deze werkwijze vaak aanleiding tot fouten. Niet alleen omdat de 2D-plannen verkeerd geïnterpreteerd kunnen worden, maar ook omdat bepaalde

aspecten bij het ontwerp uit het oog worden verloren. Denk daarbij maar aan het klassieke voorbeeld van een te kleine doorrijhoogte, een verkeerde positionering van de wachtwapening bij betonwerken enz. De praktijkvoorbeelden zijn talrijk, terwijl vele problemen vermeden kunnen worden door het ontwerp geheel of gedeeltelijk in 3D uit te werken. Dit is ook gebleken bij het project van de Noorderlaanbruggen in Antwerpen, zoals is uitgelegd in [hoofdstuk 5 – Praktisch project](#).

Aangezien de werkelijkheid rondom ons een driedimensionale omgeving is, kunnen driedimensionale plannen op een meer intuïtieve manier begrepen worden. Elke betrokkene, zelfs diegene zonder een bouwtechnische achtergrond, is in staat zich een beeld te vormen van het uiteindelijke bouwwerk. In de conceptfase kunnen 3D-modellen ook helpen het maatschappelijke draagvlak voor het project te vergroten, mede omdat 3D-visualisaties meer tot de verbeelding spreken. Ook de inplanting van de projectinfrastructuur in de bestaande omgeving zal duidelijker voorgesteld kunnen worden, wat het opsporen van externe raakvlakken bevordert.

In de ontwerpfase maken 3D-modellen de opbouw van een driedimensionaal BIM mogelijk, waardoor raakvlakken zowel visueel als automatisch opgespoord kunnen worden (zie ook [§3.3 Bouw Informatie Model](#)). Daarnaast kan het opnemen van het tijdsaspect in het BIM (4D) toelaten raakvlakken tussen activiteiten te detecteren. Geavanceerde softwarepakketten kunnen bovendien uit de 3D-modellen materiaallijsten opstellen en zo een idee geven van de bijbehorende kostprijs (5D), wat zeker bij DBFM-projecten een handig hulpmiddel kan zijn.

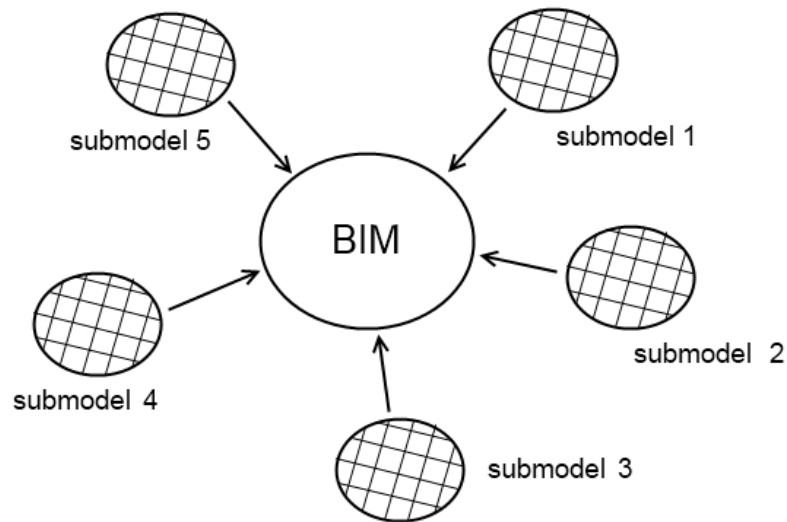
Op het eerste gezicht lijkt het maken van een 3D-ontwerp meer tijd te vragen en dus hogere kosten met zich mee te brengen. Praktijkvoorbeelden tonen echter aan dat in het verdere projectverloop de iets langere ontwerptijd ruimschoots gecompenseerd wordt. Bovendien zullen, zoals hierboven uitgelegd, raakvlakken makkelijker gedetecteerd kunnen worden. Dit betekent automatisch een daling van de risico's en de bijbehorende faalkosten.

Aangezien 2D-plannen (nog) niet weg te denken zijn uit de dagelijkse praktijk, wordt in [§4.2.4 – Opmvolging & verificatie](#) met een voorbeeld aangegeven hoe raakvlakken in dit geval beheerd kunnen worden.

3.2 Bouw Informatiemodel

Zoals aangehaald in §2.6 - [Building Information Model \(BIM\)](#), is het BIM-concept erop gebaseerd dat alle beschikbare informatie over een bouwproject, gaande van het bestek tot plannen en 3D-modellen, op één plaats wordt samengebracht. Op die manier komt een Single Point of Information (SPI) tot stand. Eén van de voor de hand liggende voordelen van een dergelijk SPI is dat iedereen steeds over de laatste versie van de gegevens beschikt. De mogelijkheden van een BIM gaan echter verder dan die van een SPI: waar een SPI enkel een verzamelpunt van gegevens is, zal een BIM de modelgegevens uit verschillende plannen ook samenbrengen in één geïntegreerd model. Dit maakt het mogelijk om snel – en op een visuele manier – na te gaan of de verschillende plannen correct op elkaar aansluiten. Het biedt meteen ook de gelegenheid om geometrische raakvlakken te controleren en op te volgen.

Zowel vanuit technisch als organisatorisch oogpunt is het echter niet vanzelfsprekend een dergelijk geïntegreerd model up-to-date te houden. Aanpassingen in de verschillende deelontwerpen moeten ook doorgevoerd kunnen worden in het BIM, al dan niet door middel van een dynamische koppeling. Anderzijds kan het handig zijn wijzigingen en opmerkingen onmiddellijk in het BIM aan te brengen, en op deze wijze te communiceren met de verschillende betrokkenen. In de centrale database zijn zowel het BIM als de aparte deelontwerpen ondergebracht. Verschillende ontwerpteams zouden zo wijzigingen in elkaars modellen kunnen doorvoeren. Dit kan worden opgelost door restricties te leggen op verschillende delen van het model, waardoor ontwerpers enkel wijzigingen kunnen aanbrengen in het door hen ontworpen deel van de infrastructuur. Een andere, meer werkbare, mogelijkheid is gebruik te maken van momentopnames van de modellen: de verschillende deelontwerpen worden samengebracht in een traditionele databank die fungeert als SPI. Uiteraard worden op deze deelontwerpen ook restricties gelegd. Vanuit het SPI kunnen de verschillende modellen dan op regelmatige tijdstippen samengevoegd worden tot één BIM.



Figuur 16: Werkingsprincipe van een BIM

De bovenstaande **figuur** geeft een schematische voorstelling van deze werkwijze, die bij een aantal bedrijven al in beperkte mate wordt toegepast voor het ontwerp van grote infrastructuurprojecten.

Om deze methode te implementeren voor volledige projecten moet aan een aantal voorwaarden voldaan zijn:

1. Er moet een functionerend SPI beschikbaar zijn zodat steeds gewerkt wordt met de laatste beschikbare versie van de plannen. Met een 'functionerend SPI' wordt bedoeld dat het cruciaal is dat de juiste bestanden op de correcte plaats in het systeem staan.
2. Om geometrische raakvlakken (zoals een doorrijhoogte) te controleren, zijn de relevante ontwerpen best in 3D gemaakt.
3. Er is een softwareplatform nodig dat de verschillende ontwerpdelen kan inlezen.

Het is vanzelfsprekend dat steeds aan de eerste en derde voorwaarde moet voldaan worden: raakvlakcontroles hebben enkel zin als de juiste versie van de plannen gebruikt wordt. Bovendien vereist het samenbrengen van de relevante plannen de juiste software die dergelijke operatie

ondersteunt. De tweede voorwaarde lijkt eveneens logisch (zie §3.1 – 3D modellering en raakvlakkenbeheer), maar in de meeste disciplines van de bouwsector wordt vooral met 2D-plannen gewerkt. Wegen en kunstwerken worden wel steeds vaker als 3D-modellen ontworpen. Dit maakt een raakvlakkencontrole tussen beide in één geïntegreerd model mogelijk.

Het integreren van 3D-ontwerpmodellen in één softwareomgeving kan op verschillende manieren gebeuren. Softwareproducenten als AutoDesk en Bentley bieden programma's aan die het mogelijk maken modellen uit verschillende bestanden samen te voegen. Daarbij is het wel belangrijk dat het gekozen pakket voldoende flexibel is om alle nodige modellen te kunnen inlezen. Met 'flexibel' wordt bedoeld dat de software in staat moet zijn om zowel verschillende bestandsformaten, als de aanzienlijke bestandsgroottes te verwerken.

De verschillende ontwerpprogramma's bieden meestal (beperkte) mogelijkheden voor het inlezen van bestanden uit andere pakketten. Deze functies zijn vaak echter ontoereikend om een volwaardig BIM op te zetten. Een positieve evolutie is wel dat steeds meer softwareproducenten ernaar streven hun pakketten een betere compatibiliteit te geven. Daarnaast kunnen gegevens over het CAD-model verloren gaan wanneer dit wordt geïmporteerd in een ander programma. Om dergelijke importproblemen te vermijden, heeft BuildingSmart¹ een producentonafhankelijk bestandsformaat, Industry Foundation Classes (IFC), ontwikkeld.

3.2.1 IFC

Aan de basis van IFC ligt een set van afspraken, waarmee wordt vastgelegd hoe een softwarepakket objecten en eigenschappen van die objecten moet wegschrijven. Dankzij deze afspraken 'weet' een ander programma op welke plaats in het IFC-bestand bepaalde informatie opgehaald kan worden. De afspraken leggen softwareproducenten die IFC willen ondersteunen op *hoe* ze hun pakket de informatie moeten laten opslaan. Op deze manier wordt afgedwongen dat programma's van verschillende producenten gegevens over structurelementen op nagenoeg dezelfde wijze wegschrijven in een IFC-bestand. Door de elementen in een

¹ BuildingSmart, voorheen gekend als het International Alliance for Interoperability (IAI).

dergelijk IFC-bestand op te slaan, kunnen deze gemakkelijk in andere pakketten herkend en geopend worden. Het zou in de toekomst handig zijn de gestandaardiseerde objecten van IFC te linken aan de objectenboom in de databankomgeving van een project. Op die manier wordt IFC in beperkte mate gekoppeld aan de SE-decomposities.

IFC bevat afspraken voor de meeste klassieke elementen zoals deuren en muren, maar het is natuurlijk onmogelijk elk object vooraf in een dergelijke standaard vast te leggen. Om die reden is er een uitbreiding voor IFC voorzien: IFD². Dit begrip wordt voor meerdere zaken gebruikt, maar in deze context kan het gezien worden als een uitbreiding van IFC. Het gaat immers over bijkomende sets van afspraken die gelden binnen een beperkte groep mensen. Deze uitbreiding van IFC laat niet enkel toe dat aan elementen die al in de IFC-standaard zitten bijkomende eigenschappen gekoppeld worden, maar dat ook nieuwe elementen in een IFC-bestand opgeslagen kunnen worden.

Het spreekt voor zich dat zowel voor IFC als voor IFD de volgende afspraak gerespecteerd moet worden: iedereen die deze standaarden gebruikt moet de informatie op de afgesproken manier structureren.

De open IFC-standaard laat zo toe alle mogelijke structuurelementen op een eenduidige manier te beschrijven in een IFC-bestand. [14]

Vele CAD-programma's ondersteunen reeds het IFC-formaat, met als belangrijke uitzondering een aantal softwarepakketten van Autodesk.

3.2.2 BIM-software

Steeds meer softwareproducenten bieden hun gebruikers de mogelijkheid bestanden op te slaan of te exporteren naar dit IFC-formaat. Zo ondersteunen een aantal pakketten van Bentley Systems en AutoDesk dergelijke operaties. Daarnaast zijn er ook een aantal programma's (bijvoorbeeld Solibri) die zich profileren als omgevingen waarin IFC-bestanden samengebracht en bekeken kunnen worden.

² IFD staat voor International Framework for Dictionaries [14].

Veel van deze omgevingen laten niet enkel een handmatig, visueel nazicht van raakvlakken toe, maar bieden ook de mogelijkheid 'automatische' controles op het geïntegreerde model uit te voeren. Deze controles - in de softwarepakketten vaak 'clash detection' of 'collision detection' genoemd - gebeuren op basis van regels die de gebruiker zelf kan definiëren. Op deze manier laat een BIM toe een stap verder te gaan dan het manueel verifiëren van geometrische raakvlakken.

Aangezien de meeste programma's in meer of mindere mate gebruikers toelaten zelf controleregels op te stellen of te wijzigen, is het mogelijk te bepalen welke raakvlakken uit een analyse naar voren moeten komen. Het is immers niet nodig elke interactie tussen objecten nader te bekijken. Denk hierbij bv. aan een rioleringsstelsel in de grond: de grond raakt het rioleringsstelsel langs alle kanten, maar dit is uiteraard een raakvlak dat niet beheerd moet worden. Anderzijds zijn er vaak raakvlakken tussen objecten die elkaar niet rechtstreeks raken, maar die bv. om onderhouds- of veiligheidsredenen ruimtelijk gebufferd moeten worden. Een voorbeeld hiervan is het voorzien van voldoende ruimte rond een ventilatiesysteem zodat nazicht of vervanging mogelijk is.

In bovenstaand voorbeeld komt de noodzaak om zelf controleregels te kunnen opstellen in het BIM, duidelijk naar voren. Onderstaande paragrafen geven voor een aantal softwarepakketten aan of deze functionaliteit aanwezig is, en schetsen kort wat de voor- en nadelen van de verschillende pakketten zijn. Hierbij dient te worden opgemerkt dat het overzicht op geen enkel moment tracht volledig te zijn. Er bestaan ook andere softwarepakketten die *clash detection* ondersteuning bieden, zoals Tekla.

SOLIBRI

Het volledige softwarepakket van Solibri bestaat uit verschillende onderdelen. Solibri Model Checker laat toe controles uit te voeren op geïmporteerde modellen. De resultaten van deze controles kunnen met alle teamleden gedeeld worden via de vrij verkrijgbare Model Viewer. Ten slotte is het mogelijk gedetecteerde problemen te visualiseren in de CAD-omgeving door middel van de Issue Locator.

De Solibri-software brengt de verschillende deelontwerpen van een project samen in een BIM-omgeving. Op deze manier kunnen modellen afkomstig uit verschillende programma's gebundeld worden in één softwarepakket. Dit maakt dat deze niet enkel visueel bestudeerd kunnen worden, maar het wordt ook mogelijk geautomatiseerde raakvlakcontroles uit te voeren dankzij de clash detection functionaliteiten.

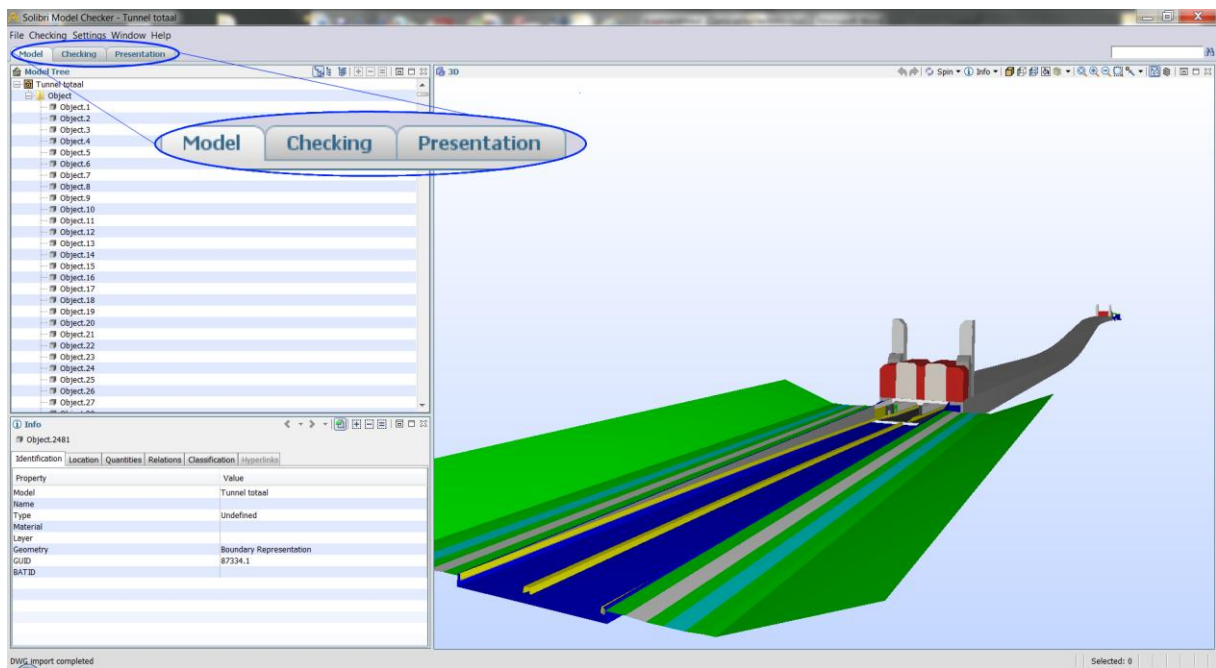
Kenmerkend voor Solibri is dat de software specifiek ontwikkeld is met het oog op IFC. Het programma ondersteunt enkel het IFC-formaat en DWG-bestanden. Voorlopig zijn er op dit moment geen plannen dit uit te breiden naar DGN (dataformaat van Bentley). Dit betekent dat de gebruikte CAD-software de mogelijkheid moet bieden ontwerpmodellen op te slaan in deze bestandsextensies.

In wat volgt worden de verschillende Solibri-pakketten kort besproken. [15]

Solibri Model Checker

Solibri Model Checker vormt de kern van het hele Solibri-pakket. Dit programma laat toe om ontwerpplannen uit een CAD-omgeving zowel handmatig als op basis van geprogrammeerde regels (zgn. 'rule sets') te controleren. Zo is het bijvoorbeeld mogelijk om na te gaan of een sparing in een betonnen balk op de juiste plaats zit voor een doorgaande leiding. Standaard wordt de Model Checker geleverd met 32 rule sets, die de gebruiker in beperkte mate kan uitbreiden met behulp van een Rule Set Manager.

Zoals op de onderstaande **figuur** te zien is, bestaat de interface van de Model Checker uit drie delen : in 'Model' kan met verschillende tools doorheen het model genavigeerd worden om een visuele controle uit te voeren. Het 'Checking'-deel biedt de mogelijkheid op basis van de eerder vermelde rule sets automatische (raakvlak)controles uit te voeren. Ten slotte maakt het 'Presentation'-luik het mogelijk om de resultaten van de controles visueel samen te brengen met de desbetreffende ontwerpdelen; deze worden verwerkt in slides tot een presentatie. [16]



Figuur 17: De interface van Solibri

Het combineren van verschillende modellen is een nodige stap om een BIM te kunnen opzetten, maar dit kan wel leiden tot zeer grote bestanden. Om de bestandsformaten in te perken heeft Solibri een 'IFC optimizer'³ ontwikkeld. Hiermee kunnen geïmporteerde IFC-bestanden gereduceerd worden tot 5 à 10 % van hun oorspronkelijke grootte.

Eén van de beperkingen van de Model Checker software is dat modellen groter dan 10 km renderproblemen kunnen geven, waardoor het pakket minder bruikbaar is voor zeer grote infrastructuurontwerpen. Zoals hierboven uitgelegd maakt IFC gebruik van standaardafspraken waarmee beschreven wordt hoe objecten gedefinieerd moeten worden. Deze afspraken zijn al vrij uitgebreid opgesteld met het oog op gebouwen, maar zijn in mindere mate beschikbaar voor civiele projecten. Uiteraard zal het detailniveau van het ontwerp mee bepalend zijn voor de maximaal te importeren modelgrootte. Bovendien werken heel wat bedrijven grotendeels met ontwerpsoftware van een welbepaalde producent (AutoDesk, Bentley ...). Het is in dit geval dan ook aangewezen een clash detection software van dezelfde producent te gebruiken. Zo zal het verlies

³ De IFC Optimizer is verkrijgbaar via <http://www.solibri.com/solibri-ifc-optimizer.html> (geraadpleegd op 11.12.2010).

aan modelinformatie bij het inladen in deze software tot een minimum beperkt blijven (zie verder: beperkingen van clash detection software).

Qua prijs zit de Model Checker van Solibri in dezelfde orde van grootte als AutoDesk Navisworks (zie verder): ± € 6000 voor een workstation licentie en ± € 7000 voor een floating netwerk licentie. [16]

Solibri Model Viewer

Uiteraard zal niet elke betrokkene in het bouwproject beschikken over de Model Checker software. Solibri Model Viewer maakt het mogelijk de analyseresultaten in een intuïtieve omgeving te delen met anderen. De Viewer is vrij verkrijgbaar en beschikt over alle navigatie- en rapportagemogelijkheden van de Model Checker. Met dit softwarepakket hoeft dus niet elk teamlid een Model Checker licentie aan te schaffen. [16]

Solibri Issue Locator

De Issue Locator doet precies wat de naam zegt: dit programma laat toe analyseresultaten uit de Model Checker te lokaliseren in het oorspronkelijke CAD-model. Het zou immers vrij omslachtig zijn één specifiek element in een complex ontwerpmodel handmatig op te zoeken. Met de Issue Locator kan vanuit de CAD-omgeving eveneens een lijst van alle resultaten uit de Model Checker opgevraagd worden. Zo wordt het mogelijk deze in de vertrouwde ontwerpomgeving op te volgen en aan te passen. De Issue Locator plug-in bestaat voorlopig enkel voor ArchiCAD-software. In de toekomst zouden ook AutoDesk Architecture en AutoDesk Revit over een importer kunnen beschikken. [17]

Het is echter waarschijnlijker dat de Solibri Issue Locator vervangen zal worden door het *Open BIM Collaboration Format (BCF)*. Deze standaard vloeit voort uit de wens om communicatie tussen verschillende BIM-softwarepakketten mogelijk te maken. De BCF-standaard legt vast op welke manier gebruikers berichten en opmerkingen kunnen linken, zodat ze deze kunnen doorsturen naar anderen. De ontvanger kan die informatie dan in zijn eigen BIM-pakket gebruiken om gemakkelijk naar de desbetreffende plaats in het model te gaan.

De Open BCF-standaard is, zoals de naam zegt, open en kan in principe door alle geïnteresseerde softwareproducenten ondersteund worden. Op dit moment wordt BCF al ondersteund door onder meer Tekla. [17]

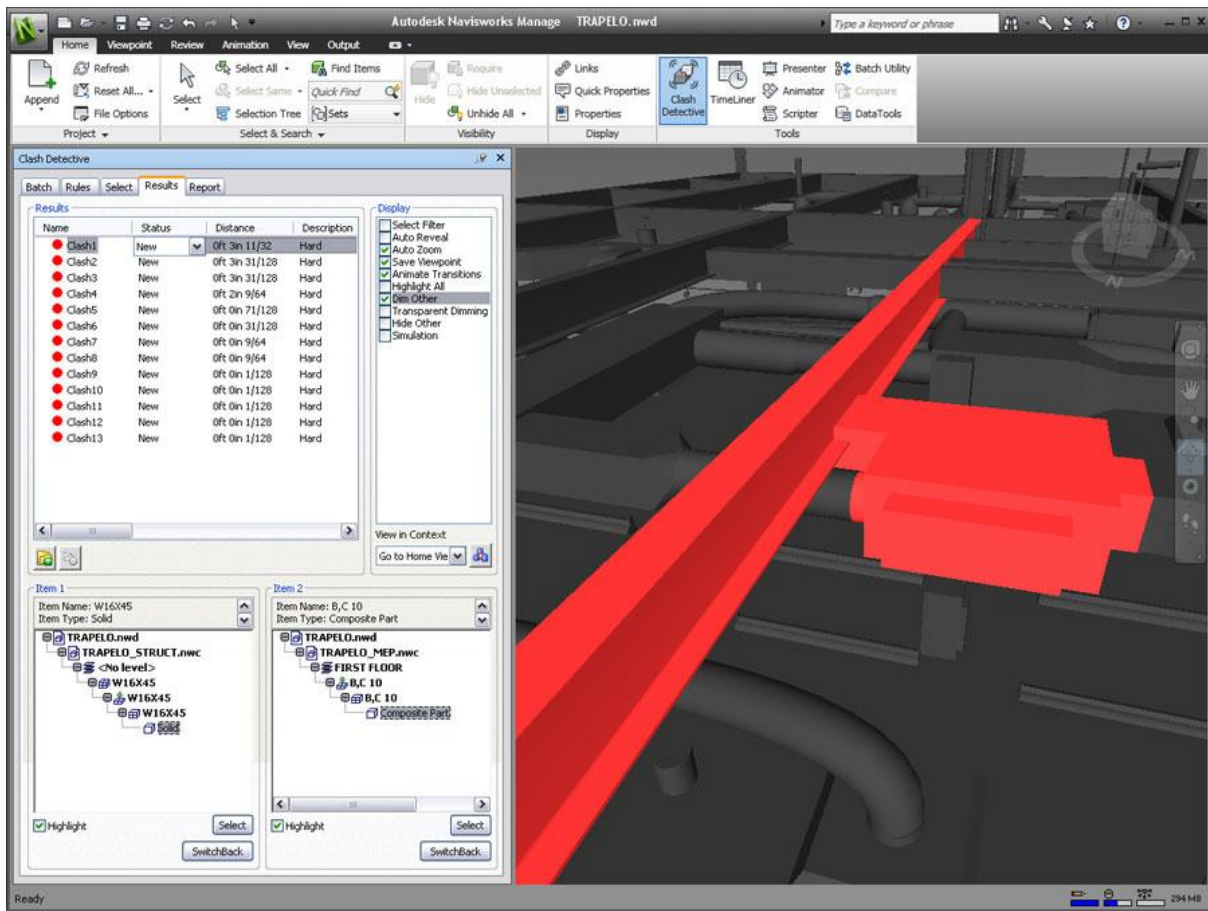
NAVISWORKS

Navisworks is de AutoDesk tegenhanger van de Solibri Model Checker, maar biedt nog heel wat bijkomende functionaliteiten. Eén van de grote sterktes van het pakket zijn de vele bestandsformaten die het ondersteunt. Dit is een direct gevolg van de historische achtergrond van het programma: tot enkele jaren geleden was het onafhankelijk van de grote CAD-softwareproducenten, waardoor het wel flexibel moest zijn. Op dit moment wordt het gros van de meest voorkomende bestandformaten door het pakket ondersteund, voor vele andere formaten zijn via een plug-in conversiemogelijkheden voorzien.

Het hele Navisworks pakket is opgesplitst volgens drie verschillende niveaus :

- *Navisworks Freedom*: Deze gratis viewing applicatie laat toe te bekijken wat in de meer geavanceerde pakketten is klaargezet. Het is eveneens mogelijk opmerkingen toe te voegen, bepaalde planningsinformatie te zien ...
- *Navisworks Simulate*: Op dit niveau zijn alle functionaliteiten van Navisworks beschikbaar, met uitzondering van clash detection. (Licentie: ± € 4000)
- *Navisworks Manage*: Dit is het meest uitgebreide pakket. Het bevat alle functionaliteiten, inclusief clash detection. (Licentie: ± € 9000)

De onderstaande **figuur** geeft een beeld van de Navisworks clash detection omgeving [18].



Figuur 18: De interface van Navisworks [18]

Als een model voor de eerste keer in Navisworks ingeladen wordt, zal op basis van de inputfile een NWC-bestand worden gecreëerd dat dienst doet als een cache-file. Dit zorgt ervoor dat de modelinformatie veel sneller kan worden opgehaald wanneer eenzelfde bronbestand opnieuw wordt geopend of toegevoegd ('append'). Daarbij is ervoor gezorgd dat, als er ondertussen wijzigingen zijn doorgevoerd in het bronbestand, het hele bronbestand opnieuw wordt ingelezen en de bestaande cache-file overschreven wordt. Een NWC-bestand is veel kleiner dan het originele modelbestand, omdat het softwarepakket enkel de voor Navisworks relevante informatie importeert. Volgens AutoDesk bedraagt deze reductie in bestandsgrootte tot 70 %, maar er zijn gevallen waar dit tot meer dan 85 % gaat. Hierbij is het belangrijk op te merken dat heel wat van de eigenschappen, die aan een object in de originele ontwerpsoftware zijn toegekend, mee geïmporteerd worden naar Navisworks.

DWG-bestanden afkomstig uit AutoDesk Civil 3D worden dynamisch gekoppeld aan het NWC-bestand dat Navisworks aanmaakt. Dit betekent dat, wanneer wijzigingen worden doorgevoerd in het Civil 3D-model, deze automatisch worden overgebracht naar de Navisworks-omgeving. Voor de meeste andere programma's – waaronder AutoDesk Revit – werkt deze dynamische link niet. Het gewijzigde model moet dan ook opnieuw geïmporteerd worden in Navisworks. [20]

Wanneer een model wordt ingeladen in Navisworks, worden naast een NWC-file ook een NWD- en een NWF-bestand gecreëerd. Deze laatste bestanden blijven eveneens veel kleiner dan de oorspronkelijke modelbestanden. Een NWD-bestand bevat alle geometrische data van het model, maar ook specifieke Navisworks-informatie (opmerkingen, clash detection resultaten enz.). Een NWD-bestand kan eigenlijk gezien worden als een soort foto of momentopname van het model. In tegenstelling tot een NWD-bestand bevat een NWF-file enkel links naar de originele bestanden, en dus geen geometrische informatie over de modellen. Naast deze verwijzingen worden ook Navisworks-specifieke gegevens bewaard. Telkens een Navisworks-bestand geopend wordt, zal door middel van de gelegde links de meest recente versie van het model worden ingeladen. Eventuele wijzigingen in de originele bestanden worden dus automatisch meegenomen naar de Navisworks-omgeving. [19] [20]

Naast het eenvoudigweg openen van bestanden in Navisworks, is het eveneens mogelijk extra bestanden toe te voegen aan de huidige Navisworks-omgeving ('append'). Wanneer dit niet via 'append', maar met 'merge' gebeurt, detecteert Navisworks automatisch of er geen informatie wordt ingeladen die al in het Navisworks-bestand aanwezig is. Daarnaast kan in bepaalde gevallen onmiddellijk een NWC-bestand worden aangemaakt vanuit de originele ontwerpomgeving. Deze exportmogelijkheid is beschikbaar voor onder meer AutoCAD, Microstation, Revit en ArchiCAD. Hierdoor is het risico op het verlies van intelligentie betreffende de modelgeometrie kleiner.

Navisworks biedt heel wat meer functionaliteiten dan Solibri. Het programma is immers opgevat als volledig geïntegreerd BIM-pakket en kan gebruikt worden om grote projecten te managen. Ontwerpmodellen worden in de Navisworks-omgeving ingeladen, waarna het mogelijk is een planning

op te stellen (4D), visualisaties te maken, materiaalhoeveelheden te berekenen (5D) enz. Een planning kan binnen Navisworks zelf opgesteld worden; het is eveneens mogelijk deze te importeren vanuit gespecialiseerde planningssoftware (bijvoorbeeld Primavera of MSPProject). Naast het planningsluik beschikt Navisworks ook over een groot aantal functies om het bestuderen van het model te vergemakkelijken. Zo is het mogelijk om op verschillende manieren door het model te 'wandelen'. Daarnaast kunnen op een relatief eenvoudige manier sneden gemaakt worden in het model, met de mogelijkheid deze te roteren in de drie dimensies. Verder kunnen gebruikers labels – met status – aan objecten hangen om erover te communiceren, wat uiteraard handig is voor het managen van raakvlakken. Het 'comment window' in Navisworks geeft dan een overzicht van alle toegevoegde opmerkingen.

Wat Navisworks echter tot een geschikt BIM-platform voor raakvlakkenbeheer maakt, is de mogelijkheid tot clash detection die in Navisworks Manage ingebouwd is. Deze module maakt het mogelijk automatisch geometrische raakvlakken in het model te controleren. De clash detection kan zowel in 3D als in 4D (inclusief tijdsaspect) uitgevoerd worden, waardoor in beperkte mate ook planningsfouten opgespoord kunnen worden. Het opnemen van het tijdsaspect bij het opsporen van raakvlakken vermijdt bv. dat objecten op hetzelfde moment op eenzelfde locatie staan.

Om aan te geven tussen welke objecten raakvlakken gecontroleerd moeten worden, is het voldoende deze elementen te selecteren en te groeperen in zogenaamde 'selection sets'. Dit groeperen kan nog worden vergemakkelijkt als in het oorspronkelijke CAD-pakket gewerkt is met een intelligente layerstructuur, waarin de opbouw van de objectenboom uit de SE-analyse verwerkt is.

In een volgende stap kan bepaald worden welke testen de geselecteerde objecten moeten ondergaan. Daarbij zijn onder meer de volgende mogelijkheden beschikbaar:

- *Hard*: Alle geselecteerde objecten die elkaar raken worden aangeduid als een probleem. Clashes worden in dit geval gedetecteerd als een negatieve afstand. Het instellen van de tolerantie ('tolerance') bepaalt meteen ook de ernst van de clash; hoe dichter de objecten bij elkaar

liggen ten opzichte van de ingestelde tolerantie, hoe meer negatief de gedetecteerde afstand en hoe ernstiger de clash.

- *Clearance*: Rond de objecten wordt een bufferzone voorzien, waardoor ook objecten die elkaar niet fysiek raken, gecontroleerd kunnen worden. Als objecten dichter bij elkaar liggen dan de ingestelde tolerantie toelaat, wordt dit gedetecteerd als een clash.
- *Duplicate*: Deze instelling maakt het mogelijk te controleren of er toevallig elementen dubbel in het ontwerp zijn opgenomen. De clash detection detecteert objecten van hetzelfde type die op exact dezelfde plaats staan.
- *Hard (Conservative)*: Aangezien 3D-objecten in Navisworks gemodelleerd worden door middel van afzonderlijke driehoeken, zou het kunnen dat een gewone 'hard' clash test bepaalde raakvlakproblemen niet detecteert (zie verder). Dit probleem doet zich voor wanneer de driehoeken van een object geen snijpunten hebben met deze van een ander element. Een 'hard (conservative)' clash test brengt alle *mogelijke* raakvlakproblemen in kaart, dus ook wanneer de driehoeken elkaar niet snijden. In tegenstelling tot een 'hard' clash test, *veronderstelt* Navisworks hiervoor dat dicht bij elkaar gelegen driehoeken interfereren. Dit betekent wel dat er ook raakvlakken kunnen gedetecteerd worden die er in werkelijkheid geen zijn.

Een clash detection kan uitgevoerd worden tussen vlakken, lijnen en punten. Navisworks beschouwt een 3D-solid dus als een verzameling vlakken, die opgebouwd zijn uit allemaal driehoeken (cfr. een triangulatie). Dit wordt in wat volgt verder uitgelegd. Raakvlakproblemen tussen objecten die zichzelf raken (bv. twee rioolbuizen in eenzelfde rioleringsstelsel) kunnen uiteraard genegeerd worden. Daarnaast is het mogelijk gelocaliseerde raakvlakproblemen op te slaan om ze achteraf opnieuw te bekijken. Ook de clash detection test zelf kan bewaard worden zodat, eens wijzigingen zijn aangebracht, dezelfde test doorlopen kan worden. Zo hoeft de hele clash detection niet opnieuw geprogrammeerd te worden en is het eenvoudig na te gaan of het probleem is opgelost. Naast de standaard ingestelde clash tests, is het ook mogelijk om als eindgebruiker zogenaamde 'rules' te definiëren en aan te passen. Deze 'rules' bepalen

welke elementen moeten genegeerd worden in een standaard ingestelde clash test. Zo is het mogelijk clashes tussen verschillende objecten, die in eenzelfde layer getekend zijn, niet op te nemen in de lijst van gedetecteerde raakvlakproblemen.

Navisworks biedt geavanceerde mogelijkheden om de gedetecteerde clashes te laten zien. Zo kunnen andere objecten van het model, die niets te maken hebben met een gedetecteerde clash, naar de achtergrond gebracht worden. Binnen de clash kan elk van beide objecten een andere kleur krijgen, zodat snel visueel duidelijk wordt waar de clash zich situeert. Navisworks zoomt ook automatisch in op de gedetecteerde clashes, zodat het niet nodig is handmatig de clash in het model te gaan opzoeken.

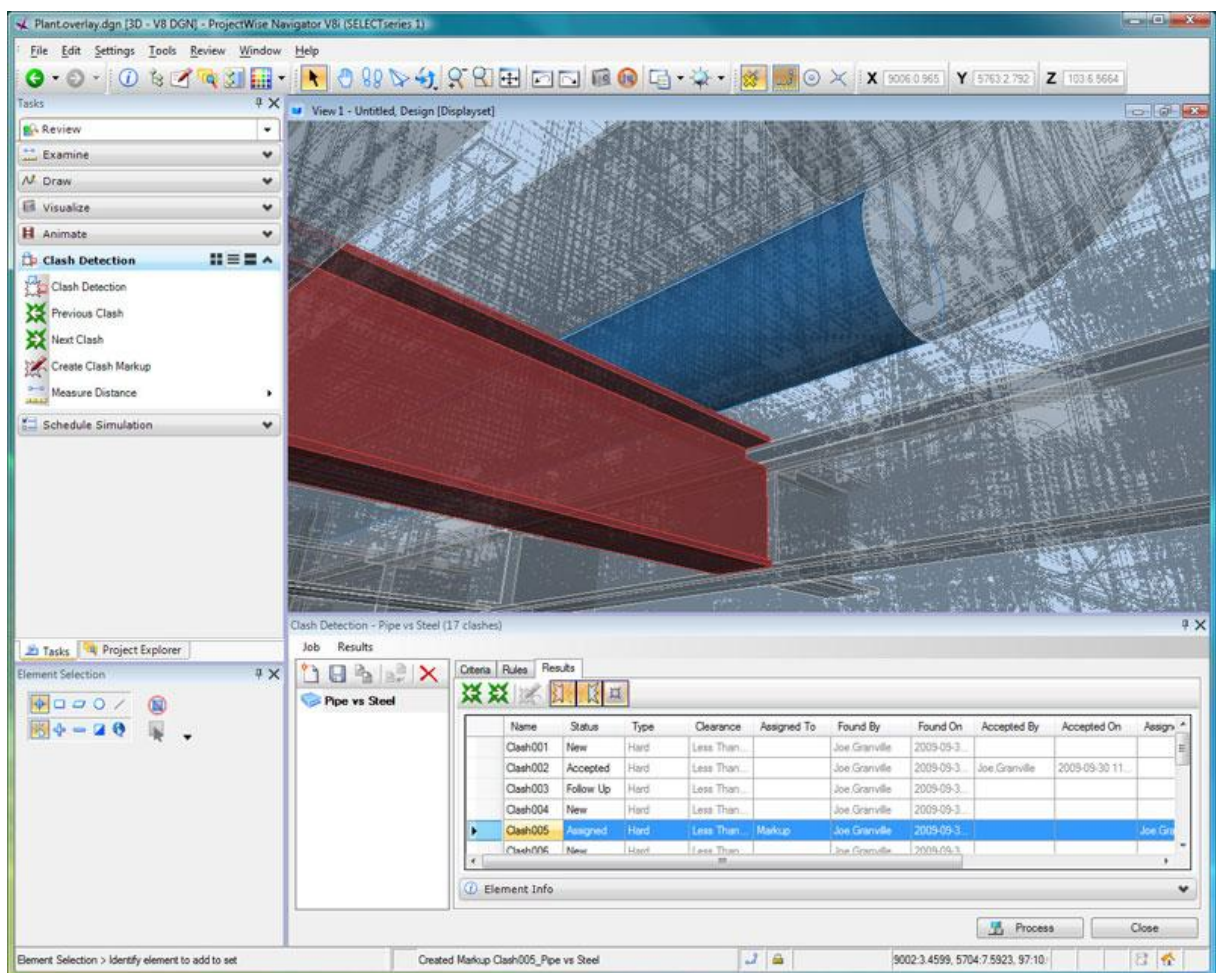
Op voorwaarde dat het geïmporteerde model automatisch kan worden ingeladen in Navisworks, kunnen de resultaten van de clash detection teruggekoppeld worden naar deze oorspronkelijke ontwerpomgeving. Deze mogelijkheid is niet beschikbaar voor modellen die via een plug-in zijn ingeladen. In bijvoorbeeld AutoCAD volstaat het om het commando 'nwload' in te geven, om vervolgens in het clash detection venster van Navisworks 'switchback' aan te klikken. Dit zorgt ervoor dat exact hetzelfde beeld als in Navisworks wordt getoond in de AutoCAD-omgeving, en ook daar onmiddellijk een gedetecteerde clash in het model kan worden opgespoord.

De resultaten van de clash detection kunnen naar verschillende bestandsformaten (o.a. XML, HTML ...) geëxporteerd worden in de vorm van een rapport. Dit rapport geeft, naast de namen van de objecten, ook de coördinaten van de clash weer.

[20]

NAVIGATOR

Ook Bentley heeft een softwareomgeving ontwikkeld waarmee ontwerpen gecontroleerd kunnen worden. De onderstaande **figuur** geeft een idee van de gebruikersinterface van het Projectwise Navigator pakket. In tegenstelling tot Navisworks is Projectwise Navigator (hierna gewoon Navigator genoemd) niet ontstaan uit een CAD-producent onafhankelijk programma, maar rechtstreeks ontwikkeld door Bentley. Dit betekent echter niet dat het aanbod van ondersteunende bestandsformaten gering is.



Figuur 19: De interface van Navigator [21]

Navigator is, net als Navisworks, opgevat als een geïntegreerd BIM-pakket. Het biedt dan ook de mogelijkheid planningen te importeren vanuit gespecialiseerde planningssoftware als Primavera en MSProject. Naast Navigator bevat het ProjectWise pakket van Bentley eveneens ProjectWise

Schedule Simulation, dat specifiek gericht is op het planningsaspect van projecten. Navisworks heeft in vergelijking met Navigator wel een streepje voor op het gebied van gebruiksgemak, omdat het ook een Gantt-view van de planning kan tonen.

Net als Navisworks maakt Navigator een nieuw bestand aan wanneer een CAD-model wordt ingeladen.⁴ Dit bestand wordt een 'overlay' file genoemd en fungeert als een containerbestand, waarin o.a. opmerkingen, die in Navigator zijn toegevoegd aan het model, opgeslagen worden. Op die manier wordt voorkomen dat het oorspronkelijke modelbestand gewijzigd wordt. Net als bij een Navisworks-bestand, wordt in een overlay file slechts de noodzakelijke informatie van het oorspronkelijke model overgenomen. Dit zorgt ervoor dat ook hier de bestandsgrootte aanzienlijk gereduceerd wordt. [22]

In Navigator kunnen onder meer bestanden van Microstation (Bentley) en AutoDesk ingeladen worden. Het spreekt echter voor zich dat Navigator vlotter overweg kan met modellen van Bentley dan met AutoDesk-bestanden. Het omgekeerde geldt natuurlijk ook voor Navisworks.

Net zoals in Navisworks kan de objectenboom van het project in Navigator gebracht worden door te werken met een intelligent lagensysteem voor de CAD-objecten. Ook is het mogelijk objecten te selecteren en toe te wijzen aan een bepaalde groep. De uiteindelijke clash detection kan dan op basis van deze indeling uitgevoerd worden. Uiteraard kan een gebruiker eveneens bepaalde objecten binnen een layer of groep selecteren om een individuele clash detection uit te voeren.

Objecten kunnen, net zoals in Navisworks, zowel tegen andere objecten als tegen zichzelf getest worden. Het is ook mogelijk om aan bepaalde objecten een 'clearance' toe te kennen, bv. om rekening te houden met de isolatie die rond een buis moet aangebracht worden.

⁴ Uitzondering hierop is het inladen van zogenaamde i-models (extensie I.DGN). Dit bestandsformaat gebruikt een optimaliseringsroutine waardoor er minder schijfruimte en geheugen gebruikt wordt. Een i-model is op zichzelf al een containerfile, waarin gegevens uit meerdere modellen samengevoegd kunnen worden.

Daarnaast is het mogelijk om maattoleranties op te nemen in de clash detection, of om bijkomende regels te definiëren.

Na het doorlopen van de test kunnen commentaren en gezichtspunten opgeslagen worden bij specifieke clashes. Net als Navisworks biedt Navigator ook verschillende mogelijkheden om de objecten waartussen de clash zit, visueel te onderscheiden van de rest van het model. Dit kan bijvoorbeeld door de twee objecten een andere kleur te geven of de andere objecten tijdelijk onzichtbaar te maken. Wat Navigator niet, of in mindere mate, ondersteunt, is het automatisch aanpassen van het zicht om een optimaal beeld op de clash te geven. In het beste geval wordt vanuit het huidige gezichtspunt ingezoomd op de clash; verplaatsen en roteren van het zicht gebeurt niet automatisch.

De huidige versie van Navigator laat niet toe een rapport van de uitgevoerde clash detection te genereren. Deze functionaliteit is in de volgende versie van het programma wel beschikbaar.

Daarbij komt nog dat de clash detection in Navigator niet gekoppeld kan worden aan de gemaakte planning. Problemen met planninggerelateerde raakvlakken kunnen met deze software dus niet gecontroleerd worden, maar voor het nagaan van zuiver geometrische raakvlakwesties vormt dit natuurlijk geen probleem.

[22]

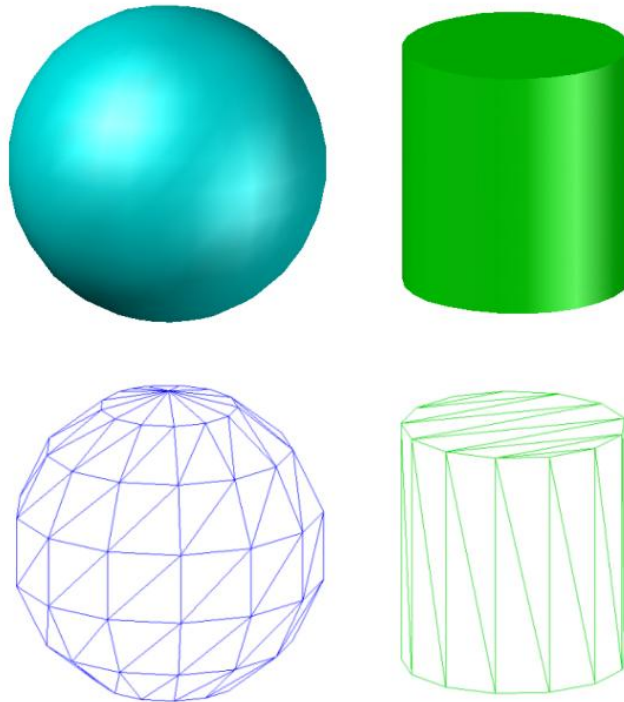
BEPERKINGEN VAN CLASH DETECTION SOFTWARE

De softwarepakketten die hierboven beschreven zijn, blijven niet meer dan hulpmiddelen om geometrische raakvlakken te beheren. In geen geval mag de output van de programma's beschouwd worden als een volledige voorstelling van alle problemen in het ontwerp. Daarnaast worden ook raakvlakproblemen gedetecteerd die er eigenlijk geen zijn. In deze paragraaf wordt op enkele valkuilen gewezen die ervoor kunnen zorgen dat ontwerpfouten niet naar voren komen in de clash detection.

Modellering in clash detection software

Autodesk Navisworks beschrijft geometrieën uit het ontwerp niet als oppervlakken, maar eerder als een collage van driehoeken, zoals ook

gebeurt bij een triangulatie. De onderstaande **figuur** illustreert hoe een 3D-volume, in AutoDesk 'solid' genoemd, in Navisworks is opgebouwd.



Figuur 20: 3D-solids in Navisworks

Bij de clash detection zal Navisworks nagaan waar punten, lijnen en driehoeken met elkaar interfereren, en deze interferenties aanduiden als 'clashes'. Dit betekent dat op plaatsen waar voorwerpen elkaar wel raken, zonder dat hun driehoeken overlappen, nooit clashes gedetecteerd zullen worden. Zo kan het gebeuren dat twee buizen niet helemaal parallel lopen aan elkaar en aan het einde lichtjes overlappen. Vermits de gekromde zijde van de buizen niet als rond, maar als afzonderlijke driehoeken wordt gemodelleerd, zou het kunnen dat Navisworks dit raakvlak niet detecteert. De gekromde zijden van de buizen snijden elkaar dan wel, maar de randen van de driehoeken niet.

Door een aangepast type te selecteren bij de clash detection ('hard conservative' in plaats van 'hard') kan dit euvel verholpen worden. Hierbij dient wel te worden opgemerkt dat er bij 'hard conservative' ook clashes worden gedetecteerd die in werkelijkheid geen raakvlakproblemen vormen.

Door middel van 'hard conservative' gaat Navisworks immers na welke objecten elkaar zouden *kunnen* snijden. [20]

Zowel Navisworks als Navigator is in staat om modellen, die gemaakt zijn in pakketten van Bentley en Autodesk, te openen. Modellen uit een Bentley-pakket zullen echter een meer doorgedreven conversie moeten ondergaan om in een Autodesk-programma als Navisworks te kunnen functioneren. Uiteraard geldt hetzelfde voor plannen uit een Autodesk-pakket die geïmporteerd worden in een programma van Bentley. Tijdens deze conversie zullen de 3D-geometrieën een deel van hun intelligentie verliezen en gereduceerd worden tot een eenvoudige mesh. Hierdoor kunnen 'fouten' geïntroduceerd worden in de modellen, waardoor extra clashes gevonden worden. Wanneer eenzelfde model wordt ingeladen in Navisworks of Navigator, zullen beide programma's dus niet noodzakelijk dezelfde clashes vinden. De clash detection software afkomstig van dezelfde producent als van de CAD-omgeving waarin het model oorspronkelijk getekend is, zal immers een minder doorgedreven conversie van het model moeten uitvoeren. Aangeraden wordt om ontwerptekeningen, geproduceerd met Autodesk-pakketten, met Navisworks te controleren en plannen, afkomstig van Bentley-producten, met Navigator.

Overige beperkingen

1. Omdat zowel Navisworks als Navigator objecten niet als 'vol' beschouwen, zullen ontwerpfouten, waarbij een voorwerp volledig omsloten is door een ander voorwerp, ook niet gedetecteerd worden. De onderstaande **figuur** toont een voorbeeld van een dergelijke fout waar een kleinere kolom volledig in een grotere zit. In dit geval zal noch Navigator, noch Navisworks clashes detecteren. Voor dit probleem is tot op heden geen bruikbare oplossing beschikbaar.



Figuur 21: Geneste kolom en clash detection

2. Een nodige voorwaarde om clashes tussen driedimensionale objecten te kunnen detecteren, is dat deze objecten in de CAD-omgeving ten minste uit tweedimensionale vlakken zijn opgebouwd. Als de objecten in het model voorgesteld zijn als zogenaamde 'polylines', kan de software immers niet bepalen tussen welke lijnen er in werkelijkheid vlakken zitten en waar er "leegtes" in het model aanwezig zijn.

De bovenstaande beperkingen tonen aan dat het automatisch opsporen van fouten in plannen op dit moment nog met de nodige omzichtigheid benaderd moet worden. Een visuele controle van de raakvlakken blijft van enorm belang. De clash detection software kan daarbij gebruikt worden als hulpmiddel, maar moet met een voldoende kritische ingesteldheid aangewend worden.

Een ander probleem is dat zowel Navisworks als Navigator veel raakvlakproblemen detecteert ten gevolge van imperfecties in de modellen, zonder dat dit ook werkelijk ontwerpfouten zijn. In uitgebreide modellen kan

dit aantal foutieve 'clashes' snel oplopen, zeker als de maattoleranties voor de clash detection te streng worden ingesteld.

3.2.3 Samenvatting

Door de structuur in 3D te ontwerpen wordt het mogelijk een 3D-BIM-omgeving op te zetten. Hoewel 3D-ontwerp een grotere investering in de beginfase van een project vraagt, biedt het grote voordelen tijdens het verdere projectverloop. Het samenbrengen van alle projectinformatie in een BIM-omgeving maakt het bijvoorbeeld mogelijk verschillende raakvlakcontroles te automatiseren.

Daarnaast kunnen in een BIM eisen, alternatieve oplossingen, communicatie over raakvlakken enz. grafisch aan het ontwerp gekoppeld worden.

Het continu up-to-date houden van een BIM-model is echter niet vanzelfsprekend. Op dit moment zou het al een hele verbetering zijn de verschillende modeldelen periodiek samen te voegen om o.a. raakvlakcontroles uit te voeren.

Eén van de grootste hinderpalen bij het samenbrengen van verschillende modellen is het gebrek aan onderlinge compatibiliteit van de software. Het producentonafhankelijke IFC-formaat tracht hiervoor een oplossing te bieden. Daarnaast kan de manier waarop modellen gedefinieerd zijn in een softwareomgeving, er voor zorgen dat bepaalde clashes niet gedetecteerd worden.

De onderstaande tabel geeft tot slot een overzicht van de belangrijkste functies – met betrekking tot raakvlakkenbeheer– van de besproken BIM-software.

	Solibri Model Checker	Navisworks	ProjectWise Navigator
Rule-based clash detection	✓	✓	✓
Mogelijkheid tot het opslaan van clash detections om deze later opnieuw te kunnen doorlopen	✓	✓	✓
Terugkoppeling ingevoegde opmerkingen naar CAD omgeving	±	✗	±
Automatische rapport generatie op basis van clash detection resultaten.	✗	✓	✗
Mogelijkheid tot het bekijken van clash detection resultaten in de oorspronkelijke CAD omgeving	±	✓	±
Importeren van een planning	✗	✓	±
Clash detection op basis van gekoppelde planning	✗	✓	✗
Comprimeren van oorspronkelijke bestandsformaten bij importeren	✓	✓	✓

3.3 Geografische Informatiesystemen

GIS staat voor Geografisch Informatiesysteem en is een softwaretool om geografische gegevens op te slaan, te beheren, te analyseren en te visualiseren. Dit laatste gebeurt door middel van kaarten die de grafische output van een GIS-systeem vormen. Eigenlijk is een GIS niet meer of niet minder dan een grafische databank, waarin gegevens op basis van hun locatie aan elkaar gerelateerd worden. In tegenstelling tot andere databeheersystemen kunnen relaties ook op een visuele manier voorgesteld worden. [23]

Zoals hierboven al uitgelegd is, kan de informatie over de te bouwen infrastructuur verzameld worden onder de vorm van een Bouw Informatie Model. Het betreft bouwtechnische eigenschappen van de verschillende bouwdelen, tijd - en kosten gerelateerde gegevens, de eisen die gesteld zijn aan de objecten enz. De infrastructuur moet bovendien ingepast worden in de bestaande omgeving, wat impliceert dat ook rekening moet gehouden worden met de ondergrond, de beschikbare vrije ruimte ... Dergelijke geografische informatie is beschikbaar in een GIS. Het lijkt dus voor de hand liggend om een koppeling te maken tussen het ontwerpmodel en zijn geografische locatie in een GIS. Op die manier functioneert de GIS-omgeving eigenlijk als een BIM. [23]

De informatie die hoort bij een bouwdeel (bijvoorbeeld welke eisen eraan gesteld zijn, of het object een raakvlak heeft e.d.) bevindt zich meestal in een tekstuele databank zoals Relatics. Indien een gebruiker of ontwerper informatie wil opvragen over een object, moet hij dus weten hoe dat object gedefinieerd is in de databank om het te kunnen terugvinden. Vervolgens kan hij de desbetreffende ontwerptekeningen en databankgegevens naast elkaar leggen. Idealiter zou bij een visuele voorstelling van het object de bijbehorende databankinformatie onmiddellijk getoond moeten worden. Omgekeerd kan een gebruiker in Relatics wel kijken welke gegevens bij een bepaald constructiedeel horen, maar is het niet onmiddellijk zichtbaar om welk object het gaat.

Een mogelijke oplossing zou zijn dat objecten in een databank zichtbaar gemaakt worden door middel van een viewer. Eén van de voordelen van een viewer is dat niet elke gebruiker van de databank hoeft te beschikken

over CAD-software om de ontwerptekeningen te zien. Zoals al aangehaald in [hoofdstuk 2 – Literatuurstudie](#) kan in Relatics enkel tekstuele informatie ontsloten worden, waardoor deze methode (nog) niet bruikbaar is voor dit databeheerssysteem. Sommige databanken beschikken echter over een ‘viewing technologie’, zoals de software aangeboden door BIWTechnologies (zie [hoofdstuk 2 – Literatuurstudie](#)).

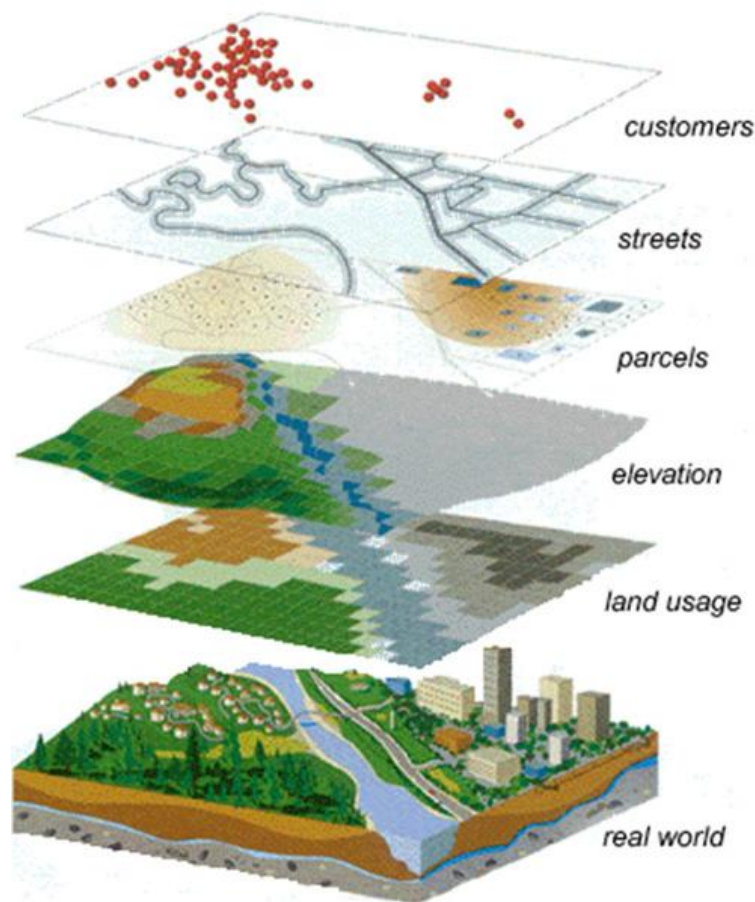
In Relatics is het wel mogelijk informatie te exporteren om ze vervolgens te koppelen aan de visuele voorstelling van de objecten. Het meest voor de hand liggend is de informatie te koppelen aan de CAD-tekening van de bouwdeelen, met de mogelijkheid om vervolgens zowel de informatie als het bouwdeel toe te voegen aan het globale BIM. De databankgegevens zouden echter ook onmiddellijk gekoppeld kunnen worden aan het centrale BIM-systeem, bijvoorbeeld een GIS-omgeving. In wat volgt, wordt kort uitgelegd hoe een GIS is opgebouwd, om vervolgens dieper in te gaan op de koppeling met CAD-tekeningen en databankgegevens.

3.3.1 Werking van een GIS

Klassiek wordt een GIS-omgeving in de bouwsector hooguit gebruikt om oppervlakte- en inplantingsgegevens op te vragen. De opbouw van een GIS maakt het beheer en het opvragen van dergelijke informatie vrij eenvoudig. Bovendien zijn verschillende GIS-systemen gratis online raadpleegbaar. In Vlaanderen geldt het Agentschap voor Geografische Informatie Vlaanderen (AGIV) als de referentie voor het opvragen van geografische gegevens.⁵

Een traditionele GIS-omgeving werkt met tweedimensionale kaarten. De geografische data worden in een GIS weergegeven in verschillende lagen (layers), die over elkaar gelegd worden (zie [figuur 22](#)). Deze layers bevatten elk een ander stukje informatie, waardoor het mogelijk wordt selectief informatie in de databank in te laden en te visualiseren.

⁵ <http://www.agiv.be/gis/>



Figuur 22: Lagenprincipe van een GIS-omgeving [24]

De basis van het lagenmodel bestaat uit een onderlegger, bijvoorbeeld een Digital Terrain Model (DTM). Een DTM, ook gekend als Digital Elevation Model (DEM), geeft een grafische neerslag van het aardoppervlak. Het is als het ware een verzameling punten op het aardoppervlak waarvan zowel de plaats- als de hoogtecoördinaten gekend zijn, en waarover een driedimensionaal vlak is getrokken.

Bovenop de onderlegger kunnen de layers met de gewenste informatie gelegd worden. Dit kan bv. een kaartje met de landgrenzen zijn, of een visuele voorstelling van de bevolkingsdichtheid in verscheidene werelddelen. Een gebruiker van de GIS-omgeving kan de gewenste informatie zichtbaar maken door de desbetreffende layers te selecteren. Bovendien is het mogelijk GIS-gegevens te analyseren. Zo kan worden nagegaan of twee geografische entiteiten met elkaar interfereren, hoe ver ze van elkaar liggen, wat het risico op bijvoorbeeld overstromingen is enz.

Het gaat dus niet enkel om de positionering van geografische entiteiten, maar ook om de onderlinge ruimtelijke relaties.

Daarnaast kan een GIS ook gebruikt worden om een Location Breakdown Structure (LBS, zie § 2.2 – [Systems Engineering](#)) te visualiseren. Zo kan voor objecten die op dezelfde plaats staan nagegaan worden dat deze niet op hetzelfde moment zullen worden uitgevoerd. Dit zijn weliswaar eerder planningskwesaties dan geometrische raakvlakken, maar ze dienen zeker ook beheerst te worden.

Bij het opzetten van een GIS-omgeving is het noodzakelijk over voldoende aanvullende gegevens, zoals een DTM, te beschikken. In Vlaanderen is al een – weliswaar beperkt – terreinmodel beschikbaar dat kan worden ingeladen in een GIS. Vanzelfsprekend is het daarbij belangrijk af te bakenen welk gebied nodig is voor een specifiek project; enkel die terreininformatie is relevant in de GIS-omgeving. Voor een overheidsopdracht kan een opdrachtnemer steeds de nodige informatie opvragen bij de opdrachtgever, maar deze terreinmodellen zijn, zoals al vermeld, vrij beperkt. Dit impliceert dat de nodige basisinformatie om de GIS-omgeving op te zetten, niet altijd even gemakkelijk verkrijgbaar is.

3.3.2 Koppeling met CAD-tekeningen en databankgegevens

Zoals al aangehaald werkt een GIS als een tweedimensionale omgeving. De meeste GIS-producenten hebben echter recentelijk driedimensionale GIS-software ontwikkeld. Daardoor wordt het mogelijk 3D-modellen in te laden in de GIS-omgeving. Aangezien het om een nieuwe evolutie gaat, staat deze technologie nog niet helemaal op punt.

Vele producenten poneren een driedimensionale GIS-omgeving te hebben ontwikkeld, maar daarbij dient te worden nagegaan of de omgeving 2.5D of 3D werkt. Vaak gaat het immers om 2D-oppervlakten waar een hoogte aan wordt toegekend. Op die manier ontstaat inderdaad een 3D-omgeving, maar deze is opgebouwd als een verzameling 'blokjes'. Dit legt niet enkel beperkingen op aan de vorm van de modellen die ingevoerd kan worden, maar ook aan het detailniveau dat bereikt kan worden. Zoals hierboven uitgelegd, is het aangewezen met 3D-modellen te werken om tot een correct raakvlakkenbeheer te komen. Daarnaast dient te worden opgemerkt dat ingevoerde 3D-modellen in een GIS meestal niet gedetailleerd genoeg

zijn om bepaalde geometrische raakvlakken te beheren. Zo wordt het moeilijk te controleren of de sparing in een balk voor een toekomstige nutsleiding op de juiste plaats is aangebracht.

Aangezien een GIS-omgeving niet ontworpen is om als BIM te functioneren, is ook het uploaden van CAD-data in GIS niet altijd even eenvoudig. CAD-data bestaan niet alleen uit geometrische entiteiten (punten, lijnen, polygonen), maar ook uit niet-grafische data (tekst, tabellen ...). In een GIS zijn procedures vastgelegd om die verschillende soorten gegevens te importeren. Deze procedures verschillen volgens de gebruikte GIS-software, maar vaak moeten nog heel wat handmatige bewerkingen uitgevoerd worden om de CAD-data te kunnen uploaden. In de praktijk is het dus vrij omslachtig dit procedé voor elk object toe te passen. Naarmate de driedimensionale GIS-software verder zal evolueren, kunnen deze procedures wel vereenvoudigd worden door ze te automatiseren. Deze automatisering zou de implementatie van een GIS-omgeving als grafisch SPI sterk vereenvoudigen, waardoor het waarschijnlijk ook sneller in de praktijk gebruikt zal worden.

Het definiëren van een workflow voor het inladen van CAD-data, gepaard met een verdere ontwikkeling van 3D-GIS, zou het in de toekomst mogelijk maken een GIS-omgeving als BIM te gebruiken.

Eén van de voordelen van een GIS-omgeving is dat deze kan ontsloten worden als webapplicatie. Dit betekent dat de projectinformatie met alle betrokkenen in het bouwproces gedeeld kan worden in een intuïtieve webomgeving. Sommige GIS-producenten hebben ook een vereenvoudigde 'viewing' versie van hun GIS-software ontwikkeld. Het gaat dan om hetzelfde programma, maar met beperkte functionaliteiten. Daardoor hoeft niet elke betrokkene te leren werken met de meer geavanceerde GIS-software.

Door tekstuele informatie over bijvoorbeeld raakvlakken, eisen enz. weer te geven in de gedeelde GIS-omgeving, is iedereen meteen op de hoogte van alle projectgegevens. Op die manier wordt het beheren van raakvlakken sterk vereenvoudigd:

- Raakvlakken die al gedetecteerd zijn, verschijnen als tekstuele informatie bij de desbetreffende objecten. Op die manier weten de betrokken partijen dat er een raakvlak is dat beheerd moet worden. Daarbij dient te worden opgemerkt dat niet alleen het raakvlak, maar

ook de verantwoordelijke voor het beheer ervan moet worden benoemd.

- In de GIS-omgeving, al dan niet toegankelijk als webapplicatie, is het ook mogelijk opmerkingen toe te voegen. Wanneer een gebruiker van het systeem een raakvlak detecteert, kan hij dit als een opmerking kenbaar maken aan de andere teamleden. Uiteraard is het dan aan de projectbeheerder of raakvlakverantwoordelijke om deze opmerkingen op te volgen en de betrokken partijen ermee te confronteren.

In de GIS-omgeving kan ook gebruik gemaakt worden van een zoekfunctie, waarmee bijvoorbeeld alle objecten met een raakvlak getoond kunnen worden. Dergelijke handige extra's maken het beheer van het hele project veel gemakkelijker. Een grote tekortkoming van een GIS-omgeving is echter dat er geen clash detection functionaliteit in verwerkt is. In een GIS-omgeving is dus enkel een handmatig nazicht van raakvlakken mogelijk.

In de toekomst zou het leggen van een dynamische koppeling tussen CAD-data en de GIS-omgeving het hele procedé verder vereenvoudigen. Hiermee zou het immers mogelijk worden bepaalde opmerkingen terug te koppelen naar de desbetreffende CAD-tekeningen, waardoor een ontwerper deze onmiddellijk te zien krijgt in zijn vertrouwde omgeving. Anderzijds zullen wijzigingen in de CAD-tekening automatisch doorgevoerd worden naar het GIS-platform. Dit betekent dat de CAD-data slechts één keer moet worden geïmporteerd.

Uit het bovenstaande blijkt dat, naast de grafische ontwerptekeningen, ook tekstuele informatie kan ontsloten worden in een GIS. De projectgegevens die zich in een traditionele databank bevinden, worden daartoe gekoppeld aan de GIS-modellen.

In een eerste variant wordt de databank rechtstreeks gelinkt aan de GIS-omgeving, met als voordeel dat gebruikers de informatie uit de databank niet handmatig in het GIS moeten invoeren. In plaats daarvan haalt de GIS-omgeving, telkens wanneer erom gevraagd wordt, gegevens op uit de databank via de gelegde koppeling. Zo kan steeds met de laatste versie van de gegevens gewerkt worden. De meeste GIS-omgevingen laten toe een koppeling te leggen naar verschillende courante databanksystemen als

Oracle en SQL. De dynamische exportmogelijkheden van Relatics zijn echter onvoldoende ontwikkeld om dergelijke koppeling met een GIS-omgeving te leggen. Hierbij komt nog dat veel bedrijven niet met een GIS-omgeving werken, waardoor het koppelen van databankgegevens aan een GIS voorlopig weinig nut zou hebben. Een GIS-omgeving brengt immers vrij grote aanschafkosten met zich mee en heeft een relatief steile leercurve. Bovendien moet, als de databankgegevens rechtstreeks aan het GIS-model worden gelinkt, een ontwerper in ieder geval toegang hebben tot de GIS-omgeving. Het zou voor hem handiger zijn om de databankinformatie rechtstreeks in de CAD-omgeving te krijgen. De koppeling van gegevens aan het CAD-model kan als een tussenstap gezien worden om in de toekomst dit CAD-model te integreren in een GIS-omgeving. Zoals eerder aangehaald heeft dit als voordeel dat ontwerpers de databankgegevens vanuit hun vertrouwde ontwerpomgeving kunnen raadplegen.

Databankgegevens naar een CAD-omgeving brengen kan op verschillende manieren. Het is echter voor alle methoden nodig dat er een één-op-één relatie bestaat tussen de objecten in de databank en de objecten in het model. De meeste databanksystemen geven immers unieke identificatiecodes aan hun objecten. Deze informatie moet overgebracht worden naar de CAD-omgeving. Om verwarring te vermijden is het noodzakelijk dat de CAD-modellen een overeenkomstige codering krijgen. Op basis van deze codering is het mogelijk in het CAD-programma gegevens, die uit Relatics naar bv. Excel zijn geëxporteerd, in te laden. De gegevens die op dergelijke manier aan een object gekoppeld zijn, kunnen dan getoond worden via het eigenschappenvenster van dat object. Het probleem met deze werkwijze is dat de gegevens niet dynamisch worden bijgewerkt. Telkens wanneer er gegevens gewijzigd worden in de databank, moeten ze opnieuw geëxporteerd worden naar het outputbestand en ingeladen worden in de CAD-omgeving. Verder is een dergelijke methode ook zeer gevoelig aan inconsistenties: een object met een foutieve identificatiecode kan het exportproces danig in de war sturen. Een bijkomend probleem is dat veel CAD-pakketten niet objectgericht werken, maar met lijnen, oppervlakten en strings. Hierdoor moeten gegevens aan bijvoorbeeld een lijn gekoppeld worden. Dit kan leiden tot onduidelijkheid wanneer een object, zoals een ligger, bestaat uit meerdere lijnen. AutoCAD 3D zal geëxtrudeerde polylines echter behandelen als

solids, eenvoudig voor te stellen als driedimensionale 'blokjes'. Aan dergelijk 'blokje' kunnen op een meer eenduidige manier gegevens gekoppeld worden. Andere softwarepakketten werken onmiddellijk objectgericht. Zo zal Tekla het stalen profiel niet als een verzameling lijnen tekenen, maar als een driedimensionale vorm.

3.3.3 GIS-producenten

Op de markt zijn verschillende GIS-omgevingen beschikbaar. Eén van de bekendste spelers is zonder twijfel Google Earth. AutoCad voorziet zelfs een automatische exportmogelijkheid naar deze wijd verspreide GIS-omgeving. Naast Google Earth Professional zijn de belangrijkste professionele GIS-applicaties ArcGIS en GeoMedia. ArcGIS is de GIS-omgeving die door ESRI op de markt is gebracht, Intergraph heeft GeoMedia als GIS-platform. De 3D-mogelijkheden zijn bij beide producten nog in volle ontwikkeling. Momenteel staat ArcGIS iets verder op dit vlak. De investeringskost voor dergelijke GIS-omgeving, zonder 3D-extensie, bedraagt ongeveer € 2500 per licentie.

Ook de traditionele CAD-producenten bieden met AutoCad Map 3D en Bentley MAP reeds de mogelijkheid kaartmateriaal en oppervlaktegegevens aan het CAD-model te koppelen. Het gebruik van CAD-software heeft echter als nadeel dat de interoperabiliteit tussen de verschillende systemen vaak heel beperkt is (zie §3.2 – [Bouw Informatie Model](#)). Door een welbepaald CAD-programma als BIM te hanteren kunnen problemen ontstaan, in het bijzonder wanneer modellen uit andere softwarepakketten ingeladen moeten worden in het BIM. Een GIS-omgeving als BIM gebruiken verhelpt dit euvel.

3.3.4 Samenvatting

Een GIS-omgeving gebruiken als grafisch SPI zorgt voor een gemakkelijker projectbeheer. Alle betrokkenen weten waar de projectinformatie te vinden is, kunnen gemakkelijk opmerkingen geven enz. Gegevens en hun onderlinge relaties kunnen in een GIS op een visuele manier worden voorgesteld, wat toelaat een dergelijke omgeving als BIM te gebruiken. Bovendien is het mogelijk GIS-informatie ter beschikking te stellen van alle teamleden door de omgeving te ontsluiten als een webapplicatie. Dit betekent meteen ook dat er geen specifieke CAD-software nodig is om over geometrische raakvlakken te communiceren.

Een GIS-omgeving is echter (nog) onvoldoende ontwikkeld om tot een efficiënt raakvlakkenbeheer te komen. Eén van de problemen is dat de ingevoegde modellen vaak te weinig gedetailleerd zijn, zeker als de GIS-omgeving met 2.5D in plaats van met 3D werkt. Daarbij komt nog dat de koppeling van het GIS-systeem met CAD- en databankgegevens niet altijd even eenvoudig gelegd kan worden.

De onderstaande tabel geeft een samenvatting van de voor- en nadelen van een GIS-omgeving met het oog op een gemakkelijker project- en raakvlakkenbeheer.

Voordelen	Nadelen
GIS biedt een grafisch SPI en werkt onafhankelijk van de gebruikte CAD-software.	Door een GIS-omgeving als BIM te gebruiken moet een extra programma worden aangeleerd.
Het grafische model wordt op een gemakkelijke wijze aangevuld met tekstuele informatie (eisen, raakvlakken ...).	Het GIS is een extra software-applicatie met bijbehorende aanschafkosten.
In een GIS-omgeving kunnen opmerkingen ingevoegd worden. Deze opmerkingen zijn zichtbaar voor de hele projectgroep.	De implementatie van een GIS-omgeving vereist het beschikken over basisinformatie, bijvoorbeeld een DTM.

Voordelen	Nadelen
Objecten die raakvlakken hebben, kunnen gemakkelijk benoemd en uitgefilterd worden.	Het linken van CAD-modellen en databankinformatie aan een GIS is niet eenvoudig en vraagt extra tijd.
Een GIS beschikt over analysemogelijkheden om bijvoorbeeld grondvolumes te berekenen.	Modellen in een GIS zijn vaak niet gedetailleerd genoeg en dus minder geschikt voor het beheer van raakvlakken.
	Een GIS-omgeving beschikt niet over clash detection functionaliteiten.

3.4 Besluit

In de praktijk wordt tot nog toe voornamelijk met 2D-plannen gewerkt, zeker in de uitvoeringsfase. Toch heeft het werken met 3D-modellen belangrijke voordelen. Enerzijds zorgt de 3D-omgeving voor een intuïtief inzicht in het ontwerp, anderzijds maken 3D-objecten het mogelijk om geometrische raakvlakcontroles uit te voeren.

Deze automatische clash detection kan met behulp van een BIM uitgevoerd worden. Softwarepakketten als Solibri, Navisworks en Navigator bieden immers niet alleen de mogelijkheid verschillende CAD-modellen samen te brengen in één omgeving, maar beschikken ook in mindere of meerdere mate over deze clash detection functionaliteit.

In principe zou ook een GIS-omgeving als BIM gebruikt kunnen worden. Op dit moment is deze software echter onvoldoende ontwikkeld, zeker op het vlak van compatibiliteit met CAD-programma's. Bovendien is in een GIS hoogstens een handmatig nazicht van raakvlakken mogelijk.

In [hoofdstuk 4 – Raakvlakkenbeheer](#) wordt het opsporen van raakvlakproblemen met behulp van clash detection software geïntegreerd in een volledig beheerproces.

HOOFDSTUK

4 Raakvlakkenbeheer

Hoewel het bovenstaande hoofdstuk hulpmiddelen aanreikt om raakvlakproblemen in kaart te brengen met behulp van 3D-geometrie en clash detection software, volstaan deze hulpmiddelen niet om problemen met raakvlakken te vermijden. Het is belangrijk een systematiek te ontwikkelen om raakvlakken op te sporen en te beheren. Alleen zo krijgen ze de nodige aandacht in de loop van het hele project.

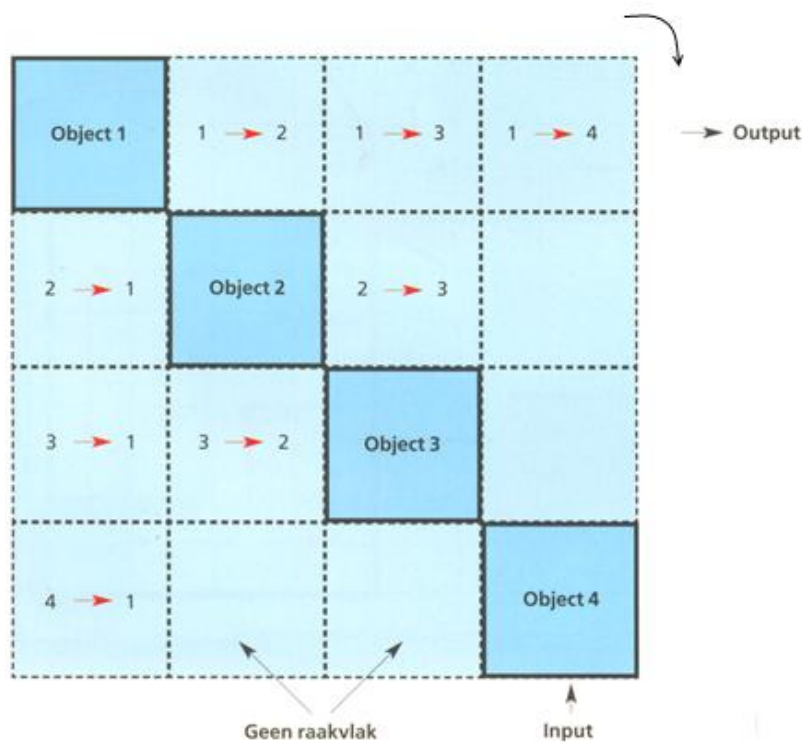
Dit hoofdstuk geeft in eerste instantie aan hoe de detectie van raakvlakken kan gebeuren aan de hand van de zogenaamde N²-chart. Dit diagram laat toe op een systematische wijze de interactie tussen de objecten op te sporen. De werkwijze, inclusief de voor- en nadelen, is uitvoerig besproken in §4.1 – N²-chart. Vervolgens wordt in §4.2 – Proces voor het beheer van geometrische raakvlakken uiteengezet welke elementen cruciaal zijn bij het beheer van raakvlakken. De uitgewerkte systematiek om tot een efficiënt raakvlakmanagement te komen is overzichtelijk weergegeven in een flowchart. Ten slotte wordt in §4.3 – Implementatie van het proces in Relatics geïllustreerd hoe het beheerproces in de praktijk kan worden toegepast in de relationele databankomgeving Relatics.

4.1 N²-chart

Binnen de methodiek van Systems Engineering wordt vaak gebruik gemaakt van een N²-diagram (ook wel N²-chart genoemd) om raakvlakken op te sporen en in kaart te brengen. Dit diagram is opgebouwd als een matrix met dimensie N. Op de hoofdiagonaal van de matrix staan de N verschillende objecten waartussen één of meerdere raakvlakken kunnen bestaan. De overige cellen worden gebruikt om deze raakvlakken aan te duiden. Wanneer één van de cellen niet ingevuld wordt, betekent dit dat er geen raakvlak bestaat tussen de desbetreffende objecten. Het aanduiden

van de raakvlakken kan op verschillende manieren gebeuren: met een kruisje of tekst in de gepaste cel, of door een pijl of lijn tussen de twee betreffende objecten op de hoofddiagonaal. Eén van de grote voordelen van de N^2 -chart is dat de opsteller ervan verplicht wordt systematisch na te denken over elke mogelijke relatie tussen alle objecten van het systeem.

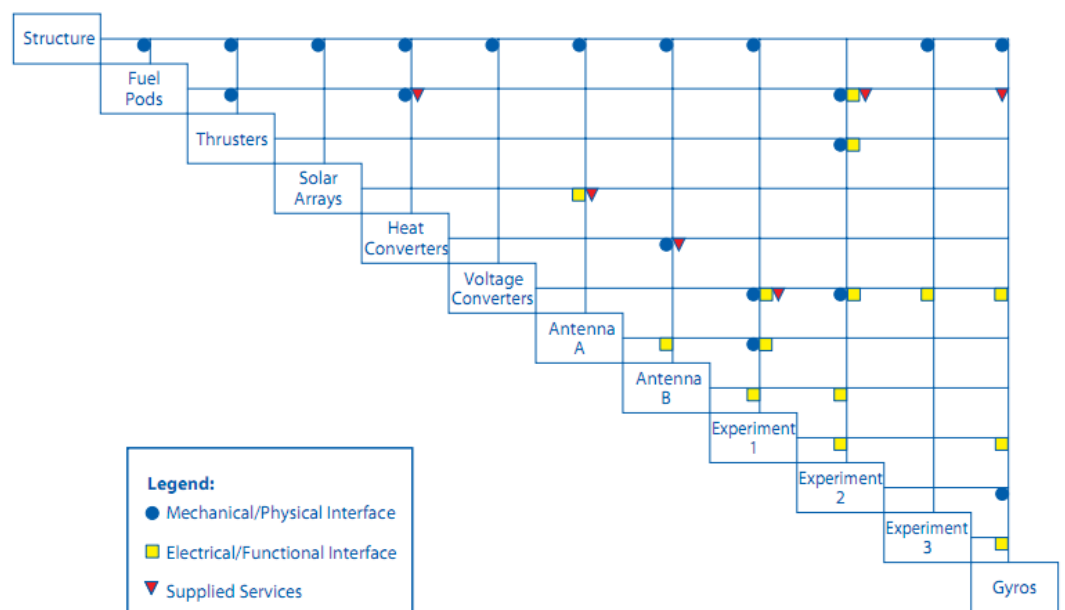
Om een overvloed aan raakvlakken te vermijden, is het aangewezen enkel de risicovolle in de N^2 -matrix op te nemen. Indien gedetecteerde raakvlakken als niet-risicovol bevonden worden, is het onnodig deze expliciet te beheersen. Er worden dan ook in het databeheersysteem geen verdere acties aan dit raakvlak gekoppeld. De beoordeling of een raakvlak voldoende risicovol is, kan in vele gevallen gebeuren op basis van ervaring. In geval van twijfel kan een risicoanalyse uitgevoerd worden. Geheel in de filosofie van Systems Engineering is het wel aangewezen vast te leggen dat het niet-risicovol raakvlak ooit is gedetecteerd. Dit kan bijvoorbeeld door het op te nemen in het databeheersysteem van het project (zoals Relatics). Daarbij kan het nuttig zijn eveneens te vermelden waarom het raakvlak niet verder wordt opgevolgd.



Figuur 23: N^2 -chart [4]

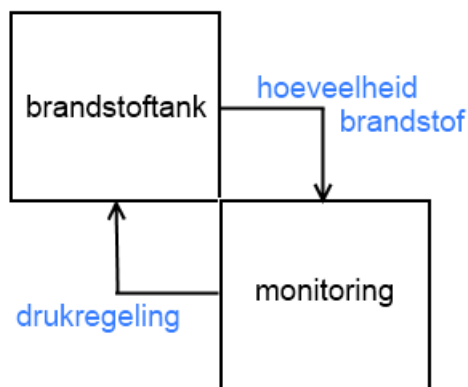
Vermits in een N^2 -chart enkel moet worden aangegeven of er één of meerdere raakvlakken bestaan tussen objecten, is het in principe niet nodig de volledige matrixstructuur uit bovenstaande **figuur** te gebruiken. Een ‘halve’ matrix, waarbij enkel de cellen van de hoofdiagonaal en deze erboven worden gebruikt, volstaat. Ook de richting van eventuele pijlen is in het kader van raakvlakkenbeheer overbodige informatie. Deze pijlen geven in het algemeen de richting van de informatiestroom aan, en bepalen dus welk object input levert aan een ander object. In het geval van raakvlakken kan deze richting aangeven welk object een ‘uitgaand raakvlak’ heeft met een ander object. Het onderscheid tussen een ‘uitgaand’ en ‘inkomend’ raakvlak kan echter niet onmiddellijk bepaald worden. De volgende paragraaf geeft een methode aan om op een consequente en objectieve manier te bepalen welke objecten ‘leidend’ zijn en dus uitgaande raakvlakken hebben.

Bij de aanduiding van de raakvlakken kan ook een onderscheid gemaakt worden tussen de verschillende types raakvlakken (geometrische, tijdsgebonden ...) door symbolen mee te geven aan de pijlen tussen de desbetreffende objecten. Onderstaande **figuur** toont een voorbeeld van een N^2 -chart uit het *NASA Systems Engineering Handbook*, waarbij dergelijke symboliek aan de pijlen is toegekend [25].



Figuur 24: N^2 -chart met aanvullende informatie bij de pijlen [25]

In voorbeelden uit de literatuur worden dikwijls niet alleen objecten, maar ook functies in de N^2 -chart opgenomen. Dit gebeurt vaak bij de analyse van softwaresystemen of productielijnen. Zo ‘vertelt’ een voorraadtank voor brandstof aan het monitoringssysteem hoeveel brandstof nog in de tank aanwezig is. Anderzijds regelt dit systeem de druk in de brandstoftank. In dit geval is het wel nuttig een vierkante matrix te definiëren. In de cellen boven de hoofddiagonaal staan alle raakvlakken die input leveren aan de onderliggende elementen op de hoofddiagonaal. De cellen onder de hoofddiagonaal kunnen gebruikt worden om aan te geven welke data deze onderliggende elementen terugsturen naar een vorig functioneel of fysiek object. In dit geval wordt dus gekozen voor een rotatie in wijzerzin, in de praktijk kan echter ook een omgekeerde conventie aangehouden worden. Deze conventie wordt aangeduid door middel van een pijltje aan de rechterbovenhoek van de N^2 -chart. Wanneer een item op de hoofddiagonaal zowel informatie ontvangt van, als terugzendt naar een ander item, ontstaat een zogenaamde ‘feedback-lus’, zoals ook voorgesteld is op de volgende [figuur](#). [25] [26]



Figuur 25: N^2 -chart met feedbacklus

In bouwkundige toepassingen, en zeker met het oog op de detectie van geometrische raakvlakken, spreekt het voor zich enkel fysieke objecten van het systeem op de hoofddiagonaal te plaatsen. De functies van het systeem worden dus niet opgenomen in de N^2 -chart.

4.1.1 *Bepaling van de kritische objecten*

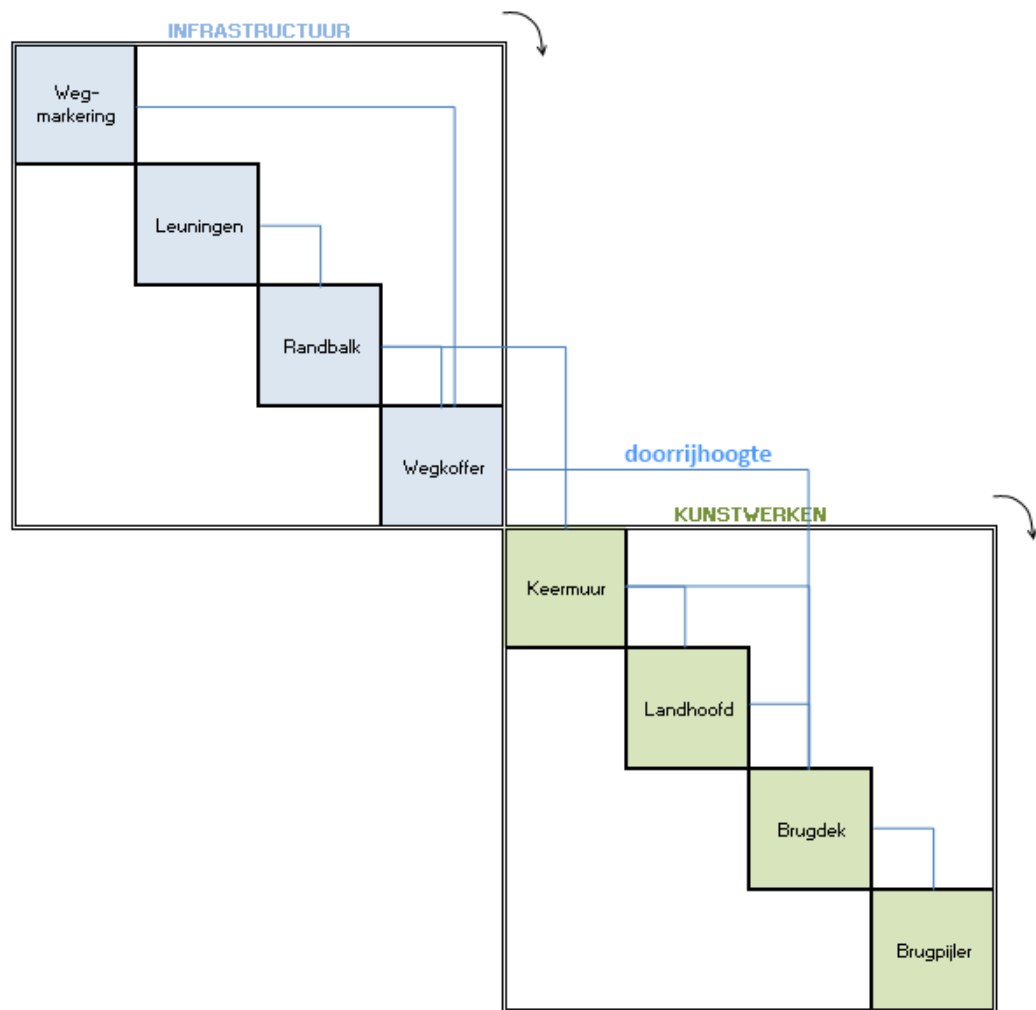
De N²-chart is een handig hulpmiddel om te bepalen welke objecten leidend zijn voor het te beheersen raakvlak. Daarnaast is het mogelijk een overzicht te krijgen van alle raakvlakken en zo de kritische objecten vast te leggen. Een kritisch object is een object dat heel wat raakvlakken met andere objecten heeft, en het is dan ook aangewezen dit object als 'leidend' aan te houden. Dit betekent dat de andere objecten die een raakvlak hebben met het kritische object, eventuele aanpassingen aan het ontwerp van dit laatste moeten volgen. Deze objecten worden 'rakende' objecten genoemd.

De bepaling van de kritische en leidende objecten aan de hand van de N²-chart kan het makkelijkst uitgelegd worden door middel van het volgende voorbeeld.

Veronderstel een raakvlak 'doorrijhoogte', waarbij bepaald wordt dat er voldoende vrije ruimte moet zijn tussen een kunstwerk (discipline KUNSTWERKEN) en het onderliggende wegvak (discipline INFRASTRUCTUUR). Dan is het voor alle betrokken partijen handig te weten welk object zich moet 'aanpassen' aan het andere, en dus als 'rakend' benoemd kan worden. Wanneer het kunstwerk een rakend object is, zal bij het ontwerp ervan de hoogteligging van de wegverharding moeten gerespecteerd worden. De weg wordt dan beschouwd als 'leidend' object. Indien de hoogteligging van de weg om een of andere reden achteraf moet aangepast worden, zal het ontwerp van het kunstwerk deze aanpassing moeten volgen. Het onderscheid tussen 'leidende' en 'rakende' objecten wordt dus vooral gebruikt om tot eenduidige afspraken te komen.

De weg en het kunstwerk bestaan elk uit heel wat objecten, die zowel onderling als interdisciplinair raakvlakken kunnen hebben met elkaar. N²-matrices kunnen dan ook opgesteld worden voor verschillende decompositieniveaus. Enkele van de objecten die deel uitmaken van de weg of het kunstwerk zijn opgenomen in de onderstaande N²-chart. Hierbij dient te worden vermeld dat deze zeker niet volledig is en louter als voorbeeld dient.⁶

⁶ De objecten in de N²-chart maken deel uit van een praktische case die is uitgewerkt in hoofdstuk 5 (§5.4 – Case 3 – Randbalk).



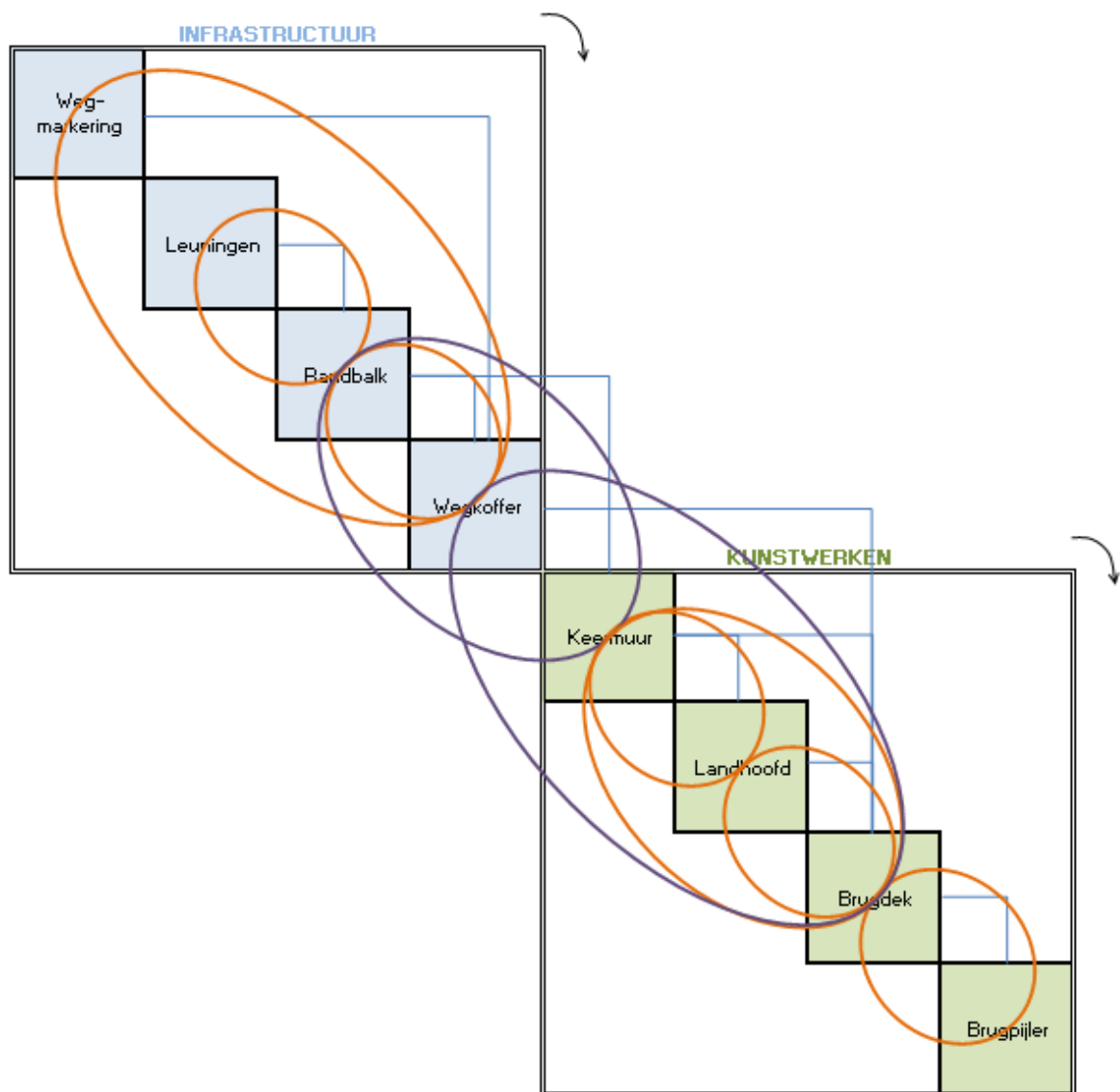
Figuur 26: Voorbeeld van een N²-chart

Zoals hierboven reeds vermeld, wordt de gekozen rotatiezin aangeduid door middel van de pijltjes in de rechterbovenhoek. Aangezien in het kader van raakvlakkenbeheer de richting van de informatiestromen weinig betekenis heeft, zouden deze pijltjes ook weggelaten kunnen worden.

Een gemakkelijke werkwijze bestaat erin de N²-matrices op te stellen per discipline. Zo is meteen duidelijk te zien welke raakvlakken de grenzen van één discipline overschrijden. Dit is belangrijk omdat interdisciplinaire raakvlakken over het algemeen moeilijker te beheersen zijn, aangezien er meestal langere communicatielijnen mee gepaard gaan. Het eerder

aangehaalde voorbeeld van de te respecteren doorrijhoogte is zo'n interdisciplinair raakvlak, zoals ook te zien is in de bovenstaande **N²-chart**.

Om te bepalen welke objecten kritisch zijn, kunnen zogenaamde controlelussen gebruikt worden. Daarbij worden cirkels getrokken tussen de objecten die elkaar onderling beïnvloeden. Objecten waar veel controlelussen op toekomen of uit vertrekken, hebben heel wat raakvlakken en kunnen dan ook als kritisch beschouwd worden. Zoals hierboven uitgelegd, is het dan ook aangewezen deze objecten leidend te maken. De bovenstaande N²-chart wordt hieronder hernomen met de aanduiding van de controlelussen. De paarse lussen duiden de interdisciplinaire raakvlakken aan, de oranje geven de intradisciplinaire weer.



Figuur 27: Voorbeeld van een N²-chart met controlelussen

Hoewel in het bovenstaande diagram slechts enkele objecten zijn opgenomen, blijkt dat de N²-chart al snel onoverzichtelijk wordt. Bovendien vraagt het opstellen ervan heel wat tijd en energie. Het is dan ook uitermate belangrijk op te merken dat deze matrices enkel gebruikt worden in geval van onduidelijkheid bij de benoeming van de leidende objecten. In de meeste gevallen kan immers intuïtief bepaald worden welke objecten het best als leidend beschouwd worden. Eens die beslissing genomen is, wordt iedereen geacht deze afspraak te respecteren.

Zoals hierboven al vermeld, kan vooraf bepaald worden welke discipline leidend is ten opzichte van de andere disciplines. Zo kan beslist worden dat INFRASTRUCTUUR steeds als leidend wordt genomen, vermits de wegen bij een infrastructuurontwerp letterlijk met alle andere disciplines interfereren.

Voor het interdisciplinaire raakvlak 'doorrijhoogte' tussen brugdek en wegkoffer (paarse controlelus) zal het ontwerp van de brug zich dan ook moeten aanpassen aan de hoogteligging van de wegverharding, die deel uitmaakt van de wegkoffer. De N²-chart toont aan dat het brugdek alleszins een kritisch object is, vermits hier maar liefst vier controlelussen op toekomen. Binnen de discipline van de KUNSTWERKEN zal het brugdek dan ook als leidend worden gekozen. Voor het raakvlak tussen de keermuur en het landhoofd kan de keermuur als leidend worden gekozen op basis van het aantal controlelussen door het object. Dit betekent echter dat het ontwerp van het landhoofd het ontwerp van zowel keermuur als brugdek moet volgen. Dit is echter niet te vermijden: in een complex project zullen steeds objecten bestaan die de ontwerpplannen van meerdere andere modellen moeten volgen. In deze situatie is het wel logischer om het landhoofd toch als leidend te beschouwen.

In conflictsituaties kan de volgende vuistregel in acht worden genomen: als een object binnen één discipline meestal leidend is, is het aangewezen om voor alle raakvlakken binnen die discipline dit object als leidend aan te houden.

Voor de discipline INFRASTRUCTUUR lijkt de situatie wat moeilijker: er zijn immers twee objecten waar drie controlelussen op toekomen. Intuïtief kan echter worden ingezien dat het weinig logisch is de randbalk als leidend te beschouwen boven de elementen waaraan deze balk raakt. Voor alle raakvlakken binnen deze discipline waar de wegkoffer deel van uitmaakt, zal er dan ook geopteerd worden deze wegkoffer als leidend te beschouwen.

In de literatuur wordt vaak aangehaald dat, om onoverzichtelijke en overladen N²-charts te vermijden, het N²-diagram best voor een hoger decompositieniveau opgesteld wordt. Hoe verder het systeem immers gedeclineerd wordt, hoe meer objecten ontstaan en hoe meer raakvlakken tussen deze objecten tevoorschijn komen. Vanuit het oogpunt van Systems Engineering lijkt het dan ook logisch het systeem niet te ver te decomponeren. In de praktijk moet de decompositie echter worden

doorgevoerd tot het niveau waarop het hele systeem duidelijk kan worden beheerd. Voor de N²-chart mag daarbij niet worden teruggegaan naar een hoger decompositieniveau, vermits daarbij informatie verloren gaat en mogelijk zeer kritische raakvlakken niet gedetecteerd zullen worden. Het vergt dan ook enige ervaring om een N²-chart op te stellen, en het juiste decompositieniveau te bepalen zodat er geen cruciale raakvlakken uit het oog worden verloren.

[25] [26] [27]

4.2

Proces voor het beheer van geometrische raakvlakken

Zoals al meermaals aangehaald, is het belangrijk het managen van raakvlakken te benaderen als een volledig geïntegreerd proces. Naast de controle van het ontwerp is het essentieel dat raakvlakken in de loop van het project worden beheerd. Daarbij worden deze raakvlakken in een zo vroeg mogelijk stadium opgespoord, om er rekening mee te kunnen houden van bij de opmaak van de ontwerpplannen. Het komt er met andere woorden niet enkel op aan de 'fouten' in een ontwerp op te sporen om problemen tijdens de uitvoering te vermijden. Uiteraard blijft deze controle van het ontwerp absoluut noodzakelijk, onder meer omwille van de onderstaande redenen:

- Eventueel laattijdig doorgevoerde wijzigingen van een ontwerp kunnen alsnog conflicten met andere ontwerpplannen veroorzaken. Deze laatste plannen moeten worden aangepast, maar een kleine communicatiefout kan er al voor zorgen dat dit niet meer gebeurt.
- Voor de raakvlakken die op voorhand in kaart zijn gebracht, wordt tevens bepaald aan welke eisen deze moeten voldoen. Daarnaast wordt vastgelegd welke acties nodig zijn om het raakvlak te beheren (zie verder). Een voorbeeld van een actie is het controleren van de ontwerpplannen, waarbij wordt nagegaan of de eis voor het raakvlak (bv. een doorrijhoogte) inderdaad gerespecteerd is.
- Menselijke fouten die in de plannen sluipen, kunnen door middel van een extra controle worden opgespoord.

In wat volgt wordt een mogelijk beheerproces voor het managen van raakvlakken verder toegelicht. Het raakvlakmanagement stoelt op de principes van Systems Engineering, en tracht aan deze methode een concrete invulling te geven met betrekking tot raakvlakken. De eerder aangehaalde softwareondersteuning wordt optimaal geïntegreerd in het hele proces. Enerzijds is het beheren van een volledig project quasi onmogelijk zonder een goed functionerende databankomgeving. Er is dan ook de nodige aandacht besteed aan de informatie die moet worden vastgelegd in een dergelijke databank. Anderzijds kan voor het opsporen van raakvlakproblemen in de ontwerpplannen gebruik gemaakt worden van clash detection software.

4.2.1 Systems Engineering – Breakdown Structures

Om het raakvlakmanagement op een goede manier te kunnen aanpakken, moet al bepaalde informatie beschikbaar zijn in het databeheersysteem.

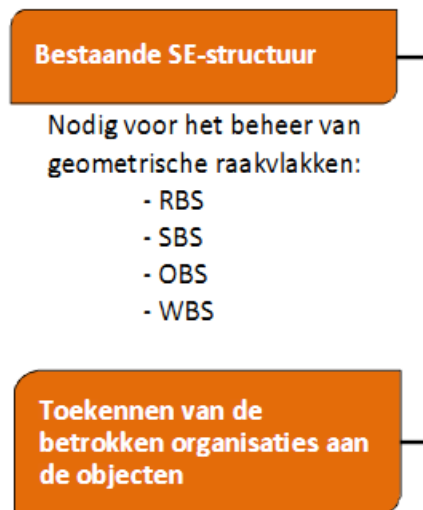
Zoals uitgelegd in §2.2 – Systems Engineering, worden binnen het project verschillende decomposities doorgevoerd. Zo zijn alle systeemelementen opgenomen in een System Breakdown Structure (SBS) of objectenboom. Vermits geometrische raakvlakken zich steeds voordoen tussen concrete objecten, moet deze objectenboom zeker gedefinieerd worden in de databankomgeving. Bovendien kunnen deze objecten de basis vormen voor het opstellen van één of meerdere N²-charts.

Aan elk raakvlak wordt exact één eis gekoppeld. Deze geeft op eenduidige wijze weer waaraan het raakvlak moet voldoen. Door per raakvlak maximaal één eis te definiëren, wordt een overvloed aan nietszeggende voorwaarden vermeden en blijft het overzicht bewaard. Alle eisen zijn gestructureerd weergegeven in de Requirements Breakdown Structure (RBS) of eisenboom.

Zoals verderop wordt uitgelegd, zal bij de activiteiten steeds worden aangegeven of er rekening gehouden moet worden met één of meerdere raakvlakken. Dit betekent wel dat de activiteiten in de vorm van een Work Breakdown Structure (WBS) of activiteitenboom in het systeem moeten vastgelegd zijn.

Ook de organisaties worden, eventueel samen met hun belangrijkste taken binnen het project, in de databankomgeving opgenomen. De Organization Breakdown Structure (OBS) lijkt in het kader van raakvlakkenbeheer op het eerste gezicht minder belangrijk. Toch kan het nuttig zijn aan te geven welke organisaties betrokken zijn bij de realisatie van een object, zodat eventuele vragen rechtstreeks aan het juiste bedrijf kunnen gesteld worden. Er wordt bovendien voor geopteerd geen verantwoordelijke personen aan de objecten te koppelen. Een object zal in de loop van het project immers continu onderhevig zijn aan veranderingen. De verantwoordelijkheid kan dan bijvoorbeeld verschuiven tijdens de ontwerpfase, maar zeker ook bij de overgang van het ontwerp naar de uitvoering. De verantwoordelijkheid wordt daarom toegekend aan activiteiten in plaats van aan objecten.

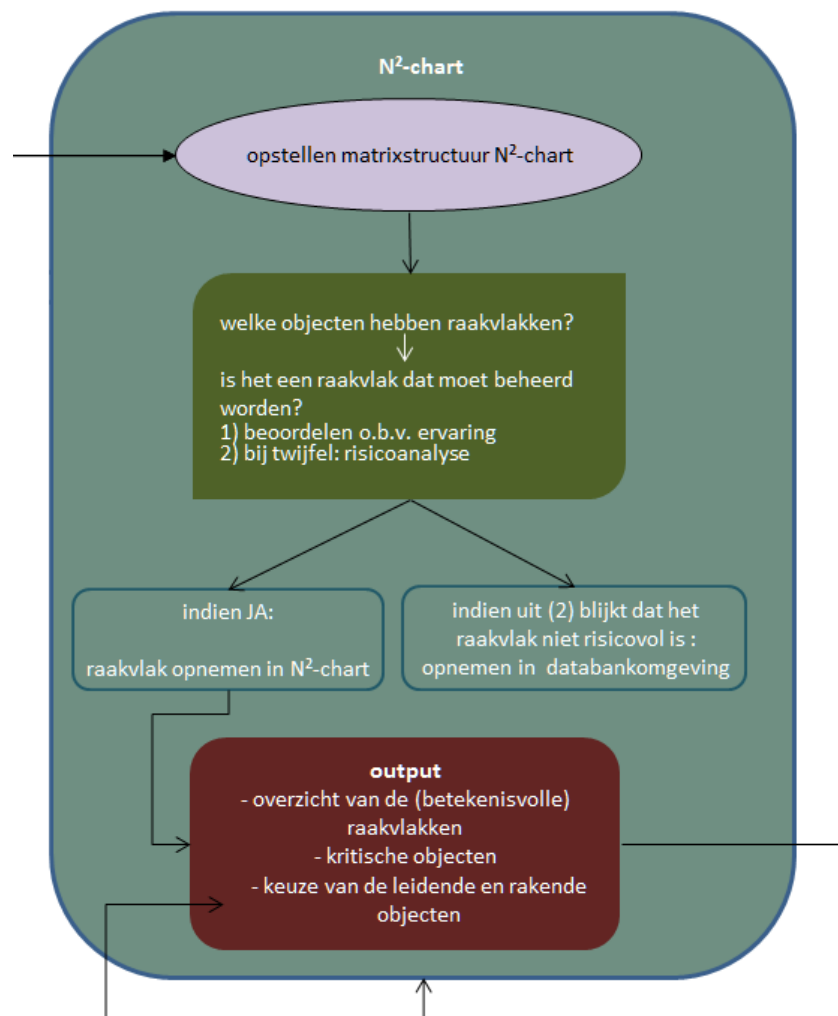
De informatie die noodzakelijk is om tot een goed raakvlakkenbeheer te kunnen komen, is weergegeven in de onderstaande **figuur**.



Figuur 28: Raakvlakkenbeheer - start van het proces

4.2.2 Raakvlakdetectie

De start van het beheerproces om raakvlakken correct te managen, bestaat uiteraard uit het in kaart brengen van deze raakvlakken. De ervaring van de betrokken personen speelt daarbij een cruciale rol. Zoals al eerder aangehaald, vergt het opstellen van een N²-matrix immers heel wat tijd en moeite (zie §4.1 – N²-chart). Wanneer de raakvlakken grotendeels op basis van ervaring gedetecteerd worden, is het niet nodig deze matrices telkens opnieuw op te stellen. Eén van de voordelen van de N²-chart is wel dat de opsteller ervan op een systematische en consequente manier de interactie tussen alle objecten moet nagaan. Het N²-diagram kan dus duidelijkheid brengen in geval van een complex interactieveld tussen verschillende objecten. De onderstaande **figuur** geeft de verschillende stappen weer om te komen tot een N²-chart.



Figuur 29: Raakvlakkenbeheer - detectie van raakvlakken aan de hand van een N²-chart

De meest aangewezen persoon om raakvlakken in kaart te brengen, is de raakvlakmanager. Deze heeft de nodige kennis in huis om indien nodig een N²-diagram op te stellen, en voor elk raakvlak te bepalen welke objecten leidend en rakend zijn. Zoals uitgelegd in §4.1 - N²-chart, kunnen soms conflicten optreden in de N²-chart. Het is de raakvlakmanager die in dergelijke situaties een eenduidige beslissing neemt. Deze manager kan uiteraard niet over voldoende kennis bezitten om elk raakvlak in het hele project tot in detail op te volgen. Op zijn initiatief worden de nodige overlegmomenten met de betrokken partijen georganiseerd.

Omdat de raakvlakmanager niet altijd voldoende diepgaande kennis van het probleem heeft, zou hij één van de bij het raakvlak betrokken personen als verantwoordelijke kunnen aanduiden. Voor het raakvlak ‘doorrijhoogte’ tussen een weg en een kunstwerk, zou dit betekenen dat bv. de ontwerper van de weg de verantwoordelijkheid voor het raakvlak krijgt. Dit impliceert dan wel dat deze ontwerper verantwoordelijk wordt voor een probleem waar hij niet volledig vat op heeft. Een manager heeft wel het nodige overzicht en kan de betrokken partijen gemakkelijker aansturen. Dit betekent dat de eindverantwoordelijkheid voor alle raakvlakken best bij de raakvlakmanager ligt. Op [figuur 34](#) is aangegeven wat de belangrijkste functies van de raakvlakmanager zijn tijdens het hele proces.

In tegenstelling tot de flowchart uit de *SE-wijzer* van BAM Infra ([figuur 7](#)), wordt hier geen expliciet onderscheid gemaakt tussen inter- en intradisciplinaire raakvlakken. Hoewel een interdisciplinair raakvlak moeilijker te beheersen is doordat er meer partijen bij betrokken zijn, vereist het beheer ervan geen specifieke methode. Door de raakvlakmanager verantwoordelijk te maken voor elk type raakvlak, zal hij ook kunnen bepalen welke extra acties nodig zijn om een dergelijk interdisciplinair raakvlak te beheren.

De gedetecteerde raakvlakken worden opgenomen in de databankomgeving. Onder [§4.3 – Implementatie van het proces in Relatics](#) wordt uitgelegd hoe dit in praktijk gebeurd is voor de relationele databank Relatics.

4.2.3 Raakvlakkenbeheer

Om een in kaart gebracht raakvlak te beheersen, wordt bepaald welke acties hiervoor nodig zijn. Deze acties worden vastgelegd en opgevolgd door de raakvlakmanager, eventueel in samenwerking met de betrokken partijen.

Een actie is semantisch verschillend van een activiteit: een actie is van korte duur en is niet onmiddellijk gekoppeld aan een concreet eindproduct. Een activiteit heeft meestal als doel een object te realiseren en loopt gedurende een langere periode. Elke activiteit heeft welgeteld één verantwoordelijke persoon onder de vorm van een 'rol' (bv. "ontwerpcoördinator"). Daarnaast kunnen een of meerdere personen belast zijn met de uitvoering van de activiteit. In tegenstelling tot de activiteiten, worden de acties niet opgenomen in de WBS. Voor het verdere onderscheid tussen deze semantische begrippen wordt verwezen naar [§4.3 – Implementatie van het proces in Relatics](#).

Een voorbeeld van een actie met betrekking tot de beheersing van een raakvlak kan "het opzoeken in het bestek van de te respecteren gronddekking op een PE-rioleringsbuis (Ø400mm)" zijn. De raakvlakmanager is de uiteindelijke verantwoordelijke voor het raakvlak, en dus ook voor alle acties die eraan gekoppeld zijn. Hij bepaalt bovendien welke personen belast worden met de uitvoering van de verschillende acties.

Hoewel een raakvlak bepaald wordt door de opeenvolging van acties die nodig zijn om dit raakvlak te beheersen, wordt het wel gedefinieerd tussen twee concrete objecten. Eén van de beide objecten is leidend, het andere is rakend.

Naast het definiëren van de nodige acties, worden de raakvlakken ook gekoppeld aan activiteiten. Bij de uitvoering van deze activiteiten is het immers belangrijk te weten met welke raakvlakken rekening gehouden moet worden. Zo zal bij het brugontwerp de opgelegde doorrijhoogte gerespecteerd moeten worden. Zoals al vermeld, heeft een activiteit normaliter een of meerdere eindproducten en is ze in het leven geroepen om een object te realiseren. Activiteiten worden dan ook in de eerste plaats gekoppeld aan dit object, maar vanuit de activiteit is het mogelijk te

selecteren welke raakvlakken ermee te maken hebben. Op die manier ziet de uitvoerder van de activiteit onmiddellijk met welke raakvlakken hij rekening moet houden. Het voordeel van deze werkwijze is dat het mogelijk wordt de evolutie van het project te volgen door telkens nieuwe activiteiten te definiëren. Net zoals bij raakvlakken, is het voor de activiteiten nuttig de status weer te geven (uit te voeren / in uitvoering / afgewerkt).

De belangrijkste richtlijnen met betrekking tot het beheer van geometrische raakvlakken zijn opgenomen in de onderstaande **figuur**.



Figuur 30: Raakvlakkenbeheer - implementatie in de databankomgeving

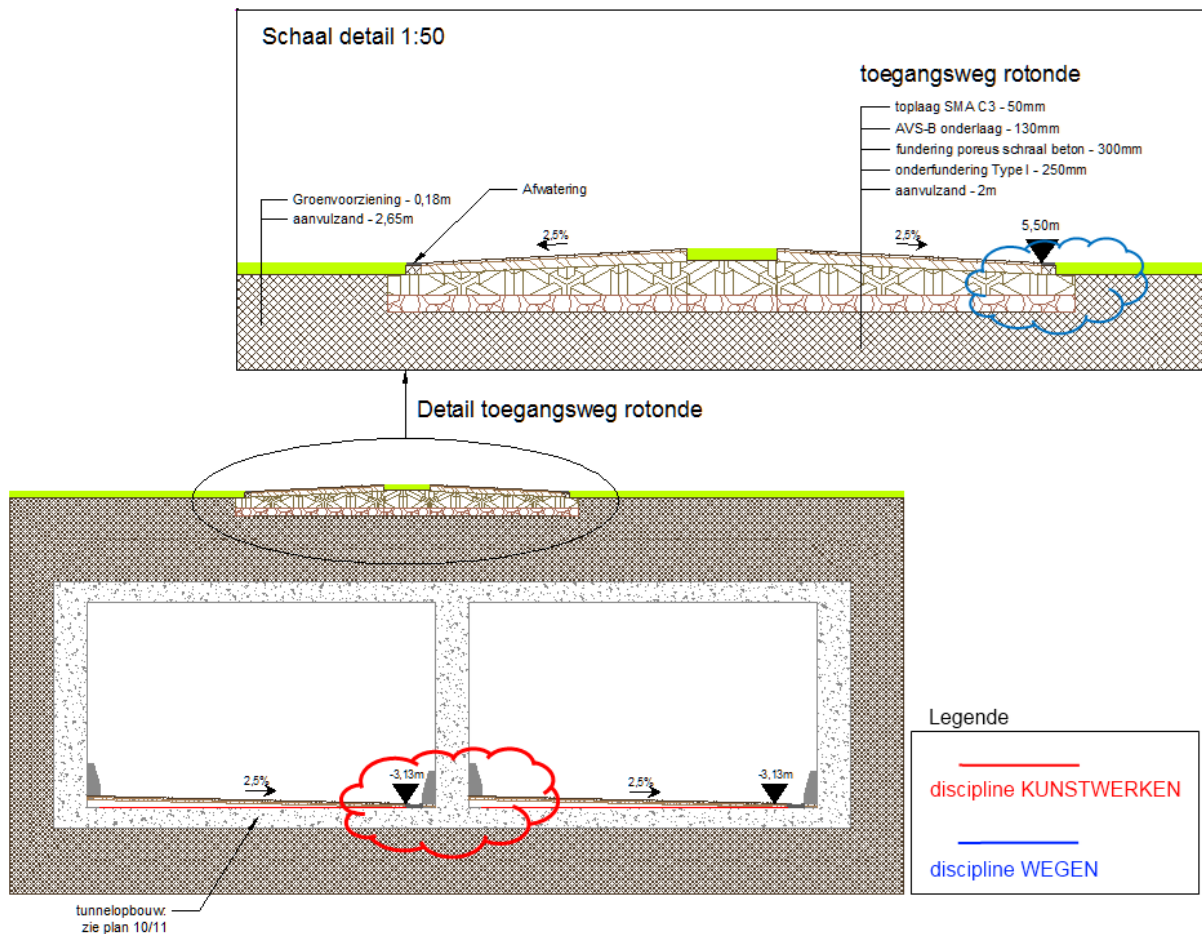
4.2.4 Opgvolging & verificatie

De raakvlakmanager ziet erop toe dat de acties met betrekking tot een raakvlak worden uitgevoerd. Zoals al aangegeven, blijft het ook nodig de ontwerpplannen te controleren. Dit kan handmatig gebeuren of met behulp van clash detection software. De raakvlakmanager kan de controles zelf uitvoeren, maar het is niet ondenkbaar dat deze manager onvoldoende inzicht in de problematiek heeft. In dit geval kan hij de controle delegeren aan een van de betrokken personen.

Raakvlakken die pas in de uitvoeringsfase ontdekt worden, of tijdens deze fase specifieke aandacht vereisen, worden vaak vanop de werf opgelost. Wanneer raakvlakmanagement geen deel uitmaakt van het volledige projectbeheer, is dit bovendien een van de weinige momenten waarop een raakvlak naar boven komt. In de praktijk is er dus nood aan een efficiënte afhandeling van deze problemen op de werf.

Hierbij komt het er vooral op aan eenduidige afspraken te maken. Zo kan worden beslist dat de discipline WEGEN leidend is boven alle andere disciplines.

Wanneer een teamlid een raakvlak ontdekt, kan dit door de verantwoordelijke van de discipline worden aangegeven met een simpele aanduiding (bijvoorbeeld in de vorm van een wolkje) op een van de plannen. De kleur van de aanduiding geeft meteen aan bij welke discipline het raakvlak gedetecteerd is. Dit is bij wijze van voorbeeld gebeurd op het onderstaande [plan](#). De plannen kunnen vervolgens op een gemeenschappelijke plaats worden opgehangen, zodat iedereen ze gemakkelijk weet te vinden. Het voordeel hierbij is dat het raakvlak overzichtelijk wordt gesignaleerd op plannen. De aanduidingen geven daarbij aan dat er op die plaats een probleem is waarvoor acties moeten ondernomen worden. De verantwoordelijkheid voor het melden van raakvlakken wordt dus bij de verschillende teams zelf gelegd. Het is bovendien belangrijk dat alle partijen op regelmatige basis rond de tafel gaan zitten. Op dat moment kan worden nagegaan of er al een oplossing is gevonden voor een gemeld raakvlak, welke maatregelen nog moeten genomen worden enz.

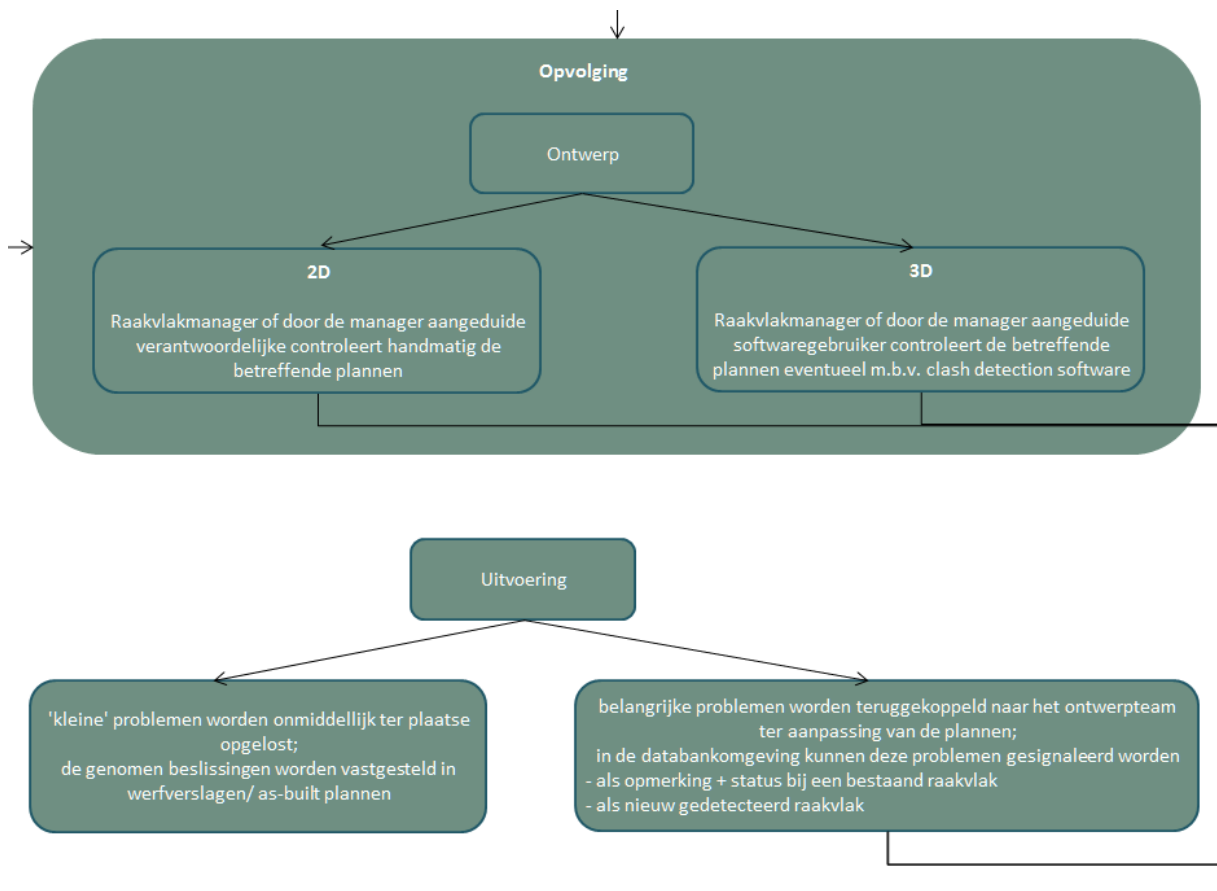


Figuur 31: Raakvlakkenbeheer tijdens de uitvoering

Deze werkwijze is niet de enige om met raakvlakken om te gaan tijdens de uitvoeringsfase, maar het is wel een heel efficiënte methode. De leden van het projectteam kunnen zonder problemen een proces uitstippelen dat in hun specifieke situatie werkbaar is. De enige voorwaarde hierbij is dat het proces voldoende ruimte laat voor communicatie tussen alle betrokkenen. Relatief 'kleine' problemen kunnen dus vaak ter plaatse worden opgelost, en de genomen beslissingen worden vastgelegd in bv. werfverslagen.

Belangrijke problemen moeten teruggekoppeld worden naar het ontwerpteam om de plannen aan te passen. Conform de denkwijze van Systems Engineering worden de gemelde problemen ook vastgelegd in de databank van het project. Dit kan in de vorm van een opmerking bij een bestaand raakvlak of als nieuw gedetecteerd raakvlak.

De onderstaande **figuur** toont op schematische wijze de nodige stappen in de opvolging van raakvlakken.



Figuur 32: Raakvlakkenbeheer - opvolging

Naast het opvolgen van de raakvlakken binnen een project, wordt op geregelde tijdstippen nog een extra verificatiestap ingebouwd. Daarin wordt, zoals schematisch weergegeven in [figuur 33](#), nagegaan of de reeds afgewerkte acties die horen bij het raakvlak correct zijn afgehandeld. Door na een deel van de acties al een verificatiestap in te bouwen, kan de evolutie van het project goed opgevolgd worden. Zo wordt het ook mogelijk de verificatie in de verschillende projectfasen (ontwerp, uitvoering ...) door een andere verantwoordelijke te laten uitvoeren.



Figuur 33: Raakvlakkenbeheer - verificatie

4.2.5 Samengevat

De bovenstaande stappen zijn essentieel voor een succesvol raakvlakkenbeheer. De onderstaande **flowchart** toont hoe de afzonderlijke componenten geïntegreerd kunnen worden in één doorlopend proces.

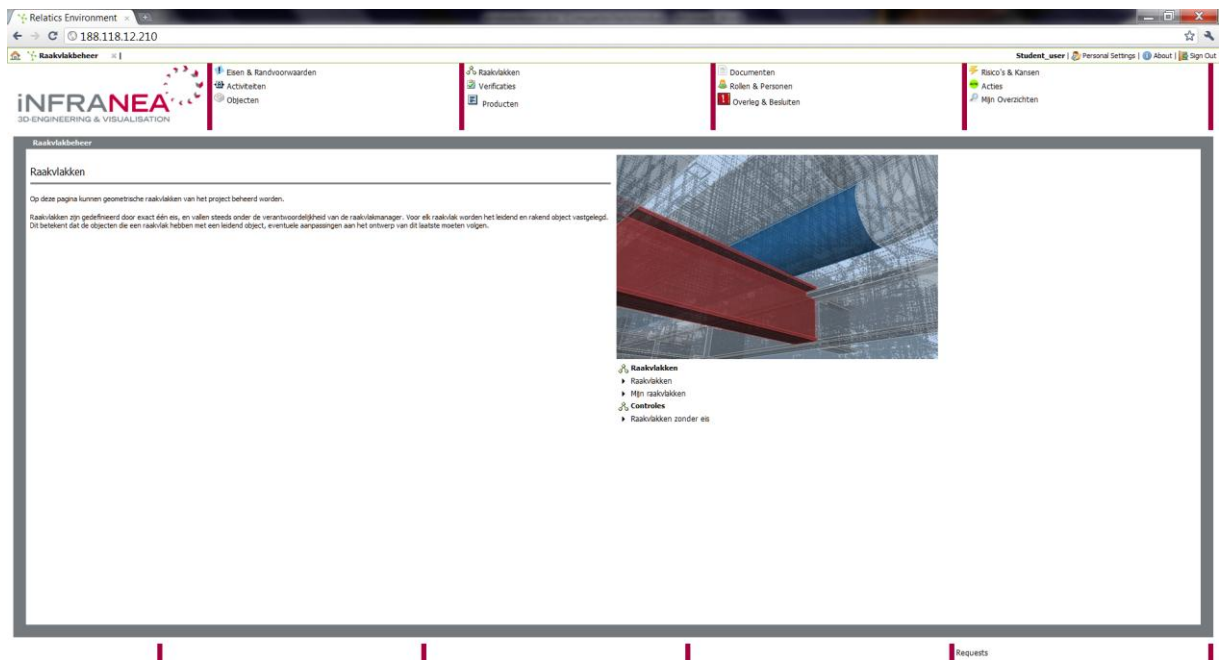


Figuur 34: Raakvlakkenbeheer - flowchart

4.3 Implementatie van het proces in Relatics

In wat volgt, wordt geïllustreerd hoe een relationele databank als Relatics kan worden opgezet om, volgens de in §4.2 – Proces voor het beheer van geometrische raakvlakken uiteengezette principes, een project efficiënt te beheren. Op de bijgevoegde cd-rom is een demofilm beschikbaar, die beeld en uitleg geeft bij de opgezette Relaticsomgeving. Hierbij dient te worden opgemerkt dat de demo beperkt is tot wat nodig is om (geometrische) raakvlakken te beheren, en bijgevolg slechts een deel vormt van de hele projectomgeving.

Het spreekt voor zich dat de onderstaande figuren in de tekst slechts een fragment vormen van alle elementen omtrent raakvlakken die in de Relaticsomgeving zijn opgenomen.



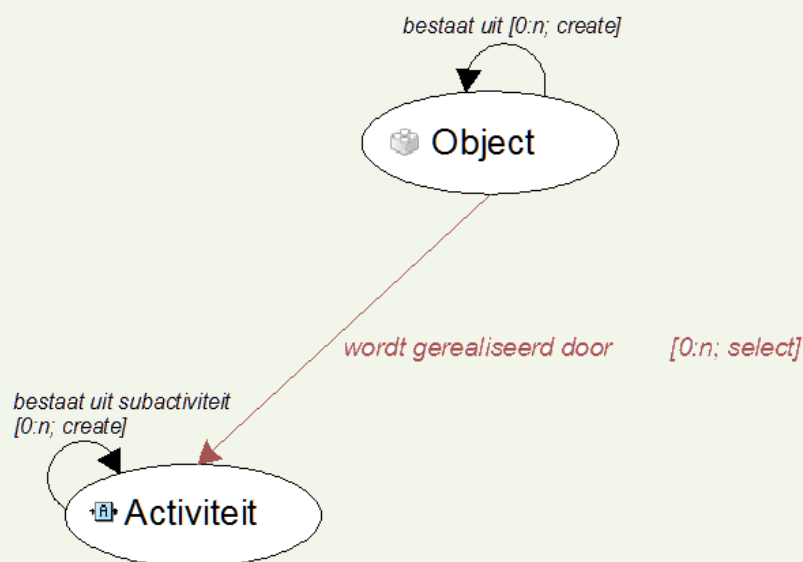
Figuur 35: Screenshot van de Relaticsomgeving voor raakvlakkenbeheer

In de bovenstaande tekst is al aangehaald dat Relatics een relationele databank is. De elementen moeten in een dergelijk databeheersysteem slechts één keer gedefinieerd worden. Met behulp van 'relaties' kan vervolgens vanop verschillende plaatsen naar al deze elementen verwezen

worden. Deze relaties kunnen in hun eenvoudigste vorm gezien worden als een soort touw dat twee unieke objecten met elkaar verbindt. Door aan deze verbinding een naam te koppelen, kan een volledig semantisch of betekenisvol netwerk worden opgebouwd, zoals ook blijkt uit het onderstaande voorbeeld. Algemener informatie over Relatics is terug te vinden in §2.4.1 – Databeheersystemen.

Voorbeeld 2

Een 'object' en een 'activiteit' zijn beide elementen in Relatics waartussen verschillende 'relaties' liggen. Een voorbeeld van een relatie is weergegeven op de onderstaande figuur.



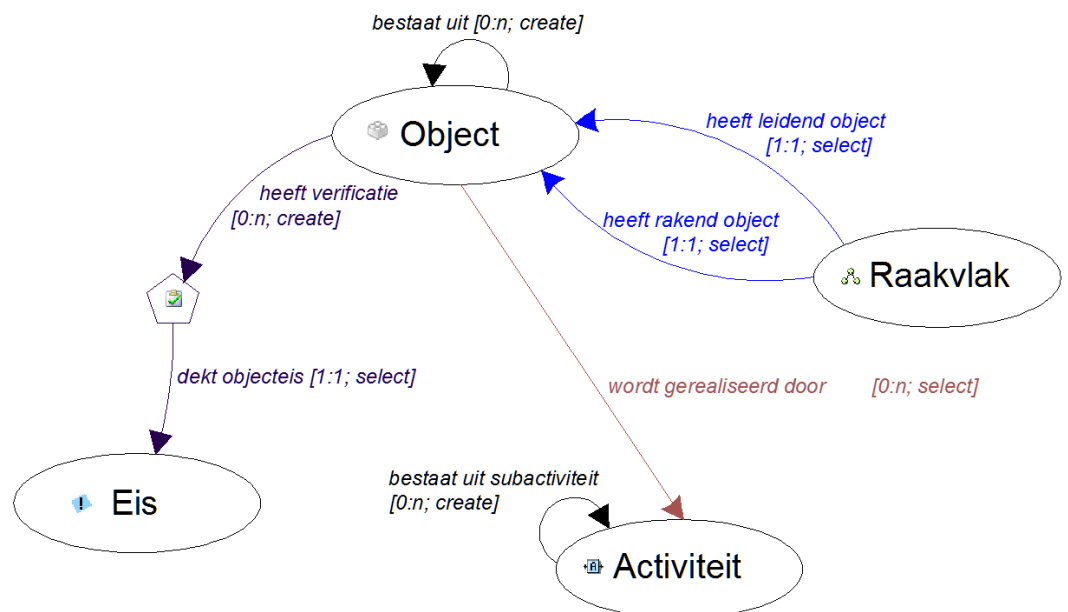
Figuur 36: Relaties tussen 'Object' en 'Activiteit'

In alle voorbeelden zijn de relaties cursief gedrukt om te illustreren hoe deze deel uit maken van de databankomgeving. Het gaat hier om de naam die aan de relaties wordt gegeven om zo een semantisch netwerk te bekomen. De 'wordt gerealiseerd door'-relatie in de figuur kan bijvoorbeeld eenvoudig worden gebruikt om aan te geven dat het object "fietspad" wordt gerealiseerd door de activiteit "geometrisch wegontwerp". Wanneer in Relatics dan naar het object "fietspad" gekeken wordt, kan deze relatie met de activiteit "geometrisch wegontwerp" getoond worden. De omgekeerde relatie, van de activiteit naar het object, geeft dan weer dat "geometrisch wegontwerp" het "fietspad" realiseert.

In het verdere verloop van deze paragraaf zullen verschillende elementen uit Relatics eerst afzonderlijk worden toegelicht, om ze vervolgens samen te voegen tot een werkbaar geheel.

OBJECT

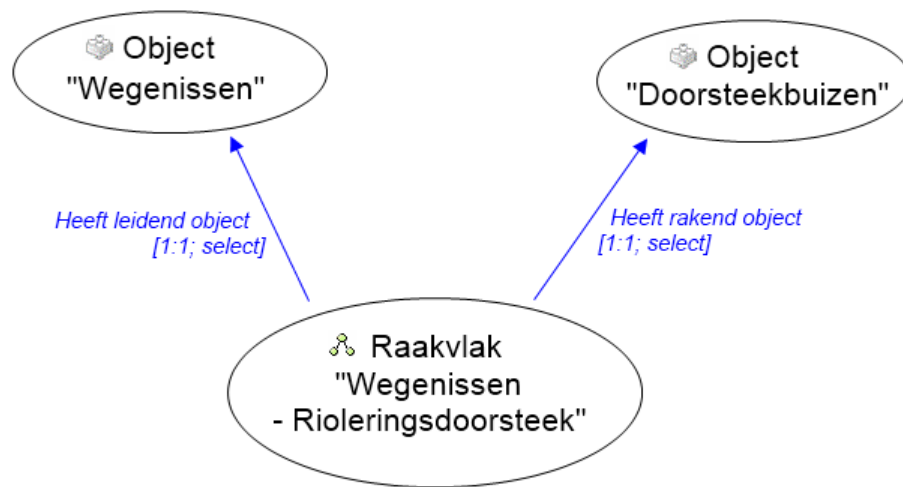
Het uiteindelijke doel van een bouwkundig project is het realiseren van de op te leveren objecten. Deze visie kan worden doorgetrokken naar de databankomgeving, waarbij de fysieke objecten worden beschouwd als centrale spil van het systeem. Zoals de onderstaande **figuur** toont, vormen deze objecten het startpunt voor heel wat relaties (de informatie tussen vierkante haken bij elke relatie is verklaard op p. 112).



Figuur 37: Relaties vanuit 'Object'

Relatie Object – Raakvlak

Zoals weergegeven in de volgende **figuur**, wordt een raakvlak tussen twee objecten gemodelleerd door dit raakvlak te verbinden met deze objecten.



Figuur 38: Modelling van een geometrisch raakvlak in Relatics

Het spreekt voor zich dat een raakvlak maar één rakend en één leidend object kan hebben. Dit onderscheid is in Relatics geïmplementeerd door gebruik te maken van 2 afzonderlijke relaties: *heeft leidend object* en *heeft rakend object*. De onderstaande **figuur** toont het resultaat van deze verbanden voor de eindgebruiker van Relatics.

Raakvlak informatie

Raakvlak Wegenissen - Rioleringsdoorsteek

Algemeen		Bijkomende informatie
Algemeen		
Naam	Raakvlak Wegenissen - Rioleringsdoorsteek	
ID	RKVLK_0057	
Omschrijving		
Status	Afgehandeld	
Fase signalisatie	Ontwerp	
Interdisciplinair	Ja	
Heeft leidend en rakend object		
Leidend object	E0500 - Wegenissen	
Rakend object	E0232 - Doorsteekbuizen	

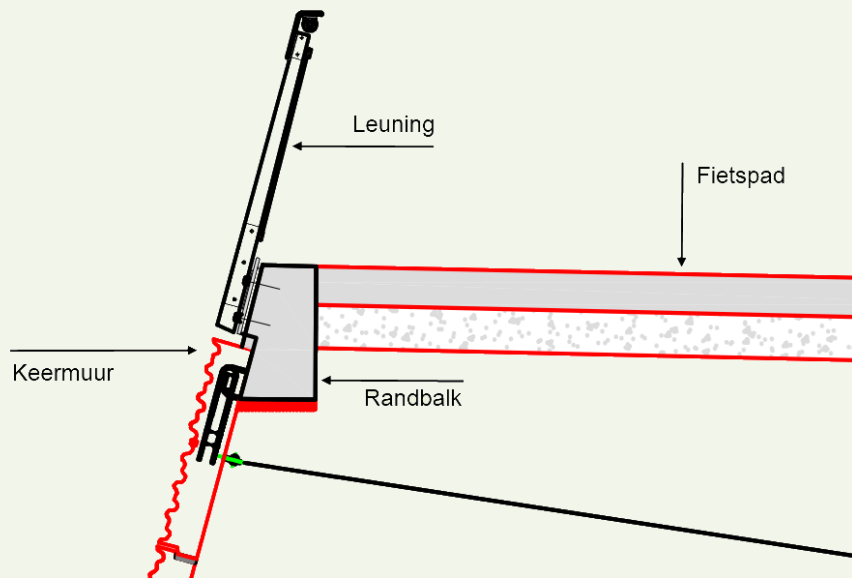
Figuur 39: Leidend en rakend object in Relatics

Als een raakvlak betrekking heeft op meer dan twee elementen wordt dit best opgesplitst naar afzonderlijke raakvlakken tussen twee objecten. Zo is het eenvoudiger om overzicht te bewaren over het systeem. Het volgende

voorbeeld, dat al is aangehaald in §4.1.1 - Bepaling van de kritische objecten, illustreert op welke manier zo'n raakvlak kan worden opgesplitst.

Voorbeeld 3

Veronderstel een randbalk die de bovenzijde van een keermuur afsluit, en waar ook een leuning aan bevestigd wordt. De randbalk moet tevens aansluiten op het fietspad ernaast. De situatie is aanschouwelijk voorgesteld in de onderstaande figuur.



Figuur 40: Schets randbalk

Wanneer zowel vanuit de leuning, de keermuur als het fietspad eisen worden gesteld aan de positie van de randbalk, gaat het hier om een drievoudig raakvlak. Dit raakvlak kan echter gemakkelijk in drie delen worden opgesplitst: een raakvlak “randbalk-leuning”, “randbalk-keermuur” en “randbalk-fietspad”. Alle informatie die bij het drievoudige raakvlak hoort, kan ook bij deze ‘gewone’ raakvlakken weergegeven worden. De randbalk wordt in praktijk wel door al de verschillende objecten beïnvloed. Dit kan in de databank worden getoond door bij het object “randbalk” de drie raakvlakken weer te geven.

Relatie Object – Activiteit

Zoals reeds uitgelegd, worden 'activiteiten' in het systeem gedefinieerd in een Work Breakdown Structure (WBS). Deze activiteiten kunnen enerzijds taken voorstellen die een concrete bijdrage aan het project leveren, zoals "storten onderfundering". Een activiteit kan anderzijds ook een handeling zijn in het kader van bv. het projectbeheer, zoals "implementeren databankomgeving". In dit geval levert de uitvoering van de activiteit geen tastbaar eindproduct op.

Het element 'activiteit' wordt, zoals in [voorbeeld 2](#) al vermeld, via de relatie *wordt gerealiseerd door* aan 'object' gekoppeld (zie onderstaande [figuur](#)). Daarnaast is het, zoals op [figuur 45](#) weergegeven, mogelijk bij een activiteit aan te geven in welke projectfase deze plaats vindt. Dit gebeurt in Relatics door de eigenschap "fase" bij de activiteit te definiëren, standaard is de fase ingesteld op "ontwerp". Door middel van een filter kunnen activiteiten opgezocht en gesorteerd worden volgens hun status. Dit kan belangrijk zijn wanneer alle activiteiten uit bv. de ontwerpfase geverifieerd moeten worden.

Object		
E0232 - Doorsteekbuizen		
Algemeen	Eisen	Verificaties
Algemeen		
ID	Obj-03456	
Naam	E0232 - Doorsteekbuizen	
Omschrijving		
Objecttype		
Locatie		
is leidend voor Raakvlak		
Naam	Status	Eis
is rakend voor Raakvlak		
Naam	Status	Eis
Raakvlak Wegenissen - Rioleringsdoorsteek	Afgehandeld	Gronddekking op rioleringsbuizen
wordt gerealiseerd door		
ID	Activiteit	Fase
Act-02250	W026d - Uitvoeringsontwerp waterbeheer	Ontwerp
Act-02271	W041 - Preventief onderhoud	Ontwerp

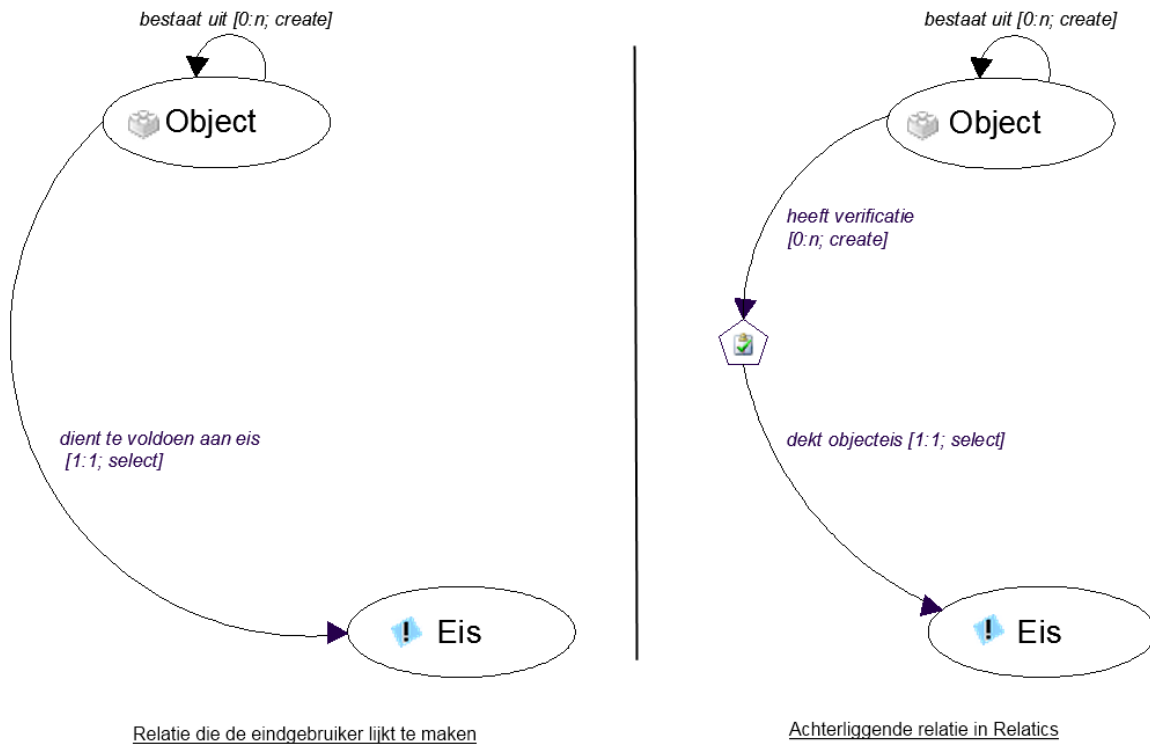
Figuur 41: 'Objecten' gerealiseerd door 'activiteiten'

Het is duidelijk dat de WBS voldoende ver gedecomposeerd moet worden opdat de link met een object betekenisvol zou zijn. Een relatie tussen het object “fundering” en een algemene activiteit “wegontwerp” zegt veel minder dan de relatie tussen “fundering” en “ontwerp wegfundering”. Door de richtlijn uit het projectmanagement te hanteren dat één activiteit slechts één object mag realiseren, kan hieraan voldaan worden.

Relatie Object – Eis

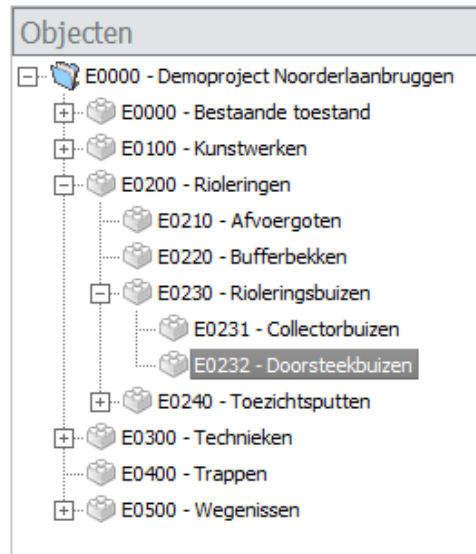
De eisen waaraan een object moet voldoen, worden door een relatie aan het object gekoppeld. In de loop van het proces moet geverifieerd worden of het object wel degelijk voldoet aan de gestelde eisen. In de Relaticsomgeving wordt, wanneer deze relatie tussen een object en een eis gelegd wordt, automatisch een ‘verificatie-element’ aangemaakt. Dit element is niet onmiddellijk zichtbaar voor de eindgebruiker, en is onlosmakelijk verbonden met zowel het object als met de eis in kwestie. Door bepaalde eigenschappen van het verificatie-element wel zichtbaar te maken voor de eindgebruiker, kan er mee aangegeven worden of de verificatie nog moet gebeuren of al volbracht is, wie de verificatie moet uitvoeren en op welke datum dit ten laatste moet gebeuren.

De onderstaande **figuur** toont het verschil tussen wat de eindgebruiker zelf doet, en wat in Relatics bijkomend gebeurt.



Figuur 42: Achterliggende object-eisrelatie in Relatics

Wanneer eisen aan objecten gekoppeld worden, kan dit zowel op het laagste niveau van de objectenboom gebeuren als op een hoger niveau. Zo kan er bijvoorbeeld een eis bestaan die op *alle* rioleringsbuizen van toepassing is. In plaats van deze dan aan elke rioleringsbuis afzonderlijk te koppelen, kan ze net zo goed op een hoger niveau aan het object “rioleringsbuizen” gelinkt worden. Dit brengt evenwel het risico met zich mee dat deze hoger gedefinieerde eisen bij een lager gelegen object uit het oog verloren worden. Om dit probleem op te vangen, wordt bij elk object een extra boomstructuur weergegeven waarin alle bovenliggende objecten, en de eisen die daaraan gesteld zijn, getoond worden. **Figuur 44** laat duidelijk zien hoe dit in Relatics is voorgesteld, de onderstaande **figuur** toont de positie van het object “doorsteekbuizen” in de objectenboom.



Figuur 43: Objectenboom bij figuur 44

De eisen in de tabellen “dient te voldoen aan Raakvlakeis(en)” en “dient te voldoen aan Activiteiteis(en)” zijn gesteld aan activiteiten en raakvlakken die gekoppeld zijn aan het getoonde object. De object-, activiteit- en raakvlakeisen van de hoger gelegen objecten in de SBS worden in de boomstructuur “Bovenliggende eisen” weergegeven.

Object 🗨

E0232 - Doorsteekbuizen

Algemeen		Eisen		Verificaties
dient te voldoen aan Objecteis(en)				
ID	Eis	Eistekst	Eiscategorie	Verificatiestatus
dient te voldoen aan Raakvlakeis(en)				
ID	Eis	Eistekst	Eiscategorie	Verificatiestatus
E&R-01152	⚠ Gronddekking op rioleringsbuizen	De gronddekking op de rioleringsbuizen moet conform het bestek zijn	⚠ 08 - Waterhuishouding, grond- en oppervlaktewater	te verifiëren
dient te voldoen aan Activiteiteis(en)				
ID	Eis	Eistekst	Eiscategorie	Status
Bovenliggende eisen				
<ul style="list-style-type: none"> 🗨 E0230 - Rioleringsbuizen <ul style="list-style-type: none"> ⚠ objecteis: Geen rioleringen in vluchtstroken - Rioleringen mogen niet in de vluchtstroken aangelegd worden. ⚠ activiteiteis: Ontwerp afwatering - Het ontwerp van de rioleringsbuizen moet voldoen aan de richtlijnen van het OCW 🗨 E0200 - Rioleringen 				

Figuur 44: Eisen bij objecten

RAAKVLAK

Geometrische raakvlakken doen zich steeds voor tussen concrete objecten, maar hebben ook relaties met activiteiten, acties en rollen.

Relatie Raakvlak – Activiteit

In de bovenstaande paragraaf werd beschreven hoe een raakvlak gedefinieerd wordt tussen twee objecten. Deze objecten hebben bovendien relaties naar de bijbehorende activiteiten. Daarom is ervoor gekozen om een raakvlak ook te linken aan één of meerdere activiteiten die bij deze objecten horen. Het voordeel hiervan is dat wie aan een activiteit bezig is onmiddellijk kan zien met welke raakvlakken hij rekening moet houden. Deze aanpak zorgt er bovendien voor dat de medewerkers in het project enkel de voor hen relevante raakvlakken te zien krijgen. De onderstaande **figuur** illustreert hoe dit in de Relatics omgeving getoond wordt.

Activiteit			
Geometrisch ontwerp brug autoverkeer			
Algemeen		Eisen	Planning en meetstaat
Algemeen			
ID	Act-02296		
Naam	Geometrisch ontwerp brug autoverkeer		
Omschrijving			
Locatie			
Start	2011-05-07		
Status	te beginnen		
Fase	Ontwerp		
heeft Eindverantwoordelijke			Uitvoerders
Rol	Persoon	Status	Rol
Ontwerpleider Wegontwerp	Peter	Aanvaard	Junior Wegontwerper
			Senior Wegontwerper
			Liesbeth
			Gunther
moet rekening houden met raakvlak			
Raakvlak	Status	Eistekst	
! Raakvlak Bredastraat - Brug autoverkeer	Op te volgen	De doorrijhoogte voor Weg1 is minimaal 4,60 meter volgens het PvE	
! Raakvlak Wegenissen - Rioleringsdoorsteek	Afgehandeld	De gronddekking op de rioleringsbuizen moet conform het bestek zijn	

Figuur 45: Raakvlakken, gekoppeld aan een activiteit

Relatie Raakvlak – Actie

Terwijl de relatie tussen een raakvlak en een activiteit enkel dient om te ontsluiten welke activiteiten met een bepaald raakvlak rekening moeten houden, kunnen er aan een raakvlak ook 'acties' gekoppeld worden. Deze

acties zijn handelingen die nodig zijn om het raakvlak te beheren, maar niet direct verband houden met de objecten van het raakvlak. Zo kan het “afstemmen van de ontwerpplannen” een actie zijn. De raakvlakmanager is ervoor verantwoordelijk dat deze acties uitgevoerd worden, maar kan ze indien nodig ook door iemand anders laten afhandelen.

Relatie Raakvlak – Rol

In §4.2 – [Proces voor het beheer van geometrische raakvlakken](#) is uitgelegd dat de verantwoordelijkheid voor een raakvlak bij de raakvlakmanager ligt: hij is het beste geplaatst om het probleem te overzien en de communicatie tussen de betrokken partijen te coördineren. De verantwoordelijkheid voor het raakvlak wordt aan de ‘rol’ van “raakvlakmanager” gekoppeld en niet aan een specifieke persoon. Als de persoon die deze functie vervult op een gegeven moment uit het project verdwijnt, zal een vervanger automatisch deze verantwoordelijkheid overerven. Wanneer in Relatics een nieuw raakvlak wordt aangemaakt, is de verantwoordelijke standaard ingesteld op de rol “raakvlakmanager”, wat ook manueel gewijzigd kan worden.

Relatie Raakvlak – Eis

De relatie tussen een raakvlak en de eis die het raakvlak omschrijft, is op dezelfde manier opgebouwd als de *object-eis*-relatie. Ook hier wordt met een verificatie-element gewerkt dat zowel aan het object als aan de eis gekoppeld is, en waaraan alle nodige gegevens voor de verificatie van de raakvlakeis gelinkt kunnen worden.

De onderstaande [figuur](#) illustreert op welke manier de eis bij een raakvlak wordt weergegeven. Naast de naam van de eis, een unieke identificatiecode en de eistekst, worden ook de categorie waartoe de eis behoort en de verificatiestatus getoond.

Raakvlak informatie

Raakvlak Bredastraat - Brug autoverkeer

Algemeen		Bijkomende informatie		
Algemeen				
Naam	Raakvlak Bredastraat - Brug autoverkeer			
ID	RKVLK_0071			
Omschrijving				
Status	Op te volgen			
Fase signalisatie	Ontwerp			
Interdisciplinair	Ja			
Dient te voldoen aan				
Eis	ID	Eistekst	Categorie	Verificatie
Doorrijhoogte Bredastraat	E&R-01061	De doorrijhoogte voor Weg1 is minimaal 4,60 meter volgens het PvE	02 - Geometrie Weg - wegplatform	te verifiëren

Figuur 46: Raakvlakeisen in Relatics

Uit wat voorafgaat, blijkt dat aan elk raakvlak heel wat informatie gekoppeld wordt. Het onderstaande overzicht geeft weer welke elementen zichtbaar zijn voor de eindgebruiker.

- **Naam**

Het is belangrijk dat een raakvlak een betekenisvolle naam krijgt in het systeem. De praktijk leert dat een generieke benaming als “raakvlak 4” al snel leidt tot een onoverzichtelijke lijst, wat ten koste gaat van de bruikbaarheid van het systeem. Het kan bv. een oplossing zijn om af te spreken dat elk raakvlak begint met “raakvlak”, gevolgd door de naam van het leidend en rakend object (zie ook [figuur 49](#)).

- **ID**

Elk raakvlak krijgt een uniek identificatienummer, om verwarring met andere raakvlakken, die een gelijkaardige naam hebben, te voorkomen. De ID-code wordt automatisch toegekend wanneer een nieuw raakvlak wordt aangemaakt.

- **Beschrijving**

Aan het raakvlak kan ook een korte beschrijving gegeven worden.

- **Interdisciplinair raakvlak**

Omdat interdisciplinaire raakvlakken vaak extra in het oog moeten gehouden worden, kan in Relatics worden aangegeven of een raakvlak onder deze categorie valt. Op

die manier is het ook mogelijk alle interdisciplinaire raakvlakken in het systeem uit te filteren om er bijvoorbeeld een specifieke actie aan te koppelen.

- Eis

Zoals al verklaard, wordt aan een raakvlak exact één eis gekoppeld.

- Leidend en Rakend object

Door zowel het rakend als het leidend object weer te geven, is meteen duidelijk op welke objecten het raakvlak betrekking heeft. Zoals eerder vermeld, hebben sommige raakvlakken een algemenere eis die van toepassing is op meerdere objecten. Zo geldt de eis “De gronddekking voor de rioleringsbuizen moet gerespecteerd worden” voor alle rioleringsbuizen, ongeacht hun type. In plaats van het raakvlak te koppelen aan elke rioleringsbuis afzonderlijk, is het ook mogelijk een relatie te leggen naar het type object “rioleringsbuis”. Alle rioleringsbuizen (bv. PE-buis) die onder dit type object worden aangemaakt, kunnen dan dit raakvlak overerven. Zo wordt veel repetitief werk bij het toekennen van raakvlakken in het systeem vermeden.

- Activiteiten

Bij elke activiteit kan worden geselecteerd met welke raakvlakken rekening gehouden moet worden. De geselecteerde activiteiten zijn ook zichtbaar bij de informatie over een welbepaald raakvlak, zoals blijkt uit [figuur 47](#).

- Risico

Het kan nuttig zijn aan te geven welk risico met het raakvlak gepaard gaat, bv. “het bezwijken van een rioleringsbuis als er niet voldoende gronddekking voorzien wordt”.

- Raakvlakmanager

De rol van “raakvlakmanager” wordt ingevuld door hier een specifieke persoon aan te duiden. Zo is voor iedereen duidelijk wie de eindverantwoordelijke is, en het aanspreekpunt in geval van problemen.

- Fase signalisatie

Idealiter wordt elk raakvlak voor het begin van het ontwerp al in kaart gebracht. Tijdens de verschillende projectfasen zullen

waarschijnlijk nog raakvlakken naar boven komen die initieel over het hoofd gezien werden. Zo is het mogelijk dat op de werf nog een belangrijk raakvlak wordt ontdekt dat moet teruggekoppeld worden naar het ontwerpteam. In de databank wordt dan aangegeven dat dit probleem tijdens de uitvoeringsfase is ontdekt. De fase is default ingesteld op “ontwerp”, maar kan ook “voorontwerp”, “uitvoering”, “onderhoud” of “afbraak” zijn.

- Status

Hiermee wordt weergegeven of het raakvlak nog actueel is of niet. De mogelijke statussen zijn: “op te volgen” / “afgehandeld”. Standaard is de status van een raakvlak ingesteld op “op te volgen”.

- Op te volgen afspraken

Beslissingen die tijdens overlegmomenten over het raakvlak genomen zijn, worden in Relatics vastgelegd zodat iedereen ze kan terugvinden.

- Heeft opmerking

Hier kan elke gebruiker van het systeem opmerkingen toevoegen i.v.m. het raakvlak. Zo nodig kan de verantwoordelijke dan een overlegmoment rond deze opmerkingen organiseren of het onderwerp op de agenda van een vergadering zetten.

Er zouden natuurlijk nog meer elementen aan het raakvlak gekoppeld kunnen worden, maar het is niet de bedoeling de eindgebruiker te overstelpen met overbodige informatie. Bovendien moet al die extra informatie ook in het systeem ingebracht worden, wat de werklust aanzienlijk vergroot. De volgende figuren geven weer hoe al de bovenstaande informatie ontsloten is in het detailvenster van Relatics. Daarbij is het ook mogelijk een risico of besluit omtrent het raakvlak weer te geven.

Raakvlak informatie

Raakvlak Wegenissen - Rioleringsdoorsteek

Algemeen		Bijkomende informatie		
Algemeen				
Naam	! Raakvlak Wegenissen - Rioleringsdoorsteek			
ID	RKVLK_0057			
Omschrijving				
Status	Afgehandeld			
Fase signalisatie	Ontwerp			
Interdisciplinair	Ja			
Dient te voldoen aan				
Eis	ID	Eistekst	Categorie	Verificatie
! Gronddekking op rioleringsbuizen	E&R-01152	De gronddekking op de rioleringsbuizen moet conform het bestek zijn	! 08 - Waterhuishouding, grond- en oppervlaktewater	te verifiëren
Heeft leidend en rakend object				
Leidend object	E0500 - Wegenissen			
Rakend object	E0232 - Doorsteekbuizen			
is raakvlak bij activiteit				
Activiteit	Status	Rol	Naam	
Detailontwerp weginfrastructuur Noorderlaan	te beginnen	6. Directie Besix	Walter	
Geometrisch ontwerp brug autoverkeer	te beginnen	Ontwerpleider Wegontwerp	Peter	
Geometrisch ontwerp weginfrastructuur Noorderlaan	te beginnen	Ontwerpleider Wegontwerp	Peter	
Uitvoeringsontwerp weginfrastructuur Noorderlaan	te beginnen	6. Directie Besix	Walter	
W026d - Uitvoeringsontwerp waterbeheer	te beginnen	Ontwerpleider Integraal waterbeheer	Eddy	
Verantwoordelijke				
Rol				Persoon
1.5 - Raakvlakmanager				Thibaud
Acties				
Actie	Status	Rol	Uitvoerder	
Opzoeken gronddekking rioleringsbuizen in bestektekst	Gereed	Ontwerpleider Integraal waterbeheer	Eddy	

Figuur 47: Detailvenster van een raakvlak in Relatics - tabblad "Algemeen"

Raakvlak informatie

Raakvlak Wegenissen - Rioleringsdoorsteek

Algemeen		Bijkomende informatie	
heeft afspraken			
Afspraak	Datum	Status	
De ontwerpplannen worden ter beschikking gesteld op sharepoint	Today		
heeft risico			
ID	Name	Description	
Ris-00059	⚡ Het PVR beperkt de hoogte van bouwmaterieel tijdens de uitvoering		
heeft besluiten			
Besluit	Status	Datum	
heeft informatief document			
Informatief document	Bestand		

Figuur 48: Detailvenster van een raakvlak in Relatics – tabblad “Bijkomende informatie”

Op de linkerzijde van de raakvlakpagina in Relatics wordt een eenvoudige lijst van alle raakvlakken getoond, inclusief de vermelding van het leidende en rakende object. Het naastliggende tabblad stelt een filter ter beschikking waarmee de raakvlakken gesorteerd kunnen worden volgens hun status. Op die manier zijn de raakvlakken, die nog opgevolgd moeten worden, gemakkelijk te onderscheiden van de reeds afgehandelde en vice versa. De onderstaande figuren geven een beeld van de informatie onder beide tabbladen, links op de raakvlakkenpagina.

Lijst		Filter	
Lijst			
Raakvlak	Leidend object	Rakend object	
⚠ Raakvlak Bredastraat - Brug autoverkeer	🏠 E0570 - Bredastraat	🏠 E0110 - Brug autoverkeer	
⚠ Raakvlak Keermuur - Randbalk	🏠 E0130 - Keermuur toegangshelling	🏠 E0560 - Randbalk	
⚠ Raakvlak Keermuur Oostzijde - Landhoofd	🏠 E0132 - Keermuur Oostzijde	🏠 E0122 - Landhoofd brug openbaar vervoer	
⚠ Raakvlak Keermuur Westzijde - Landhoofd	🏠 E0131 - Keermuur Westzijde	🏠 E0112 - Landhoofd brug autoverkeer	
⚠ Raakvlak Randbalk - Leuning	🏠 E0560 - Randbalk	🏠 E0140 - Leuning	
⚠ Raakvlak Randbalk - Wegkoffer	🏠 E0530 - Noorderlaan	🏠 E0560 - Randbalk	
⚠ Raakvlak Wegenissen - Rioleringsdoorsteek	🏠 E0500 - Wegenissen	🏠 E0232 - Doorsteekbuizen	

Figuur 49: Linkse venster van de raakvlakpagina - tabblad "Lijst"

Lijst		Filter	
Parameters for: Lijst			
Status (use * to find all)		Op te volgen	
		<input type="button" value="Search"/>	
Lijst			
Naam	Leidend object	Rakend object	Status
⚠ Raakvlak Bredastraat - Brug autoverkeer	🏠 E0570 - Bredastraat	🏠 E0110 - Brug autoverkeer	Op te volgen
⚠ Raakvlak Keermuur - Randbalk	🏠 E0140 - Keermuur toegangshelling	🏠 E0560 - Randbalk	Op te volgen
⚠ Raakvlak Randbalk - Leuning	🏠 E0560 - Randbalk	🏠 E0150 - Leuning	Op te volgen
⚠ Raakvlak Randbalk - Wegkoffer	🏠 E0530 - Noorderlaan	🏠 E0560 - Randbalk	Op te volgen

Figuur 50: Linkse venster van de raakvlakpagina - tabblad "Filter"

ACTIVITEIT

In wat voorafging werd uitgelegd hoe activiteiten verbonden zijn met objecten en raakvlakken. Verder worden er nog eisen en verantwoordelijken aan gekoppeld. Dat een activiteit moet voldoen aan (een) bepaalde eis(en) spreekt voor zich. Om te kunnen garanderen dat deze activiteit tot een goed einde wordt gebracht, is het belangrijk dat één persoon de eindverantwoordelijkheid draagt. Zoals vermeld in [§4.2.3 - Raakvlakkenbeheer](#) zijn handelingen, die in het systeem gedefinieerd worden als activiteiten, zaken die gedurende een langere termijn lopen. Het is dan ook niet ondenkbaar dat projectmedewerkers in de loop van die periode een andere functie krijgen, of zelfs overstappen naar andere projecten. Om te vermijden dat het verdwijnen van een persoon uit het project aanleiding geeft tot activiteiten zonder verantwoordelijken, worden activiteiten in de eerste plaats toegekend aan een 'rol'.

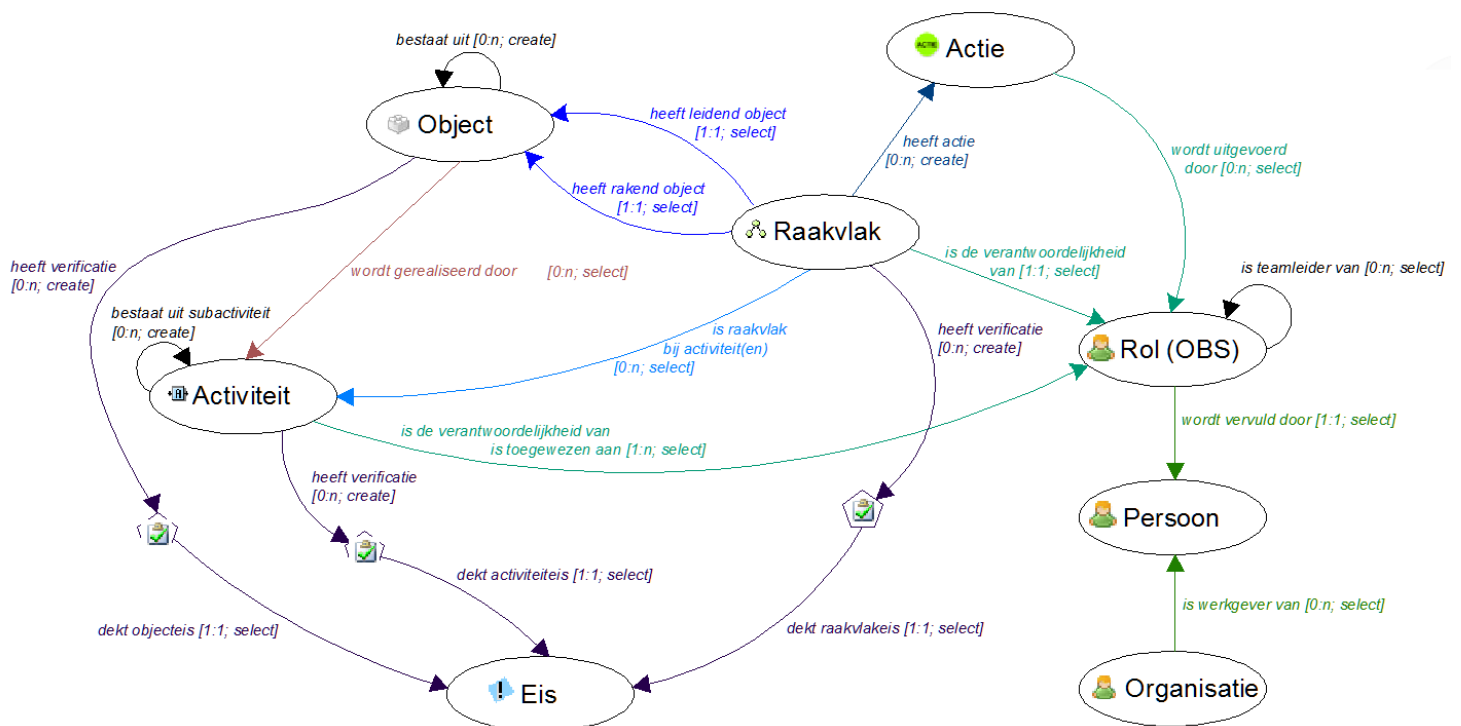
ACTIE

Hoewel een actie meestal van korte duur is, kan het toch zinvol zijn de uitvoering ervan te koppelen aan een 'rol' in plaats van aan een 'persoon'. De reden hiervoor is eerder al aangehaald.

ROL

Een rol komt in principe overeen met een functie binnen het project (bv. "ontwerpcoördinator"), en wordt steeds vervuld door exact één persoon.

Wanneer al de bovenstaande elementen samengevoegd worden, ontstaat het onderstaande **schema** van de Relaticsomgeving.⁷ Er wordt nogmaals opgemerkt dat het hier enkel gaat om het deel van het systeem dat nodig is voor het beheer van (geometrische) raakvlakken. Dit schema zal zeker niet volstaan om een volledig projectbeheersysteem op te zetten. Bij het opstellen van de structuur is er echter steeds over gewaakt dat het voorgestelde schema naadloos in een volledig projectbeheersysteem kan worden inpast.



Figuur 51: Semantische structuur voor raakvlakkenbeheer in Relatics

⁷ De informatie tussen vierkante haken leggen bij elke relatie de zgn. 'cardinality' en de selectiemogelijkheden (select/create) vast. Het eerste cijfer bepaalt hoeveel elementen de relatie minimaal moet hebben; het volgende cijfer geeft het maximaal aantal elementen van een relatie weer ("n" staat voor oneindig veel elementen).

4.4 Besluit

Met behulp van een N²-chart kan op een systematische manier worden nagegaan hoe de verschillende componenten van een ontwerp met elkaar interageren. Hierbij is het mogelijk de kritische objecten te bepalen en vast te leggen welke objecten leidend of rakend zijn. Het opstellen van een N²-chart is echter arbeidsintensief en tijdrovend, en zal dus enkel gebeuren in geval van twijfel of conflicten. Voor de meeste raakvlakken kan met gezond verstand of op basis van ervaring bepaald worden welke objecten best rakend of leidend gesteld worden.

Eens de raakvlakken binnen een ontwerp gedetecteerd zijn, is het belangrijk deze te blijven opvolgen tijdens het hele project. Dit kan bereikt worden door de raakvlakken in een databankomgeving op te slaan en er acties aan te koppelen. Op de bijgevoegde cd-rom is een demo beschikbaar die meer tekst en uitleg geeft bij de manier waarop het raakvlakmanagement geïmplementeerd kan worden in Relatics.

In de databankomgeving wordt een raakvlak steeds gedefinieerd tussen twee concrete objecten en gekoppeld aan activiteiten die met dit raakvlak rekening moeten houden. Verder is het belangrijk dat er juist één eis aan een raakvlak toegekend wordt.

Om te vermijden dat elementen in de databank verloren gaan doordat bijvoorbeeld hun verantwoordelijke verdwijnt of verandert, worden geen personen maar rollen gekoppeld aan de elementen in Relatics. De raakvlakmanager is het best geplaatst om een overzicht te houden over de hele problematiek in verband met raakvlakken.

Uiteraard is het ook belangrijk dat op verscheidende momenten in het proces een verificatie van de tot dan toe afgewerkte acties en activiteiten wordt ingebouwd. Clash detection software kan daarbij helpen om fouten in de ontwerpplannen alsnog op te sporen.

In [hoofdstuk 5 - Praktisch project](#) zal de systematiek voor het beheer van geometrische raakvlakken die in dit hoofdstuk is uiteengezet, toegepast worden op een voorbeeld uit de praktijk.

HOOFDSTUK 5

Praktisch project

De beheermethode die in [hoofdstuk 4 - Raakvlakkenbeheer](#) is uiteengezet, moet natuurlijk in de praktijk bruikbaar zijn. In dit hoofdstuk wordt aan de hand van een voorbeeld uit de realiteit geïllustreerd op welke manier dit kan gebeuren.

Het is natuurlijk onmogelijk om binnen het bestek van deze masterproef het volledige project uit te werken volgens de voorgestelde beheermethode. Daarom is ervoor gekozen een aantal cases uit het project te bespreken en op basis daarvan de methode verder toe te lichten. Daarnaast worden de clash detection functionaliteiten van enkele softwareprogramma's uit [hoofdstuk 3 – 3D modellering](#) toegepast op deze cases. De bijgevoegde cd-rom bevat een demofilm waarmee de werking van twee clash detection programma's (Navisworks Manage & Projectwise Navigator) wordt toegelicht.

Op basis van de uiteenzettingen in deze masterproef is een artikel geschreven rond het praktisch project. Dit artikel is achteraan in de masterproef toegevoegd, en is ook in digitale versie beschikbaar op de begeleidende cd-rom.

5.1 Het project

Het project waarvan verder enkele onderdelen besproken worden, omvatte de heraanleg van de Noorderlaanbruggen over het Albertkanaal in Antwerpen. Dit project liep tussen februari 2008 en mei 2010 en vormt het eerste afgewerkte waterwegenproject uit het Masterplan Mobiliteit Antwerpen. Het doel van de heraanleg was om de geplande verbreding van het Albertkanaal mogelijk te maken, en meteen de bruggen te verhogen om dit kanaal toegankelijker te maken voor grotere schepen.

De onderstaande **figuur** geeft een idee van de situering van de werkzaamheden [28].



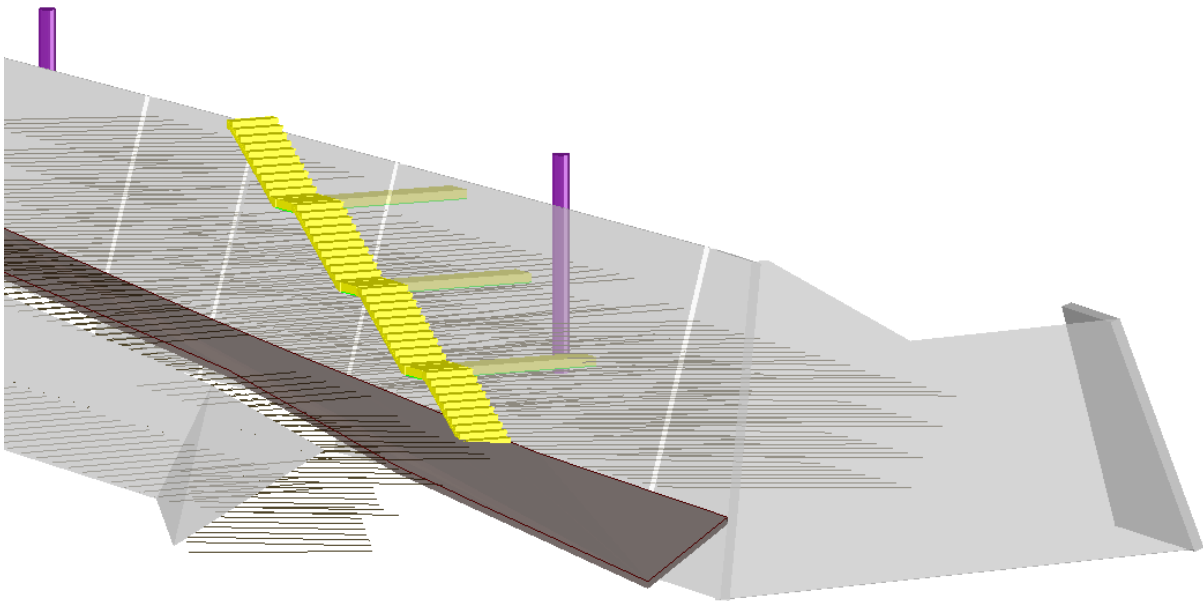
Figuur 52: Situering project Noorderlaanbruggen [28]

De nieuwe brug bestaat uit twee naast elkaar gelegen brugdekken, waarvan er één voorzien is voor het openbaar vervoer en één voor het autoverkeer. De westelijk gelegen brug voor het autoverkeer is de breedste en telt twee rijstroken in elke richting. Voor het openbaar vervoer is een brug voorzien met twee tramsporen. Beide brugdekken bestaan uit drie overspanningen: een middenoverspanning over het Albertkanaal met een lengte van 80 meter, en twee zijoverspanningen over elke oever van het kanaal van telkens 50 meter. De onderstaande **foto** geeft een impressie van de afgewerkte Noorderlaanbruggen. [29]



Figuur 53: De Noorderlaanbruggen in afgewerkte toestand

Het spreekt voor zich dat in een dergelijk project verschillende raakvlakken bestaan. Het gaat immers niet enkel om de realisatie van twee brugdekken; ook de rioleringen, de wegenissen van de toegangshellingen, verschillende technieken als de straatverlichting, de trambaan ... moesten aangelegd worden. In de landhoofden van beide bruggen komen bijvoorbeeld heel wat ontwerpdelen uit verschillende disciplines samen, gaande van rioleringsbuizen tot wapeningsstrippen voor grondkerende wanden. Het is van het grootste belang een goed inzicht te hebben in de manier waarop deze objecten met elkaar interageren. Uit de onderstaande **figuur**, waarop het landhoofd getoond wordt met daarin een aantal constructie-elementen, blijkt dat het zeker zonder 3D-voorstelling niet vanzelfsprekend is om een overzicht over alle interacties te krijgen.



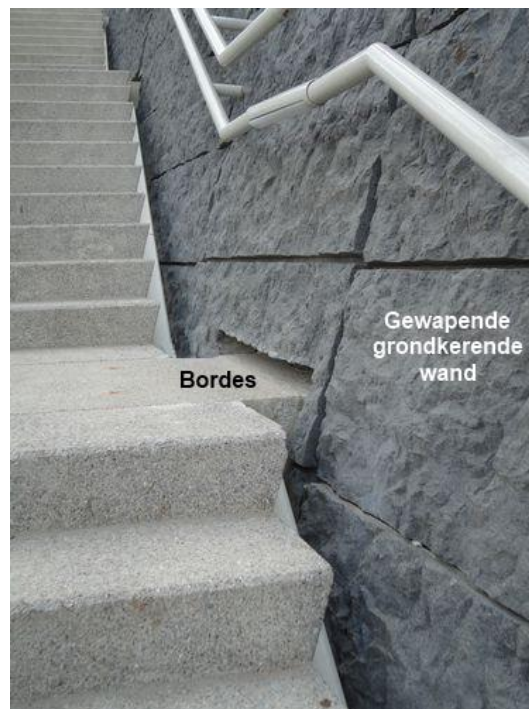
Figuur 54: Landhoofd van de Noorderlaanbruggen

In eerste instantie is het project op een klassieke manier ontworpen. De behoefte aan een 3D-modellering werd echter steeds groter, zodat er uiteindelijk toch voor geopteerd is verschillende ontwerpdelen in 3D te controleren. Hierdoor zijn een aantal conflicten in de ontwerpplannen naar voor gekomen die tot belangrijke faalkosten en een achterstand in de planning hadden kunnen leiden. Hieronder volgt een kort overzicht van een aantal cases waar met behulp van de 3D-benadering raakvlakproblemen zijn ontdekt. Zoals reeds eerder uitgelegd, worden raakvlakken idealiter in een vroeg stadium opgespoord en door middel van een geïntegreerd beheerproces gemanaged. Het ontstaan van raakvlakproblemen in de ontwerpplannen kan dus niet alleen achteraf gecorrigeerd worden, maar moet ook op voorhand worden vermeden. Voor elke case wordt een deel van de voorgestelde oplossing om raakvlakken efficiënt te beheren nader toegelicht.

5.2 Case 1 – Gewapende grondkerende wanden

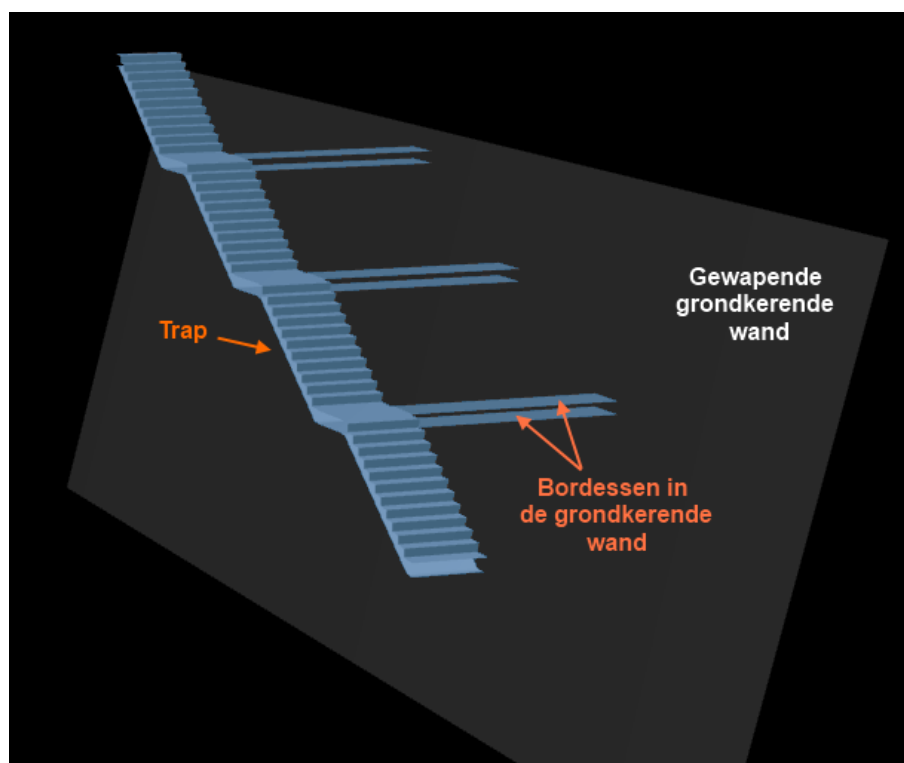
Op de bovenstaande foto (figuur 53) is duidelijk te zien dat de sterke hellingen aan de zijkanten van de landhoofden gerealiseerd zijn met gewapende grondkerende wanden. Er zijn ook telkens trappen voorzien om

voetgangers en fietsers gemakkelijk toegang te verlenen tot de oevers van het Albertkanaal. De grondkerende wanden waren niet eenvoudig te berekenen, noch uit te voeren. De driedimensionale helling van deze wanden zorgt er immers voor dat bepaalde aspecten gemakkelijk verkeerd berekend worden. Bovendien moeten de wapeningsstrippen van deze wanden voldoende diep in het landhoofd steken, waar ze kunnen raken met andere objecten (bijvoorbeeld de fundering van de verlichtingspalen). Naast het realiseren van de wanden zelf, moet er ook rekening mee gehouden worden dat er genoeg openingen voorzien zijn voor de bordessen waar de trappen op steunen. Vermits de wanden geheld zijn in drie richtingen, moeten de trappen deze kromming volgen en zijn ook de doorsteken van de bordessen geheld. Tussen de trap en de keermuur is een speling van 3 cm voorzien, op de doorsteken moet boven-en onderaan nog 2 cm speling aanwezig zijn. Op de onderstaande foto is duidelijk te zien hoe de trap steunt op bordessen die in de grondkerende wand steken.



Figuur 55: Trap met bordes

Dit alles impliceert dat een tweedimensionaal ontwerp quasi onvermijdelijk tot onjuistheden zal leiden. Het is bovendien onbegonnen werk de driedimensionale krommingen handmatig uit te rekenen. Om de continu veranderende helling van de treden te kunnen bepalen, en daaruit ook de positie en helling van de doorsteken voor de bordessen te berekenen, is een 3D-model van de trap gemaakt. De onderstaande **figuur** toont dit driedimensionaal model.

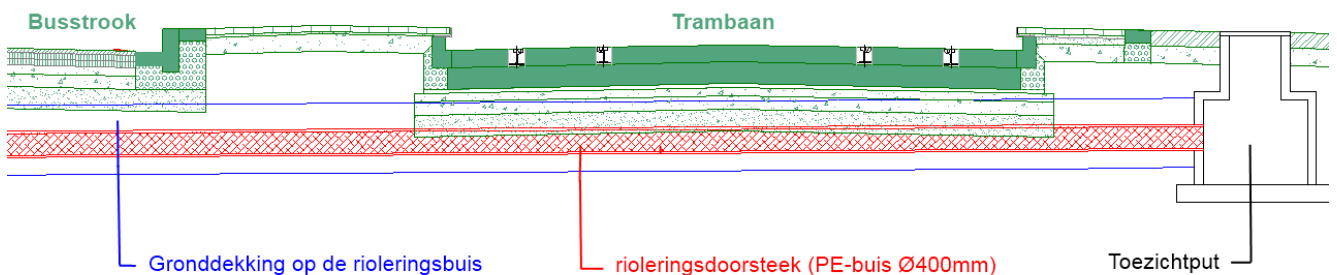


Figuur 56: 3D-model van de trap met de bordessen

5.3 Case 2 - Rioleringsdoorsteek

Zoals hierboven in §5.2 – Gewapende grondkerende wanden is aangetoond, is het gebruik van 3D-modellering onontbeerlijk bij geometrisch complexe projecten. Aan de hand van deze case zal aangegeven worden hoe 3D-modellen gebruikt kunnen worden om na te gaan of de verschillende ontwerpplannen op elkaar afgestemd zijn.

Voor dit project zijn het hele rioleringsnet en de wegenissen van de Noorderlaan in één 3D-model samengebracht. Op die manier zijn op elke dwarsdoorsnede van de wegenissen ook de rioleringsbuizen te zien. Hierdoor wordt het mogelijk op een eenvoudige manier na te gaan of de hoofdrioleringen, doorsteken, toezichtputten ... voldoende diep onder de wegwakker zitten. De toezichtputten worden ontworpen en geprefabriceerd volgens de opgegeven BOK-peilen (peil aan de binnenzijde van de buis, onderaan). Door op regelmatige afstanden dwarsdoorsneden te maken, werd duidelijk dat de opgegeven peilen niet realiseerbaar waren. Eerst en vooral bleek dat de bovenzijde van de doorsteek tussen de toezichtputten (PE-buis Ø 400 mm) door de fundering van de trambedding kwam. Deze trambedding is uiteraard dieper gefundeerd dan de wegenis, waar de buis niet door de fundering stak. De gronddekking op de doorsteek bleek wel in de funderingslaag van de autoweg te zitten. Beide problemen zijn duidelijk voorgesteld op de volgende **figuur**.

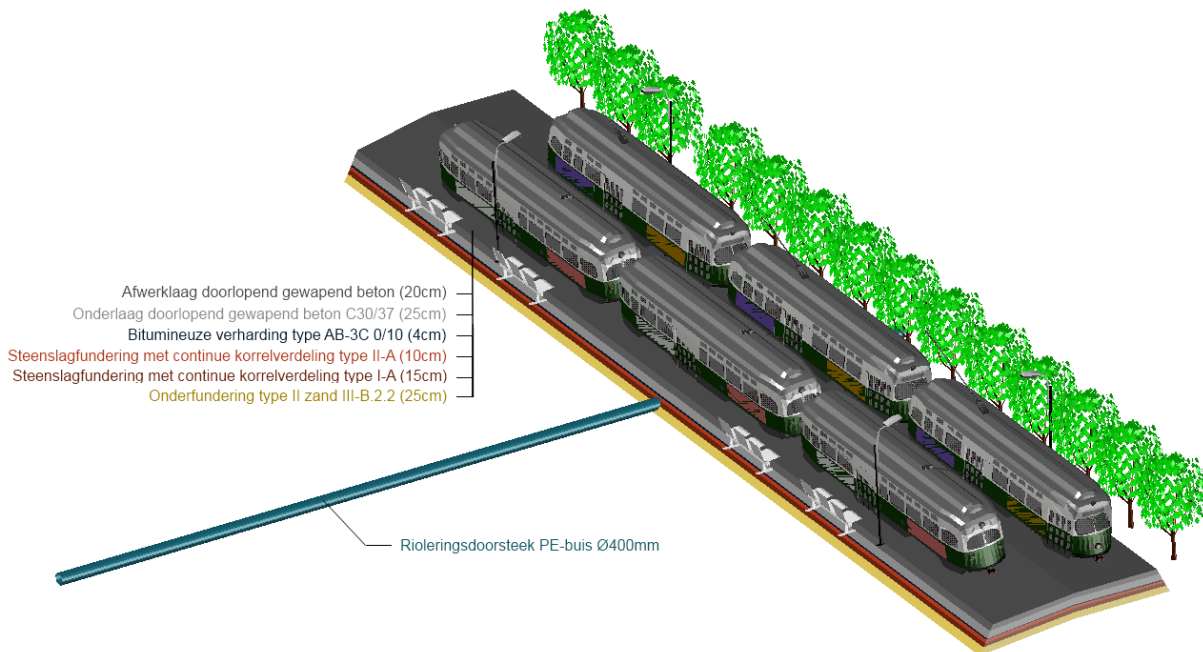


Figuur 57: Situatieschets rioleringsdoorsteek

Bij het leggen van de fundering voor de wegen en de trambedding zou dit raakvlak dus voor grote problemen zorgen. Niet alleen moeten de funderingswerken worden stilgelegd, ook moeten de toezichtputten opnieuw besteld en aangelegd worden. Dit betekent meteen ook een achterstand in de planning, vermits de werken moeten stilgelegd worden en de levering van nieuwe toezichtputten ook een zekere leveringstijd vraagt.

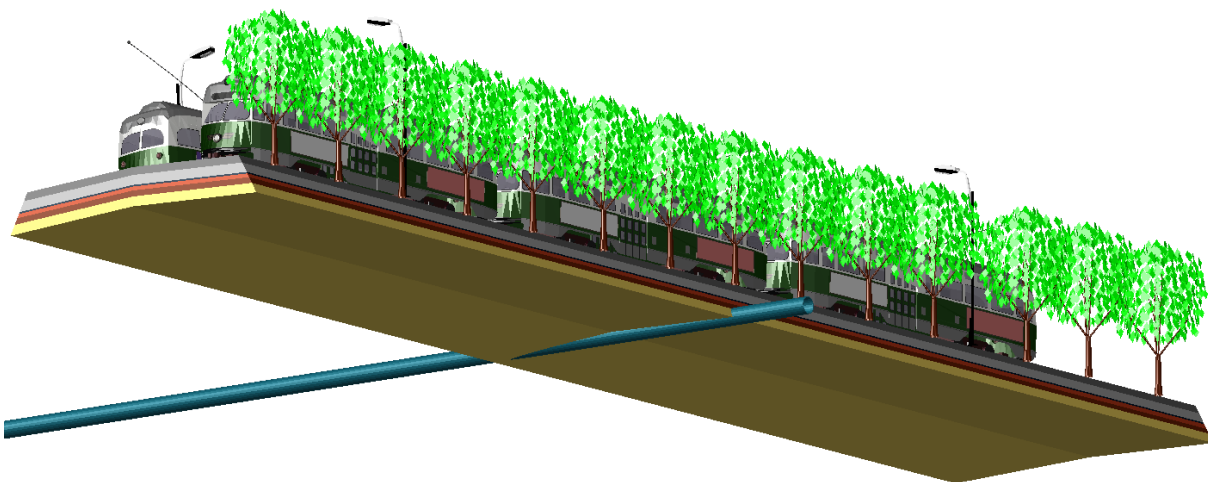
De onderstaande **figuur** toont het 3D-model met een opbouw van de trambaan en de rioleringsdoorsteek. Een basisversie van dit model wordt gebruikt om een clash detection met behulp van enkele softwarepakketten uit te voeren. Vermits in België meestal gewerkt wordt met pakketten van

Autodesk of Bentley, zullen voor deze test enkel Navisworks en Navigator gebruikt worden (zie ook demofilm op de bijgevoegde cd-rom).



Figuur 58: 3D-model van de rioleringsdoorsteek

In het model hierboven zit enkel de lagenopbouw van de trambedding zonder de onderliggende grond. De onderstaande **figuur** illustreert dat de rioleringsbuis wel degelijk door de onderfundering en de fundering van de trambaan loopt.



Figuur 59: Raakvlak tussen de rioleringsdoorsteek en de trambedding

5.3.1 Autodesk Navisworks

Het model kan op een eenvoudige manier in de Navisworks-omgeving ingeladen worden. In dit geval zitten de rioleringsbuis en het grondpakket in hetzelfde bestand, maar in de praktijk is het ook mogelijk om meerdere modellen samen te voegen.

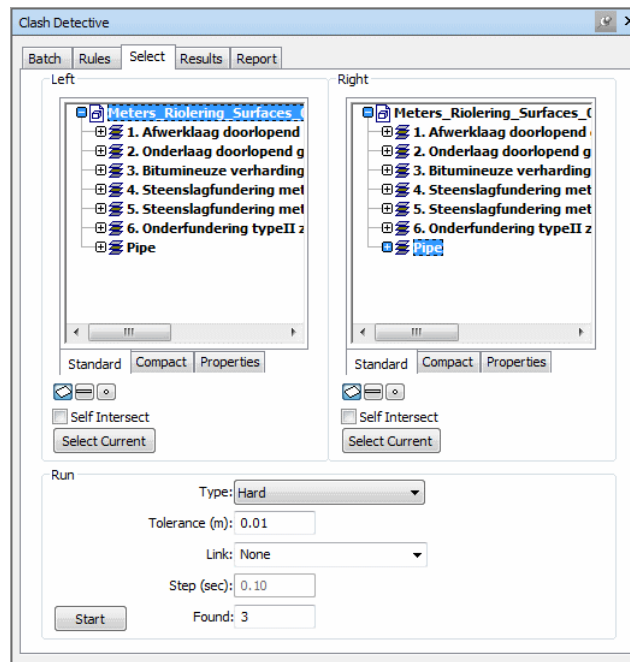
In de clash detection test kan, zoals in [hoofdstuk 3 – 3D-modellering](#) is uitgelegd, gekozen worden welke onderdelen onderling gecontroleerd moeten worden. Aangezien het in dit voorbeeld enkel gaat om het raakvlak tussen het lagenpakket van de trambaan en de rioleringsdoorsteek, is ervoor gekozen om het hele model tegen de rioleringsbuis te testen.

De PE-buis heeft een dikte van 2 cm, wat betekent dat de tolerantie voor de test zeker niet groter mag gekozen worden. Met een tolerantie van 1 cm zou de clash normaal gedetecteerd moeten worden. In de praktijk is uiteraard niet altijd op voorhand te voorspellen waar zich eventueel raakvlakproblemen in het ontwerp zullen voordoen. De tolerantie moet dan oordeelkundig gekozen worden: voor beton lijkt een waarde van 1 cm vrij streng, in geval van een staalconstructie is dit al eerder groot. Bij het kiezen van de tolerantie moet er bovendien op gelet worden dat de eenheden van het model correct zijn ingesteld. Vermits Navisworks standaard de eenheden op millimeter vastlegt, zal een model dat in meter is getekend met een verkeerde schaal beoordeeld worden. Een tolerantie van 1 cm in Navisworks zou dan overeenkomen met 10 m in werkelijkheid. Hierdoor zal het programma naar alle waarschijnlijkheid geen clashes detecteren.

Vermits in dit model geen problemen zullen ontstaan met randeffecten ten gevolge van niet-overlappende driehoeken (zie [§3.2.2 - BIM software](#)), wordt een 'hard' type clash detection gekozen.

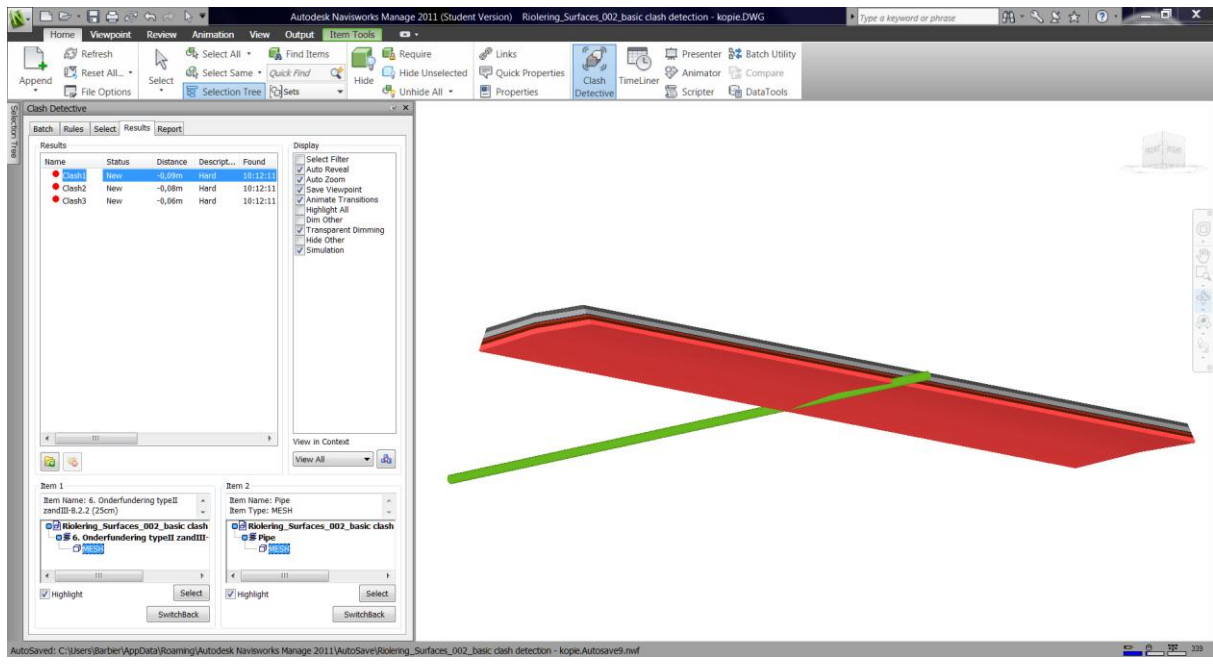
De onderstaande [figuur](#) toont de gebruikte instellingen voor de clash detection test. Op de begeleidende cd-rom is ook het Autocad-model van de rioleringsdoorsteek toegevoegd, een proefversie van Navisworks Manage is beschikbaar op de website van Autodesk.⁸

⁸ Beschikbaar op <http://usa.autodesk.com/navisworks/trial/>.



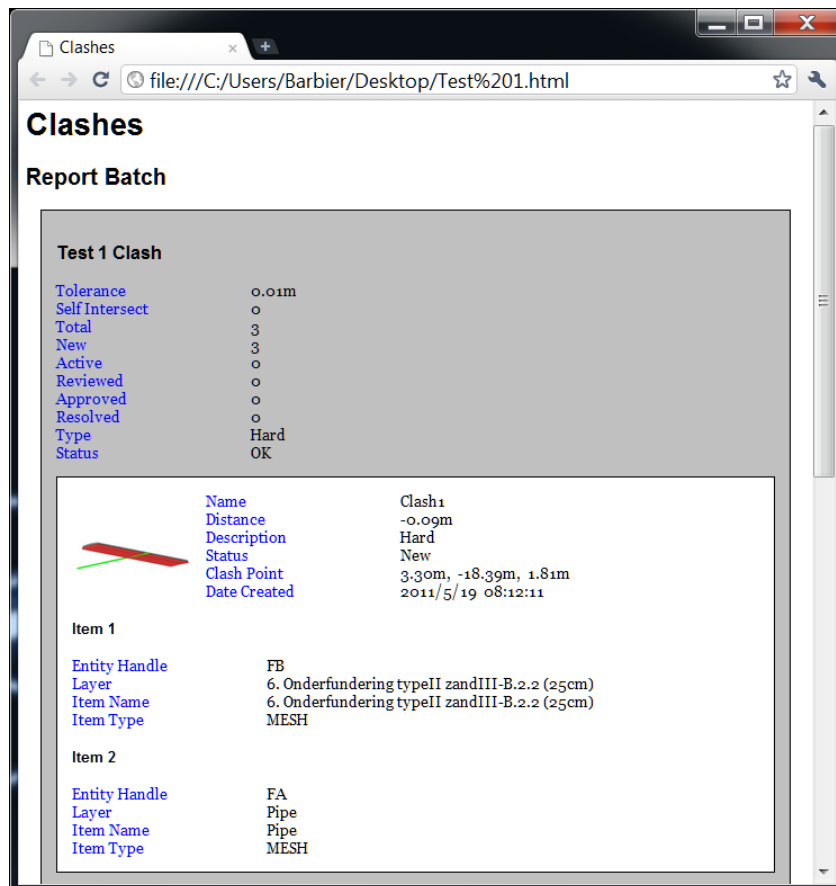
Figuur 60: Instellingen clash detection test in Navisworks

Na het doorlopen van de clash detection test worden in Navisworks drie clashes gevonden. Deze raakvlakproblemen worden gevormd door de drie onderste lagen van de trambedding die de PE-buis kruisen. [Figuur 61](#) toont een van de gedetecteerde clashes. Daarbij kunnen beide objecten in de clash met een andere kleur aangeduid worden; de objecten die met de clash niets te maken hebben kunnen gedimd worden.



Figuur 61: Resultaat van de clash detection test in Navisworks

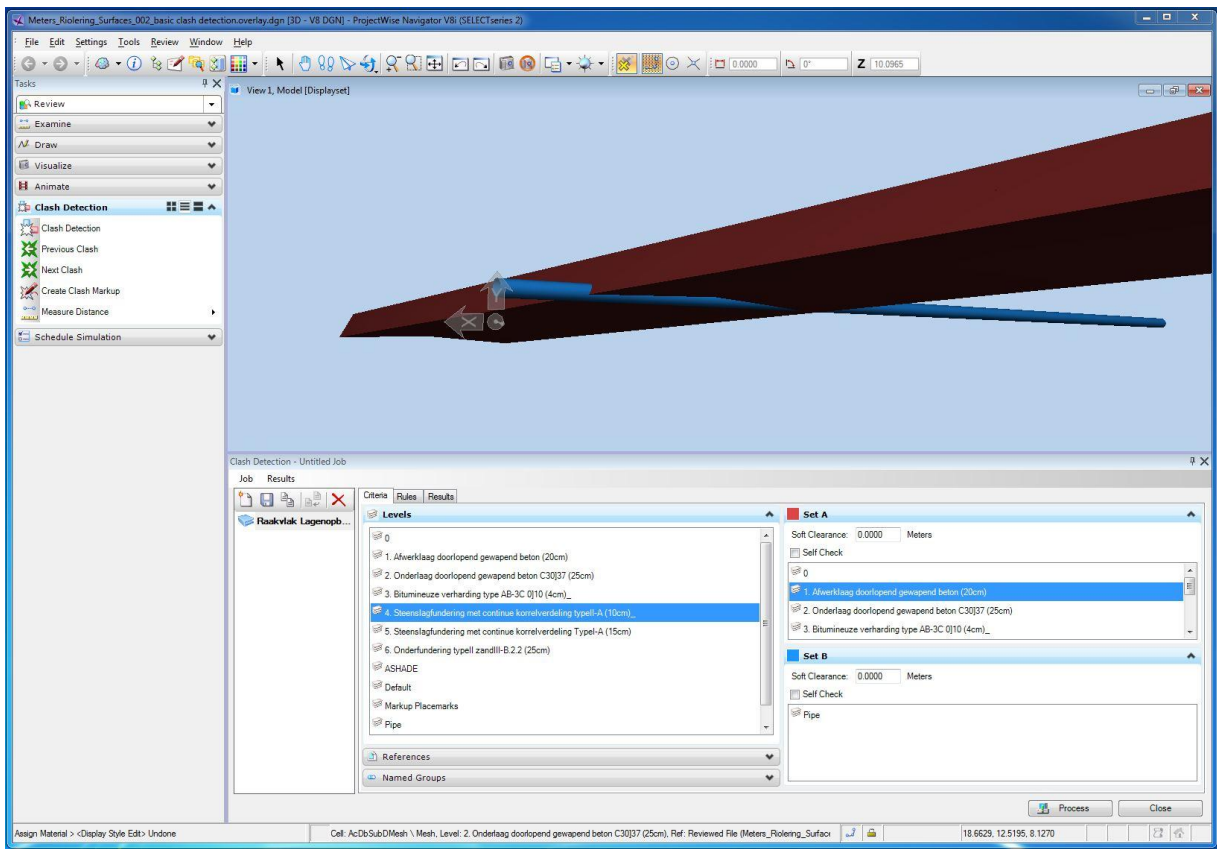
Zoals uitgelegd in §3.2.2 – BIM software, kunnen op basis van de clash resultaten automatisch rapporten gegenereerd worden in verschillende bestandsformaten. De gebruiker kan dit rapport in beperkte mate aanpassen door er meer of minder gegevens in op te nemen. Navisworks zal bij elke clash standaard een zo goed mogelijk beeld van het probleem trachten weer te geven, maar de gebruiker kan dit natuurlijk aanpassen. Het door de gebruiker ingestelde gezichtspunt wordt mee opgenomen in het rapport van de test. Voor deze clash detection test is een rapport in HTML-formaat opgenomen in bijlage (zie cd-rom). De voordelen van de rapportagemogelijkheden in Navisworks zijn aanzienlijk, alleen al omdat de communicatie tussen de verschillende betrokkenen op een eenduidige en vlotte manier kan verlopen. Verder is het mogelijk dergelijke rapporten ook in de gebruikte databankomgeving in te laden. De onderstaande figuur toont een deel van het clash detection rapport ter illustratie.



Figuur 62: Clash detection test rapportage in HTML-formaat

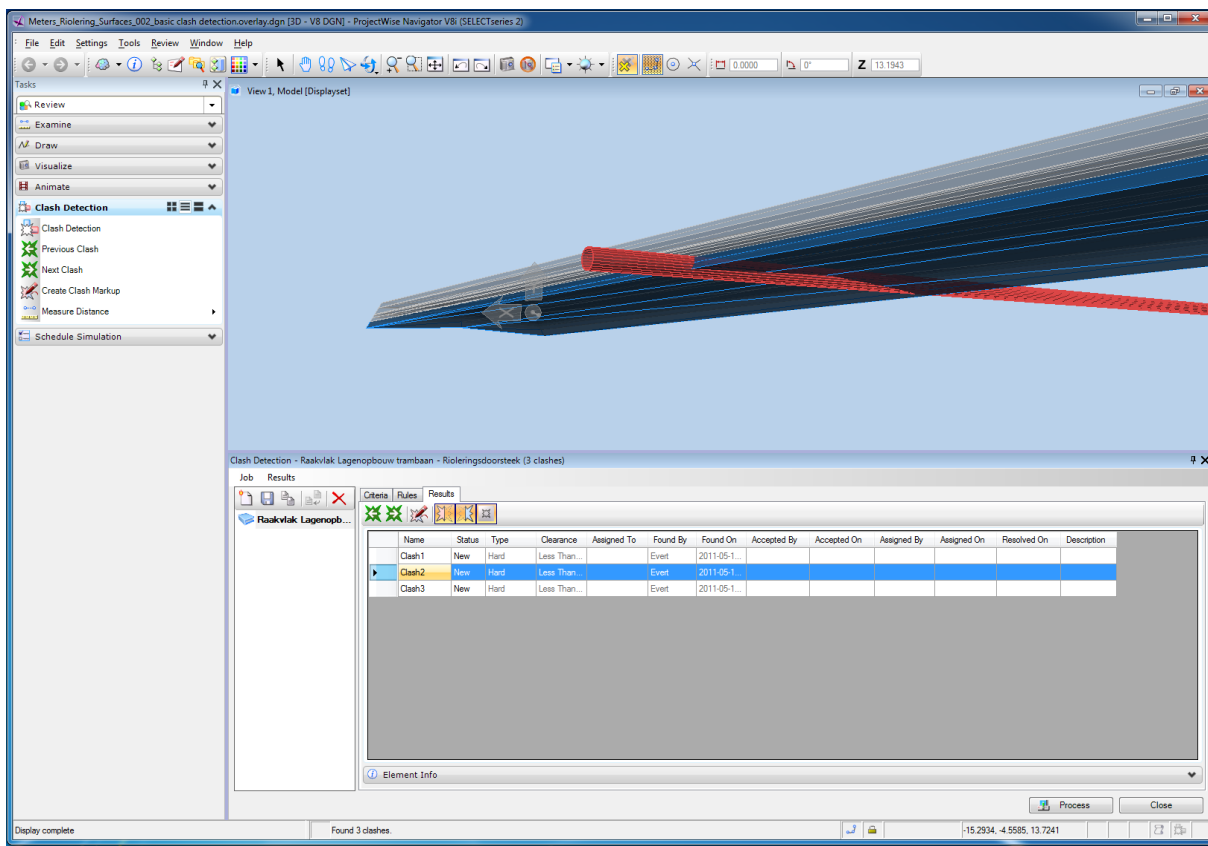
5.3.2 Projectwise Navigator

Net als Navisworks laadt ook Navigator het model zonder problemen in. Om de clash detection uit te voeren, wordt een nieuwe 'Job' aangemaakt, ook dit keer wordt het volledige lagenpakket tegen de doorsteekbuis gecontroleerd. De toleranties worden op dezelfde manier ingesteld als hierboven in Navisworks. De volgende **figuur** toont een beeld van de instellingen voor de 'Clash Detection Job' in Navigator. Alle objecten die in 'set A' zitten worden rood gekleurd, terwijl de objecten in 'set B' een blauwe kleur krijgen.



Figuur 63: Instellingen voor de clash detection job in Navigator

Als de clash detection ingave verwerkt is, vindt ook Navigator dezelfde drie clashes terug. De onderstaande figuur toont één van de gedetecteerde clashes. De objecten die niet voorkomen in de clash worden ook hier gedimd weergegeven.



Figuur 64: Resultaat van de clash detection test in Navigator

Net zoals Navisworks toont Navigator verschillende velden waarin kan worden aangegeven of de clash al in behandeling is of niet, wanneer hij gedetecteerd is enz.

Zoals vermeld in §3.2.2 - BIM software, is een van de grote gebreken van Projectwise Navigator dat de resultaten van de clash detection niet geëxporteerd kunnen worden naar een automatische rapportage zoals bij Navisworks.

5.4 Case 3 - Randbalk

Zoals in [figuur 53](#) te zien is, vertrekken vanaf beide landhoofden verschillende gewapende grondkerende wanden. Deze moeten bovenaan aansluiten op een randbalk die naast het fietspad ligt. De randbalk is duidelijk zichtbaar op onderstaande [foto](#). Zoals in §5.2 – Case 1 – Gewapende grondkerende wanden is uitgelegd, waren de grondkerende

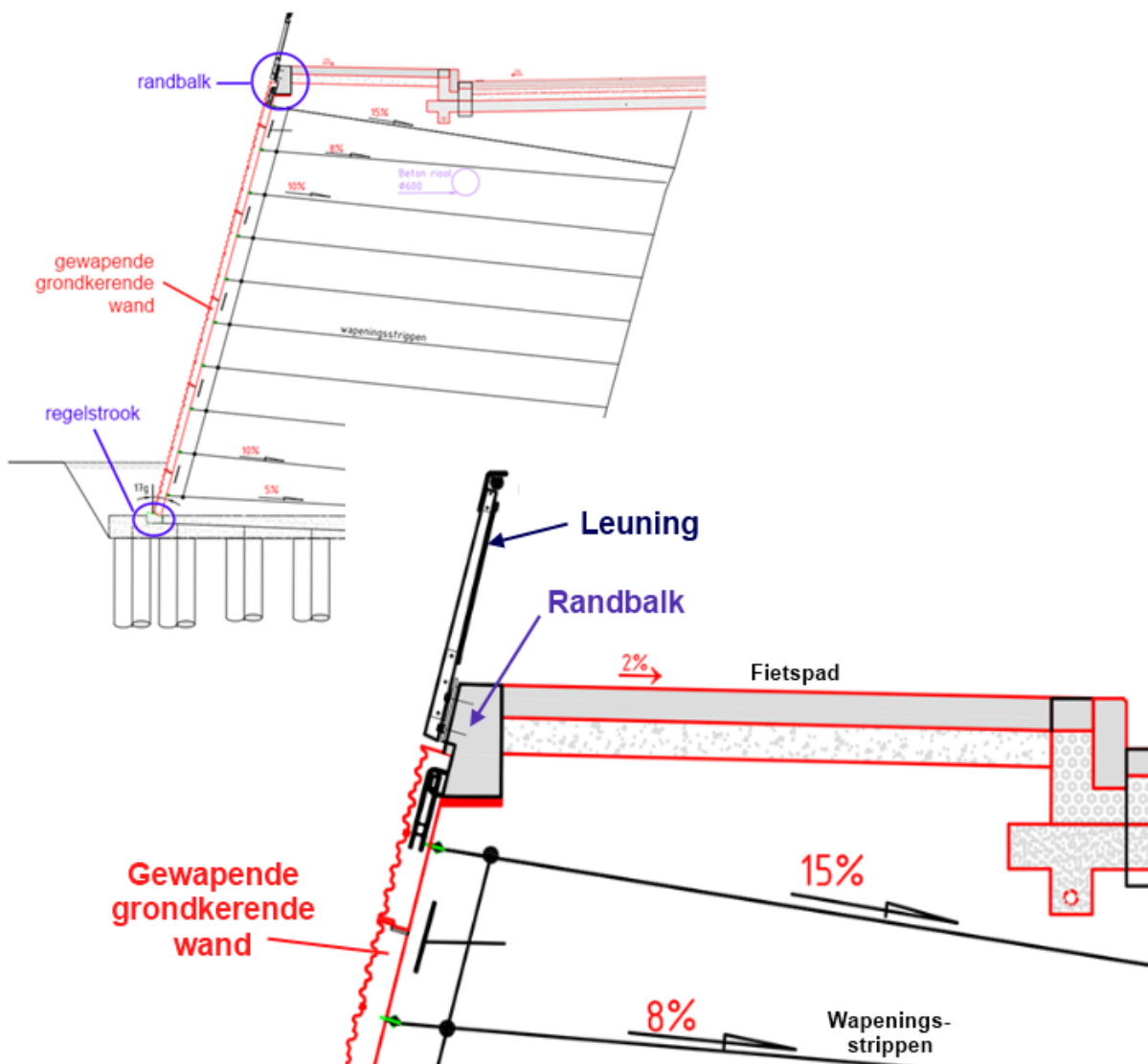
wanden waren niet eenvoudig te berekenen omwille van hun driedimensionale helling.



Figuur 65: Foto randbalk

Het niveau van de grondkerende wanden bovenaan wordt bepaald door de regelstrook aan de voet van deze wanden. Het peil van deze stroken was echter op sommige plaatsen te hoog, op andere plaatsen te laag. Een te hoog aangezette regelstrook zorgt ervoor dat de gewapende grondkerende wanden te hoog toekomen op de randbalken, waardoor er niet genoeg ruimte overblijft om de leuning te bevestigen. Plaatselijk de randbalk verhogen biedt geen oplossing voor dit probleem: het is immers de bedoeling dat de leuning doorloopt over de volledige brug zonder hoogteverschillen. De randbalk mag ook niet te laag komen, omdat dit nadelig zou zijn voor de afwatering van het fietspad. Het **plan** hieronder geeft een idee van de beschreven situatie.

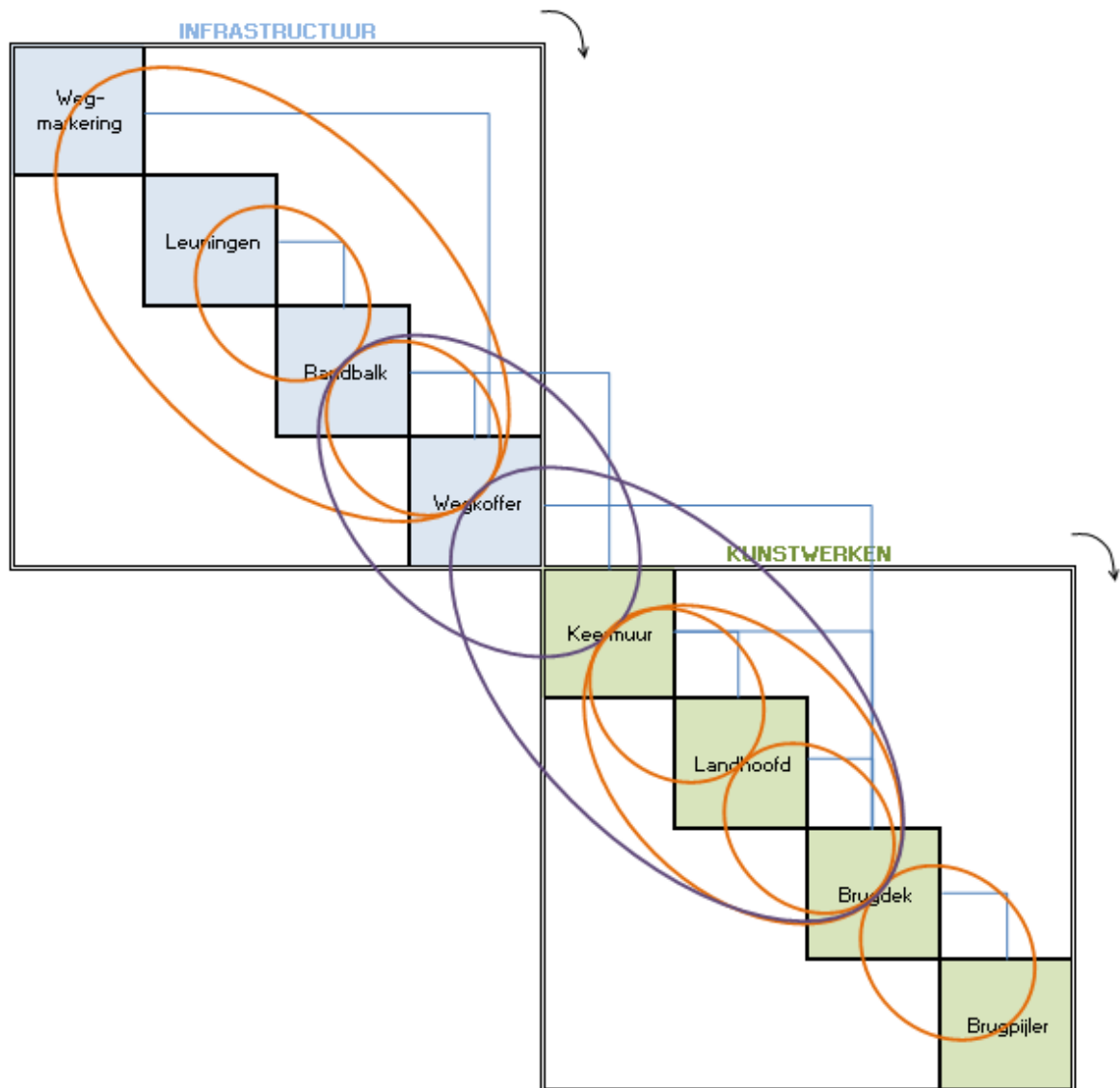
Om ervoor te zorgen dat een eventuele ontwerpfout uit de ene discipline niet meegenomen wordt in de berekening van een ander object, is als volgt te werk gegaan. Zoals hierboven vermeld, zijn de wegenissen volledig in 3D getekend. Uit dit model kunnen gemakkelijk de hoogtecoördinaten van de randbalk worden afgelezen. Vervolgens is vanaf de onderzijde van de regelstrook aan de grondkerende wand naar boven toe gerekend. In principe zou er geen verschil mogen bestaan tussen het niveau van de randbalk dat op deze manier is berekend, en het peil uit het 3D-model.



Figuur 66: Plan randbalk

Zoals in §4.1 – N²-chart beschreven, is het mogelijk voor complexe interacties een N²-diagram op te stellen en zo te bepalen welke objecten leidend en rakend zijn. De N²-chart die toen is opgesteld, wordt hieronder hernomen. Verwacht wordt dat de randbalk een kritisch object zal vormen, vermits het raakvlakken heeft met de leuning, de aangrenzende wegeis en de grondkerende wand. Uit de N²-chart blijkt bovendien dat één van deze drie raakvlakken interdisciplinair is (de paarse controlelus tussen de randbalk en de keermuur). Intuïtief zou kunnen besloten worden de randbalk steeds als rakend object te beschouwen. Voor de interactie tussen de leuning en deze balk lijkt het op het eerste gezicht wel logischer een uitzondering te maken en de randbalk als leidend te kiezen. Dit betekent echter dat het ontwerp van de leuning de aanpassingen aan het ontwerp van de randbalk moet volgen. In de praktijk is dit niet het geval, wat betekent dat de randbalk voor de drie raakvlakken steeds als rakend object kan beschouwd worden.

Het is belangrijk de informatie die met behulp van deze N²-chart bepaald is, ook vast te leggen in een databankomgeving en er daar verdere acties aan te koppelen. Voor de volgende case is de implementatie in een relationele databank verder uitgewerkt.



Figuur 67: N²-chart voor de randbalk

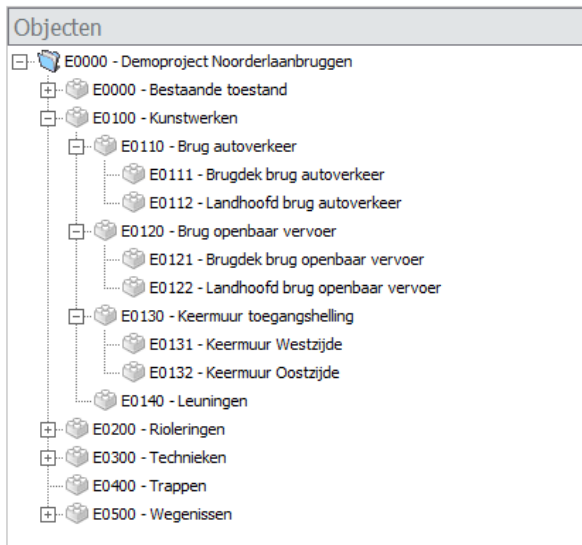
5.5 Case 4 - Landhoofd

De gewapende grondkerende wanden moeten uiteraard ook aansluiten op de betonnen landhoofden van de brug, zoals ook te zien is op [figuur 53](#).

Het project is oorspronkelijk volledig in 2D ontworpen, wat betekent dat van het betonnen landhoofd zowel een grondplan als vooraanzicht gemaakt is. Voor het ontwerp van de keermuur is vertrokken van het grondplan, terwijl de uitvoeringsplannen voor het landhoofd gebaseerd zijn op het vooraanzicht. Na een controle in 3D bleek dat het vooraanzicht van het landhoofd niet exact overeenstemde met het grondplan. Dit had als resultaat dat bij het eerste ontwerp de keermuur maar liefst 30 cm *in* de zijkant van het landhoofd zou eindigen. Dit illustreert opnieuw het belang van een driedimensionaal ontwerp, zeker bij geometrisch complexe projecten.

Op basis van dit voorbeeld zal aangetoond worden hoe de Relaticsomgeving, beschreven in [§4.3 - Implementatie van het proces in Relatics](#), gebruikt kan worden om dit raakvlak te beheren. Door het detecteren en opvolgen van raakvlakken te integreren in een volledig beheerproces, kunnen dergelijk problemen vermeden worden.

Het databeheersysteem vertrekt van een (beperkte) objectdecompositie van het systeem, zoals weergegeven in de onderstaande [boomstructuur](#). Deze decompositie mag niet te ver doorgedreven worden en moet altijd functioneel blijven. Een opsplitsing tot het niveau van de individuele grondkerende wandpanelen is weinig relevant en zal de projectomgeving enkel logger maken.



Figuur 68: Objectenboom in Relatics

Zoals werd uitgelegd in §5.2 – Case 1 - Gewapende grondkerende wanden is het object “Keermuur toegangshelling” rakend voor het raakvlak tussen het landhoofd en de keermuur. Dit wordt weergegeven in het detailscherm van het object, zoals te zien is in de volgende figuur.

Object			
E0140 - Keermuur toegangshelling			
Algemeen		Eisen	
Verificaties			
Algemeen			
ID	Obj-03484		
Naam	E0140 - Keermuur toegangshelling		
Omschrijving			
Objecttype			
Locatie			
heeft betrokken organisatie			
Organisatie			
Antwerpse Bouwwerken			
Architectenbureau			
INFRANEA			
is leidend voor Raakvlak		is rakend voor Raakvlak	
Naam	Status	Eis	
⚠ Raakvlak Keermuur - Randbalk	Op te volgen	⚠ Randbalkhoogte	
Naam	Status	Eis	
⚠ Raakvlak Landhoofd - Keermuur toegangshelling	Afgehandeld	⚠ Aansluiting keermuur op landhoofd	
wordt gerealiseerd door			
ID	Activiteit	Fase	
Act-02301	➡ Ontwerpplannen keermuur	Ontwerp	
Act-02300	➡ Stabiliteit keermuur	Ontwerp	

Figuur 69: Detailscherm object "Keermuur toegangshelling"

Bij het raakvlak “Landhoofd - Keermuur toegangshelling” wordt ook aangegeven welke activiteiten rekening moeten houden met dit raakvlak.

Figuur 70 toont het detailscherm van het raakvlak.

Raakvlak informatie				
Raakvlak Landhoofd - Keermuur toegangshelling				
Algemeen			Bijkomende informatie	
Algemeen				
Naam	Raakvlak Landhoofd - Keermuur toegangshelling			
ID	RKVLK_0058			
Omschrijving				
Status	Afgehandeld			
Fase signalisatie	Ontwerp			
Interdisciplinair	Ja			
Dient te voldoen aan				
Eis	ID	Eistekst	Categorie	Verificatie
! Aansluiting keermuur op landhoofd	E&R - 01153	De helling van de wapeningsplaten van de keermuur moet aansluiten op de helling van het landhoofd	! 18 - Geotechniek	te verifiëren
Heeft leidend en rakend object				
Leidend object	E0130 - Landhoofd			
Rakend object	E0140 - Keermuur toegangshelling			
is raakvlak bij activiteit				
Activiteit	Status	Rol	Naam	
Geometrisch ontwerp brug autoverkeer	te beginnen	Senior Wegontwerper	Gunther	
Ontwerplannen keermuur	te beginnen	7. Directie Geotechniek	Evi	
Stabiliteit keermuur	te beginnen	7. Directie Geotechniek	Evi	
Uitvoeringsontwerp brug autoverkeer	te beginnen	Ontwerpleider Kunstwerken/bruggen	Arne	
W0210a - Opmaken bekistings- en wapeningsplannen	te beginnen	Ontwerper uitvoeringstekeningen	Pieter-Jan	
Verantwoordelijke				
Rol	Persoon			
1.5 - Raakvlakmanager	Thibaud			
Acties				
Actie	Status	Rol	Uitvoerder	
Nagaan of plannen keermuur en landhoofd juist zijn afgestemd	Nog te beginnen	Ontwerper Geotechnische constructies	Nathalie	

Figuur 70: Detailscherm raakvlak "Landhoofd - Keermuur toegangshelling"

Aan het raakvlak kunnen ook acties gekoppeld worden die nodig zijn voor het beheer van het raakvlak, in dit geval bijvoorbeeld het nagaan of de ontwerplannen van de grondkerende wand en het landhoofd juist op elkaar zijn afgestemd. Deze acties worden aangemaakt door de verantwoordelijke raakvlakmanager, die tevens aanduidt wie de actie moet uitvoeren.

Zoals in [hoofdstuk 4 - Raakvlakkenbeheer](#) al uitvoerig behandeld werd, is aan het raakvlak exact één eis gekoppeld. Bij de eis wordt ook nog

weergegeven tot welke categorie deze behoort, en of de verificatie al is uitgevoerd.

5.6 Besluit

De cases die in dit hoofdstuk voorgesteld werden, maken duidelijk dat een 3D-aanpak een aanzienlijke meerwaarde kan hebben voor een project. Zo zou het realiseren van de trap zeer moeilijk zijn zonder een 3D-model.

Een ander voordeel van deze modellen is dat het mogelijk wordt een raakvlakkencontrole van de ontwerpplannen uit te voeren. Dit is duidelijk naar voor gekomen bij de case in verband met de rioleringsdoorsteek.

Verder is, met het relatief eenvoudige voorbeeld van de randbalk, aangehaald hoe complexere raakvlakken op een systematische manier geanalyseerd kunnen worden.

Ten slotte illustreert de laatste case, op basis van het raakvlak tussen de keermuren en het landhoofd, dat de databankomgeving een nuttige rol speelt om alle projectgegevens overzichtelijk bij elkaar te brengen en te managen.

In [hoofdstuk 6 – Besluit](#) wordt een volledig overzicht gegeven van de belangrijkste conclusies uit deze scriptie. De principes die praktisch toegepast zijn in het bovenstaande hoofdstuk, zijn ook terug te vinden in deze conclusies.

HOOFDSTUK

6

Besluit

Deze masterproef is een studie naar een optimalisering van het huidige raakvlakkenbeheer. In dit hoofdstuk volgt een overzicht van de belangrijkste conclusies die uit het onderzoek naar voor komen. Aansluitend wordt aangegeven welke stappen in de toekomst nog nodig zijn om het (geometrisch) raakvlakkenbeheer verder te optimaliseren.

6.1 **Conclusies**

Bij aanvang van deze masterproef is de volgende drieledige doelstelling geformuleerd:

Optimalisering van het raakvlakkenbeheer door middel van:

- 1) het uitvoeren van een marktonderzoek, waarbij de raakvlakkenproblematiek in kaart wordt gebracht;
- 2) het zoeken naar een systematiek om raakvlakken vast te leggen en te beheren;
- 3) het onderzoeken van de verschillende softwaremogelijkheden om geometrische raakvlakcontroles mogelijk te maken.

Hieronder wordt weergegeven hoe het onderzoek deze doelstelling tracht in te vullen, en welke conclusies hieruit getrokken zijn.

6.1.1 Verkennend marktonderzoek

Uit het marktonderzoek is gebleken dat er nog veel onduidelijkheid bestaat over wanneer een interactie tussen verschillende componenten als 'raakvlak' bestempeld kan worden. Voor alle raakvlakken die als zodanig beschouwd worden, is het aanbevolen deze altijd expliciet vast te leggen.

De resultaten uit het marktonderzoek geven verder aan dat het beheer van raakvlakken vooral werkbaar moet blijven. Het is niet nodig eindeloos raakvlakken en eisen te definiëren, maar veeleer op een duidelijke manier te communiceren met elkaar. Daarnaast wordt er gestreefd naar zoveel mogelijk transparantie. Dit betekent dat steeds wordt bijgehouden welke eisen, beslissingen enz. in de loop van het proces vastgelegd worden, en om welke redenen.

In de praktijk blijkt dat het beheer van raakvlakken vaak eerder impliciet gebeurt, wat niet zelden aanleiding geeft tot het ontstaan van problemen in verband met raakvlakken. Dit maakt de behoefte aan een eenduidige beheersystematiek bijzonder groot.

6.1.2 Beheersystematiek

Door een 3D-benadering te combineren met een (relationele) databank, is het niet enkel mogelijk op geometrische wijze raakvlakken op te sporen, maar ook het raakvlakmanagement te kaderen binnen een geïntegreerd projectbeheer.

Een eerste stap voor een succesvol beheer van raakvlakken, bestaat erin de interacties in een zo vroeg mogelijke projectfase op te sporen. Het is niet alleen belangrijk de raakvlakken tijdig te detecteren, maar ze ook op te volgen in het volledige verloop van het project. Dit kan door de raakvlakken in een databankomgeving op te slaan en er acties aan te koppelen.

Om fouten en laattijdig doorgevoerde wijzigingen in de plannen alsnog op te merken, is het belangrijk om op geregelde tijdstippen een verificatiestap in te bouwen. Vermits het gaat om geometrische raakvlakken, is het

mogelijk bij de verificatie van de ontwerpplannen ondersteunende software te gebruiken.

6.1.3 Softwareondersteuning voor raakvlakkenbeheer

Door een 3D-ontwerp wordt het mogelijk raakvlakcontroles gedeeltelijk te automatiseren. Hoewel de clash detection software geavanceerde mogelijkheden biedt, hebben praktische testen aangetoond dat een kritische ingesteldheid ten opzichte van de verkregen resultaten noodzakelijk blijft. Niet alle gedetecteerde clashes zijn in werkelijkheid betekenisvol, en er zijn geen garanties dat steeds alle raakvlakproblemen gesignaleerd worden.

3D-geometrie en ondersteunende software vormen een grote hulp bij het beheren van raakvlakken, maar het is toch belangrijk in te zien dat zij op zichzelf geen oplossing bieden voor de problematiek. Het gebruik van software moet immers steeds kaderen in een overkoepelend beheer.

6.2 Verder onderzoek

In deze masterproef is meermaals gewezen op welke vlakken in de toekomst nog ruimte is voor verbetering. Zo is op het gebied van softwareondersteuning voor de projectbeheersing nog heel wat vooruitgang nodig. Onderzoek heeft uitgewezen dat de onderlinge compatibiliteit van (clash detection) software vaak te wensen overlaat, met informatieverlies en tegenstrijdige resultaten tot gevolg.

Daarnaast legt ook het gebrek aan interactie tussen grafische software en een tekstuele databank beperkingen op aan een vlotte communicatie tussen de betrokkenen.

Hoewel in de toekomst op verschillende vlakken verbetering mogelijk is, kunnen vele problemen omtrent raakvlakken al vermeden worden door open communicatie en transparante beheerprocessen binnen het project. Zowel de problemen die in deze scriptie aan het licht kwamen, als de voorgestelde oplossingen vormen een bijdrage in de verdere evolutie naar een gestructureerd en transparant raakvlakkenbeheer.

BIBLIOGRAFIE

- [1] Delporte T et al. (2009) Outputspecificaties. Een leidraad voor opdrachtgevers en opdrachtnemers van PPS-contracten, Vlaams Kenniscentrum PPS, Brussel
- [2] Van Garsse S et al. (2009) DBFM Handboek, Vlaams Kenniscentrum PPS, Brussel
- [3] De Boer G et al. (2009) Leidraad voor Systems Engineering binnen de GWW-sector, Rijkswaterstaat, Prorail, Bouwend Nederland, Vereniging van Waterbouwers, NLIingenieurs, Den Haag
- [4] Jacobs MMJ et al. (2007) Handboek Oplossingsvrij specificeren, CROW, Ede
- [5] Haskins S et al. (2010) Systems Engineering Handbook. A guide for system life cycle processes and activities, International Council on Systems Engineering (INCOSE), San Diego
- [6] Zwakhals T et al. (2008) SE wijzer. Handleiding Systems Engineering voor BAM Infra, Koninklijke BAM Groep, Gouda
- [7] Schaap HA, Bouwman JW (2006) Toekomst voor het bouwproces. Een 3D-objectbenadering, Programma COINS, CUR, Gouda
- [8] AutoDesk (2010) AutoCad Civil 3D 2011 Tutorials - Autodesk Vault [Computer software en handleiding]. Beschikbaar op <http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=14246800>. Geconsulteerd op 08.12.2010
- [9] PKM Solutions (2009) Toelichting Template Systems Engineering 02. PKM Solutions, Ridderkerk
- [10] PKM Solutions (2010) Relatics – Informatiebeheer van de toekomst!. Beschikbaar op <http://www.relatics.com/>. Geconsulteerd op 05.11.2010
- [11] BIW Technologies Limited (2010) BIW Technologies – Applications for projects and programmes. Beschikbaar op <http://www.biwtech.com/>. Geconsulteerd op 15.11.2010
- [12] Wilkinson P. (2005) Construction Collaboration Technologies: The Extranet Evolution BIW. Taylor & Francis, New York
- [13] ISO (2010) International Organization for Standardisation – ISO/IEC 15288:2008. Beschikbaar op <http://www.iso.org/iso/>. Geconsulteerd op 15.01.2011

- [14] Team BouwICT - TNO Bouw en Ondergrond (2010), BIM Woordenboek
- [15] Solibri inc. (2010) Solibri – The World Leader in Model Checking. Beschikbaar op <http://www.solibri.com/>. Geconsulteerd op 17.10.2010
- [16] L. Khemlani, AECbytes (2009) Solibri Model Checker. Beschikbaar op <http://www.aecbytes.com/review/2009/SolibriModelChecker.html>. Geconsulteerd op 01.12.2010
- [17] Solibri Inc. (2010) Solibri Releases Solibri Model Checker v6 and Introduces Information Takeoff (ITO). Beschikbaar op <http://www.solibri.com/press-releases/>. Geconsulteerd op 08.12.2010
- [18] Longitude Media, LLC (2010) Cadalyst. Beschikbaar op <http://www.cadalyst.com/>. Geconsulteerd op 17.12.2010
- [19] Bulens W (2008) 4D Visualisatie, Hogeschool voor Wetenschap & Kunst – campus De Nayer, Sint-Katelijne-Waver
- [20] Autodesk (2010) Autodesk Navisworks Manage 2011 User Guide [Computer software en handleiding]. Beschikbaar op <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=11621516>. Geconsulteerd op 13.04.2011
- [21] Bentley Systems Inc. (2010) ProjectWise Navigator. Beschikbaar op <http://www.bentley.com/>. Geconsulteerd op 23.12.2010
- [22] Bentley Systems Inc., ProjectWise Navigator V8i (SELECTseries 2) Help [Computer software en handleiding]. Beschikbaar op <http://docs.bentley.com/docinfo.php?doc=745>. Geconsulteerd op 22.04.2011
- [23] Esri Nederland (2010) Esri Nederland – Geographic Thinking. Beschikbaar op <http://www.esri.nl/>. Geconsulteerd op 26.11.2010
- [24] Kalaitzidis C et al. (2010) SEOS – Remote Sensing and GIS in Agriculture. *Supplement - Geographic Information Systems (GIS)*. Beschikbaar op <http://www.seos-project.eu/>. Geconsulteerd op 18-11-2010
- [25] National Aeronautics and Space Administration – NASA (2007) Systems Engineering Handbook, NASA Headquarters, Washington
- [26] Grady Jeffrey O (2006) System Requirements Analysis, Elsevier Inc., London
- [27] National Aeronautics and Space Administration – NASA (2006) NAS System Engineering Manual, version 3.1, NASA HQ, Washington

- [28] Google Inc. (2011) Google Maps. Beschikbaar op www.maps.google.be. Geraadpleegd op 12.05.2011
- [29] Beheersmaatschappij Antwerpen Mobiel (2008) Noorderlaanbrug. Beschikbaar op <http://www.antwerpen.be/projecten/noorderlaanbrug.aspx>. Geconsulteerd op 12.05.2011

