SACJ 30(1) July 2018
Research Article

An integrative modelling technique bridging the gap between business and information systems development

P. Joubert[a], C. de Villiers[a], J.H. Kroeze[b]

[a] Department of Informatics, University of Pretoria, South Africa
[b] School of Computing, University of South Africa, South Africa

ABSTRACT

The goal of this research is to develop an integrative modelling technique that is easy enough to be used by most business users with little training, but robust and structured enough to be used in subsequent Information Systems Development (ISD) modelling. This technique attempts to bridge the current gap between modelling on a business level and modelling on a technical level. The overall research methodology is design science research, embedding a grounded approach, to develop an integrative modelling technique. The resultant artefact is applied to a case study to test its applicability and suitability, and the results are evaluated. The results show that the proposed integrative modelling technique that is based on a better understanding of the fundamental entities in business and ISD modelling and their properties, attributes and relationships, can be used as a method to model business situations easily and expressively. By overcoming the divide between business and ISD modelling, the technique also advances informal, mostly textual, business modelling. The paper makes a methodological contribution by establishing a new technique that integrates business analysis with ISD, as well as demonstrating how a single case study could serve as an exemplar of a theory.

Keywords: Information systems development, business modelling, modelling techniques, grounded approach, design science research

Categories: • **Applied computing ~ Enterprise computing** • *Business process management ~ Business process modelling*

Email:
C. de Villiers carina.devilliers@up.ac.za,
J.H. Kroeze kroezjh@unisa.ac.za (CORRESPONDING)

Article history:
Received: 2 September 2016
Accepted: 12 March 2018
Available online: 10 July 2018

1 INTRODUCTION

While a variety of modelling languages and techniques exist for Information Systems Development (ISD), few of them can easily and simply be applied to business modelling (Wand, 1996; Wyssusek, 2006; ZurMuehlen & Indulska, 2010; Gould, 2016). Wilcox and Gurău (2003) identify a number of problems with using IDEF (Integration DEFinition) for business modelling, including rigidity, lack

Joubert, P., de Villiers, C. and Kroeze, J.H. (2018). An integrative modelling technique bridging the gap between business and information systems development. *South African Computer Journal 30*(1), 40–65. https://doi.org/10.18489/sacj.v30i1.413

SACJ is a publication of the South African Institute of Computer Scientists and Information Technologists. ISSN 1015-7999 (print) ISSN 2313-7835 (online).

of standardisation and low portability. They propose UML (Unified Modelling Language) instead, but with the provision that it should be amended to cover business aims, rules and procedures satisfactorily, which suggests that these modelling problems are mostly related to complexity and expressiveness. Most business users find it too difficult to express their business needs in a complex ISD language or technique, while ISD analysts are often not able to represent everything they require using business models (Recker, 2010). In practice, most business modelling is done in unstructured text format, which leads to unclear, ambiguous descriptions (ZurMuehlen & Indulska, 2010; Gould, 2016; Sinha, Paradkar, Kumanan, & Boguraev, 2009).

The problem statement is:

> Current modelling techniques do not bridge the gap between business modelling and ISD.

The main research question is:

> How can an integrative modelling technique be developed to bridge the gap between business modelling and ISD?

Business and ISD modelling is fundamentally concerned with design and, therefore, design science research (Purao, Baldwin, Hevner, Storey, & Pries-Heje, 2008) is used in this research as the overall research methodology while embedding a grounded approach to discover and clarify the fundamental constructs of business and ISD modelling. The paper makes a methodological contribution by proposing a technique to integrate business analysis with ISD, as well as demonstrating how a single case study could serve as an exemplar of a theory (Flyvbjerg, 2006). Case studies can be used to generate or test hypotheses during theory-building—according to Flyvbjerg (2006, p. 219), "[s]ocial science may be strengthened by the execution of a greater number of good case studies".

The paper is structured as follows. Section 2 gives background information about current ISD modelling techniques and presents a critical literature review. Section 3 discusses the research approach in more detail. In Section 4, the insights gained from the discovery phase (filtered down using a grounded approach) are synthesised into the fundamental modelling entities that will be used in the proposed artefact—i.e., a new technique (the proposed integrative modelling technique)—to illustrate how business and ISD modelling could be integrated. The steps of this technique, as well, as the proof of concept, can be found in Section 5 where the procedural aspects are spelled out and applied to a business case demonstrating its validity. Section 6 is the discussion section that evaluates the preceding research and the proposed technique. Finally, the conclusion (Section 7) summarises the most salient points of the article and its findings and makes some recommendations for future work. In order to make the article more accessible a number of tables and figures were moved to a supplementary material appendix[1] and these will be referred to as "Table SM" or "Figure SM" with consecutive numbering.

[1] http://sacj.cs.uct.ac.za/index.php/sacj/article/downloadSuppFile/413/216

2 BACKGROUND

Conceptual ISD modelling is done by using various techniques, including object-oriented techniques like UML diagrams (Schach, 2004); holistic techniques like soft system methodology (SSM) and rich pictures (Checkland, 2000); data techniques like entity-relationship diagrams (ERDs) (Chen, 1976; Umanath & Scamell, 2014; Prescott, 2015); process techniques like IDEF0 (National Institute of Standards and Technology, 1993) and business process modelling notation (BPMN) (Object Management Group, 2009). These techniques have many commonalities between them, but also many differences. Some of the differences are more obvious, for example specific concepts may exist in one technique but not in another. Other differences are more subtle, such as different naming conventions, or they may be represented differently in different techniques.

New ISD modelling techniques were developed over time, pragmatically, to solve specific problem issues. For example, ERD and normalisation were developed to formalise the modelling of data and databases to overcome database problems (such as creating, updating and deleting anomalies) (Chen, 1976), while BPMN was developed to model processes in an organisation and to overcome the lack of standardisation in current process modelling techniques (Recker, 2010).

One of the main problems experienced by practitioners and students of business and ISD modelling is how to integrate all of the various ISD modelling techniques (Wyssusek, 2006; Wand & Wang, 1996; Salam & Yusof, 2015; Hassan & Mathiassen, 2017). When using more than one of these techniques in the same context, for example, how do the corresponding concepts of *agent* in agent-object relationship (AOR), *external agent* in data flow diagrams (DFD) and *actor* in use case diagrams relate to each other? And even more confusing, how do the corresponding concepts of *use case* and *system* in use case modelling, *behaviour* in class diagrams, *process* in DFD, IDEF0 and BPMN, *transformation* in SSM, *activity* in activity diagrams; *how* in Zachman, and many other action-oriented concepts relate?

The integrative modelling technique that is proposed, implemented and evaluated in this research is the result of an in-depth qualitative data analysis of existing business and ISD techniques. The main techniques of the past, the main techniques that are currently popular, as well as lesser-known techniques, were included to give as broad a range of techniques as possible. The ISD approaches and techniques studied were adapted from Avison and Fitzgerald (2006), who describe two types of techniques: representational techniques describe ways of modelling some universe of discourse, while process techniques describe a set of actions to achieve a certain goal, e.g., the joint requirements planning (JRP) technique describes how to get requirements but not how to represent or model them. Only representational techniques were considered here. The basic principle employed in the selection of techniques was to use generally accepted, internationally recognised standards, like the IDEF0 standard. If there was no such standard, a source on the relevant techniques was used, e.g., Chen (1976) for data modelling. A summary of modelling techniques is shown in Table SM1. Tables SM2 and SM3 give an overview of these techniques and an illustration of the grounded process followed to synthesise concepts from these into the fundamental modelling entities of the integrative modelling technique proposed in Section 4.

3 RESEARCH APPROACH

The main research approach followed is design science research. This approach provides another view that complements the positivist and interpretivist perspectives of IS research (Niehaves & Bernd, 2006). It distinguishes between natural science and the science of the artificial, and concentrates on phenomena that are created (designed artefacts), rather than objects occurring naturally (Levy & Hirschheim, 2012). Designed artefacts can be, among other things, algorithms, human-computer interaction (HCI) constructs, ISD methodologies and ISD techniques.

Vaishnavi and Kuechler (2015), Vaishnavi, Kuechler, and Petter (2017) developed a taxonomy of design science research output as summarised in Table 1.

Table 1: The outputs of design science research (Vaishnavi & Kuechler, 2015; Vaishnavi, Kuechler, & Petter, 2017)

Output	**Description**
Constructs	The conceptual vocabulary of a domain.
Models	A set of propositions or statements expressing relationships between constructs.
Instantiations	A set of steps used to perform a task: ‘how-to’ knowledge.
Better theories	Artefact construction as analogous to experimental natural science.

The artefact suggested is an integrative modelling technique between business and ISD. In terms of design science research outputs as defined in Vaishnavi and Kuechler (2015) and Vaishnavi et al. (2017) (see Table 1), this modelling technique can be seen to incorporate the first three outputs: constructs, a model and (to some extent) a method.

Design science research involves the following steps:

1. problem identification and motivation;
2. defining the objectives of a solution;
3. design and development;
4. demonstration;
5. evaluation; and
6. communication.

These six activities (Geerts, 2011) were approached as follows in this research:

1. Identify **problem** and **motivate**: There is a non-trivial gap between business modelling and ISD modelling. ISD modelling is precise and has many widely used modelling techniques to enable it. Business modelling is mostly done using textual descriptions without a specific technique employed. (See the introduction and background sections above.)

2. **Define** the **objectives of a solution**: The solution to this problem is to define an integrative modelling technique that will bridge the gap between business and ISD modelling. The objectives of this technique are specifically as follows:

   - **Ease of use**: This technique must be simple enough so that non-technical users should be able to use it with minimum training.
   - **Expressiveness**: At the same time, the technique must be expressive enough so that detailed ISD level modelling can be derived from it without further interactions with users or analysts. This can be stated in a different way. Luukkonen, Korpela, and Mykkänen (2010) developed a classification of modelling techniques used in the earlier phases of the ISD life cycle. They identified two dimensions: degree of structure related to the syntactic aspect of a model (ranging from unstructured to highly structured), and scope related to the semantic and pragmatic aspects of the model (ranging from technical to human and organisational). In terms of this classification, the purpose of the integrative modelling is to be highly structured and also to cover the full range of the scope dimension.

3. **Design and develop**: Such an integrative modelling technique is developed using a grounded approach and qualitatively analysing existing modelling techniques, and by synthesising the salient aspects into a new technique (see Sections 2 and 4 and the methodological parts of Section 5).

4. **Demonstrate**: This technique is demonstrated by looking at an example of very common business modelling situations where the output will be used by ISD modelling techniques. A case study is used to illustrate the application of the proposed modelling technique. The case study will incorporate organisational, business and IT aspects and also the different phases of the systems development life cycle (see the application parts of Section 5).

5. **Evaluate**: This technique is evaluated theoretically in Section 6 using principles from grounded theory and components of IS design theory while considering the objectives of the research identified in step 2 above.

6. **Communicate**: The purpose of this article is to communicate the results of the study to an academic audience. Therefore, the focus is on the theoretical (methodological) contribution of the research.

4 INTEGRATIVE MODELLING TECHNIQUE

Since modelling is closely linked to linguistics, a linguistic-type framework is the best way to structure the integrative technique in a consistent, coherent and integrated way (Joubert, Kroeze, & Villiers, 2013). The fundamental **modelling entities** of business and ISD can be divided into three categories:

- **Base entities**: corresponding to the morphological level in linguistics
- **Structure entities**: corresponding to the syntactic level in linguistics
- **Role entities**: corresponding to the semantic level in linguistics

The **base entities** are the most basic building blocks in the modelling process. They represent the real-world objects that make up organisations and systems. In the same way that verbs, nouns, adjectives, adverbs, pronouns and others make up the words of natural languages, these entities form the **words** of the proposed modelling language.

The **structure entities** form, like their natural language counterparts, the model sentences and phrases with which systems, organisations and situations can be described. They use the base entities plus specific language constructions to form these model sentences and phrases. A number of model sentences together form a model. In turn, specific subsets of the model form model views.

The **role entities** provide insight into the meanings of the base and structure entities. For example, an actor can either play an agent role or patient role in a model phrase.
Figure 1 gives a high-level overview of these modelling entities.

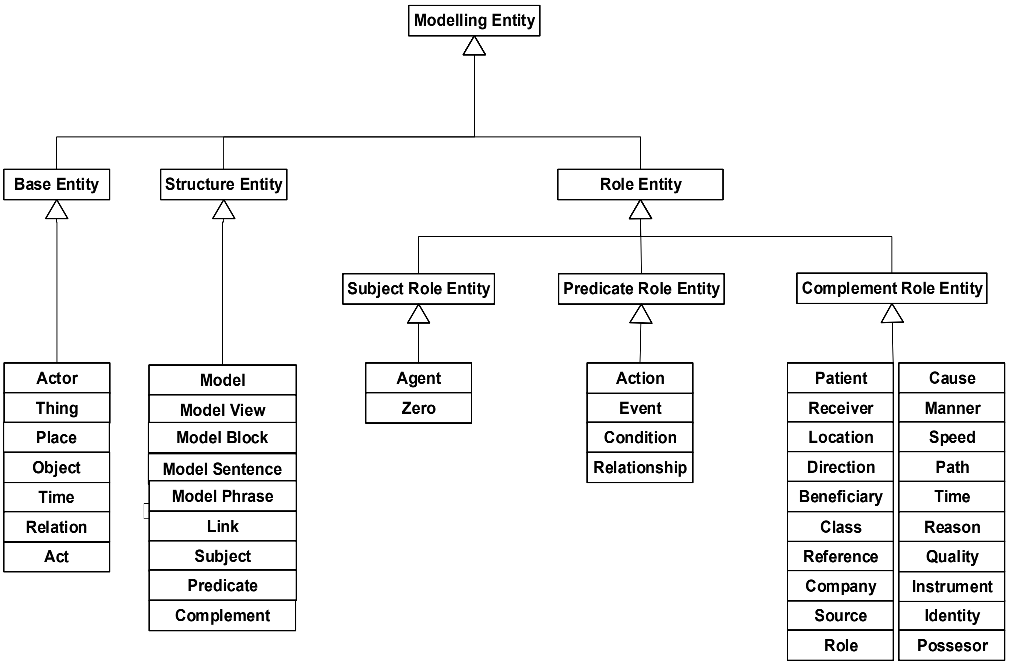

Figure 1: A high-level overview of the modelling entities

A note on the choice of modelling entity names is important at this stage. In the history of ISD modelling, various names have been used for the same modelling entities. It makes it very difficult to decide on a name that has so many synonyms. The approach followed in this research is to use names that are either the most popular names (used by most techniques), or to use names that are more general rather than more specific (for instance, rather thing than entity or object), or to use novel names when many different, conflicting names are used. Base, structure and role entities are defined below, while the underlying morphological, syntactic, semantic and pragmatic linguistic aspects of these constructs are discussed in more detail in Joubert et al. (2013).

4.1 Base entities

To model any information system or organisation, surprisingly few entities are needed. Corresponding to the morphological level of linguistics, these base entities are as follows:

- **Things**: intelligent entities (actors) and non-intelligent entities (objects) corresponding to nouns and noun phrases in linguistics.
- **Actors** (intelligent things): human, institutional and artificial entities that can act and make decisions. Actors correspond to animate nouns in linguistics.
  - **Human actors** (or persons) are individual humans like clients, employees and users.
  - **Institutional actors** (or human activity systems) are groups of people where the group has an own identity and is involved in purposeful activities, for example, companies, departments and project teams. Institutional agents are made up of other base entities.
  - **Artificial actors** are human-engineered artefacts that simulate human behaviour, such as bank account systems, pocket calculators and robotic manufacturing systems. Artificial actors consist of other base entities.
- **Objects** (non-intelligent things): physical, conceptual or informational (semiotic) entities employed during actions by agents. Objects can consist of other objects. Objects correspond to inanimate nouns in linguistics.
  - **Physical objects** are material things that can be detected by the senses, such as raw material, products, furniture and vehicles.
  - **Conceptual objects** (or **information**) are concepts that are created by humans to make their world more understandable and to communicate with other actors. Three important specific types of conceptual objects are listed below:
    - **Places** (or **locative conceptual objects**) are physical two- or three-dimensional areas or conceptually demarcated areas. Examples include countries, geographical areas, residential plots, offices in a building and areas on a screen, form or report.
    - **Times** (or **time-related conceptual objects**) are indications of absolute (e.g., 12/01/2011) or relative (e.g., two days after month-end) times.
    - **Model blocks** are structure modelling entities (described below) that can be handled as if they were base entities. For instance, the process *withdraw money from ATM* (a model block) can be seen as a single conceptual object and used as such, although it consists of a number of base entities and their relations and acts.

  - **Informational objects** are combinations of information and physical objects (acting as the media) created with the specific purpose of capturing, storing and displaying information, like files, databases, books, spreadsheets, input forms and whiteboards. Informational objects are related to conceptual objects in that they both have information as basis, but the physical medium of a conceptual object is not of importance.

- **Acts and relations** are dynamic relationships (acts) and static relationships (relations) between things corresponding to verb and verb phrases in linguistics. Relations are common to all situations and comprises a (theoretically) finite specific set of auxiliary verbs and verb phrases, such as *is-a-type-of*, *consists-of*, *has*, *is-above*, and *is-equal-to*. Acts, on the other hand, are specific to every situation and consists of all active verbs that can be used in a situation under discussion, for example *withdraw* (money), *print* (receipt), *deposit* (money), *transfer* (money) and *pay* (beneficiaries) in the ATM situation. Acts and relations correspond to verbs in linguistics. Figure 2 provides a graphical overview of these modelling base entities.

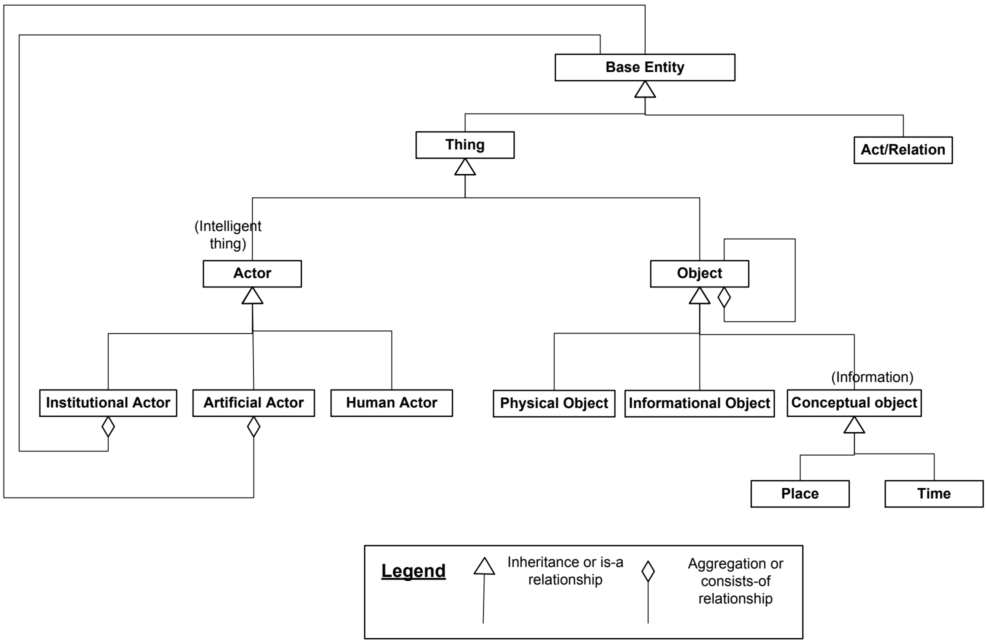

Figure 2: Modelling base entities

4.2 Structure entities

In natural language, words can be structured together to form sentences. In a similar way, base entities can be structured together with specific language constructs to form **structure entities** (see Figure 3). Structure entities correspond to the syntactic level in linguistics.

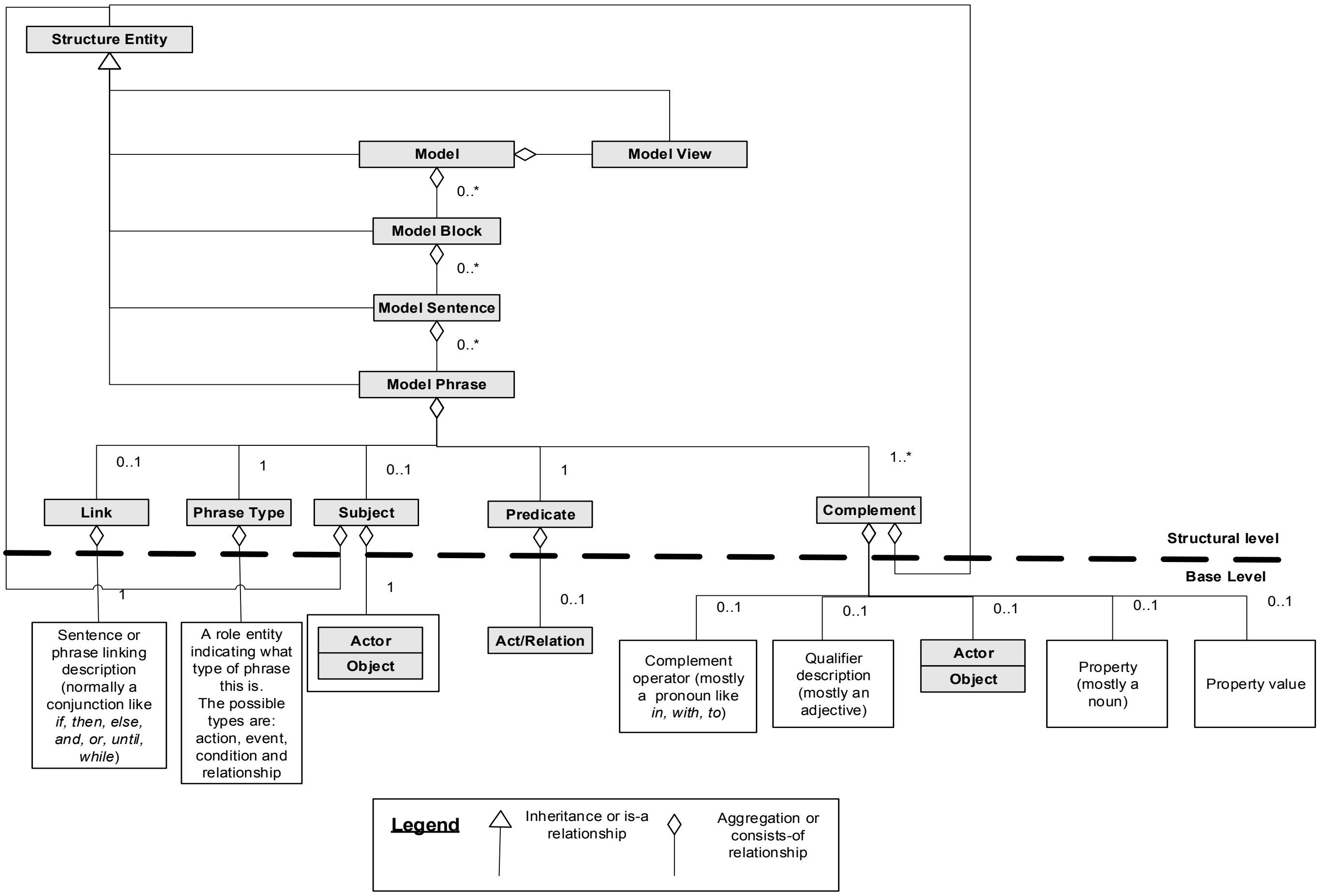
Structure Entity
Model
Model View
0..*
Model Block
0..*
Model Sentence
0..*
Model Phrase
0..1
1
0..1
1
1..*
Link
Phrase Type
Subject
Predicate
Complement
Structural level
Base Level
1
1
0..1
0..1
0..1
0..1
0..1
0..1
Sentence or phrase linking description (normally a conjunction like *if, then, else, and, or, until, while*)
A role entity indicating what type of phrase this is. The possible types are: action, event, condition and relationship
Actor
Object
Act/Relation
Complement operator (mostly a pronoun like *in, with, to*)
Qualifier description (mostly an adjective)
Actor
Object
Property (mostly a noun)
Property value
Legend
Inheritance or is-a relationship
Aggregation or consists-of relationship

Figure 3: Modelling structure entities

4.2.1 Models and model views

A distinction must be made between the "real" and the "representational" (or modelled) world. Business and ISD modelling involves taking a part of real-world organisation, representing it using some technique like UML or ERD to represent the real system **as-is**, and then designing a new real system *to-be*. This representation of the new system to-be is then developed and implemented to become part of (or change) the real system. If a new system must be developed on the same part of the real-world organisation, the created information system is now part of it.

Although there is one real world, there can be many views (representations) of it. A view will always only represent a part of the real world and will also represent only a specific aspect of it.

For instance, in an ERD, all other aspects are ignored (although they are as important) and the focus is only on the data aspect of the system.

The **model** represents all information available on the real-world situation, organisation or system modelled. By implication, the model is only a representation of the real world. From this one model, various **model views** can be produced, representing any specific aspect that the model viewer wants to emphasise.

Views can be based on a number of aspects. The first aspect is the type of base entity that needs to be considered. It can be seen as a horizontal view. For instance, data modelling concentrates on informational objects, an organogram on human and institutional actors, a work breakdown structure on actions and a bill of material on physical objects.

The second aspect is the level of detail that needs to be expressed. It can be seen as a vertical view. For instance, using use case modelling, sometimes only the systems and subsystems involved are considered (model block level), other times only case names are used (action level) and at other times, the focus is on the use case detail such as steps, actors, conditions and business rules (action step level).

The third aspect is stakeholder perspectives. It can combine characteristics of the first two aspects. The rows ‘Planner’ (scope, contextual), ‘Owner’ (enterprise), ‘Designer’ (system, conceptual, logical), ‘Builder’ (physical, technical) and ‘Subcontractor’ (component) of the Zachman framework illustrate the typical stakeholder perspectives well. Over time (during the life cycle of the system), different perspectives will be more important than other. There is also the design issue of postponing certain decisions until a later phase, for instance, to concentrate during conceptual design on how the system is to operate but without taking the specific technology or implementation into account.

A fourth aspect is the history view. It can be combined with any or even all of the previous views. An *as-is* view shows how things are currently while a *to-be* view shows how things are designed or planned to be in the future. In practice there is not just one *as-is* and one *to-be* view. Different versions of both these views can exist.

A fifth aspect is that of linguistics. In this view, the entities and their relationships are seen as the elements of a language. As in any language, the various linguistic levels of morphology, syntax, semantics and pragmatics are considered.

Model phrases and model sentences can be grouped into model blocks. They are named for reference and usage. Once a model block has been created, it becomes a conceptual object and can be used as such.

A model block has the following three properties, over and above the constituent model phrases and sentences:

- A **name** by which it can be referenced and used either as a conceptual object or as an executable set of instructions.
- Zero or more **input parameters** that can be used to let the model function like a subroutine or function when it is used as an executable.
- Zero or more **output parameters** generated from the execution of the model block.

Model phrases are joined together to form model sentences. Model phrases and sentences are either events, conditions, actions or relationships. The most fundamental structure entity is the model phrase. In its simplest format, it consists of the basic parts of the simplest sentence in natural language, namely subject-predicate-object. They can be read from left to right as a complete humanly understandable sentence. Conversely, these model phrases can be created from natural language statements, but, importantly, the structured approach in creating model phrases make them also open to manual or system-assisted analysis and manipulation. For instance, it is possible to algorithmically create a partial-use case diagram for the first phrase and a partial non-attributed ERD for the second. It is also possible to lay down specific rules for these phrases. For instance, the subject of an act can only be an actor, while the subject of a relation can be an actor or an object.

This model phrase structure can be extended to also indicate other objects involved in an action or relationship, over and above the direct object. The **object** construct is therefore renamed **complement** and a **complement operator** is inserted before it, indicating the relationship of the complement to the act or relation. The Complement Operator mostly consists of prepositions, but can occasionally include other word types.

The model phrase can be further extended by adding a **complement qualification**, which qualifies the relation between the predicate and the complement. It can be as user-friendly (e.g., only one, one-to-many, not more than 50, optional one) or as precise (e.g., 1, $1 \ldots n$, ≤ 50, $0 \ldots 1$) as needed for the analysis situation.

Model phrases are joined together to form **model sentences**. A **link** is added and used to link model phrases together into sentences.

The structure entities making up a model phrase are as follows:

1. The **link** identifies the relationships between actions. The most common links between actions are sequence (actions following one another), repetition (actions forming a loop), decision (actions dependent on some condition) and concurrency (actions executed simultaneously).

2. The **subject** of the relationship. It refers to the base entity (actor or object) which is the main party in the model phrase.

3. The **predicate** of the model phrase. It describes the action taking place or the type of relationship between the subject and the object. The predicate can play any of the following role types:

   - **Events**—Describe external triggers that cause agents to initiate actions or cause the state of things to change. For example, an important event type is the **timer event**, where either absolute time, like “on 2 August 2007”, or relative time, like “at month-end” and “at the end of the day”, cause actions to take place.

   - **Conditions**—Describe conditions, as part of a bigger model sentence, affecting later actions’ steps, mostly action phrases. For example, “if the reorder level goes below 20”.

  - **Actions**—Describe how agents (human and non-human) perform work needed to reach the objectives of one or more individual, institutional or artificial agents. For example, “client places order”.
  - **Relationships**—Describe static relationships between subject and complement.

4. The optional **complement operator** of the model phrase. It identifies the relationships of the various complements (direct object, indirect object, other complements) involved in the model phrase. Complement operators are mostly prepositions such as “for”, “in”, “from” and “to”.

5. The optional **complement qualifier** of the model phrase. It identifies qualifications that must be placed on the complement. The most common object operators are multiplicity indications like “one or more” and “one only”.

6. The **complement** of the action refers to the base entities (actor or object) that are affected by the predicate of the model phrase. It can also refer to the property of a base entity.

4.3 Role entities

Every type of base and structure entity has a specific meaning depending on its use in a model sentence or phrase. These meanings are specified by the role entities. Role entities can be divided into four categories:

- **Subject role entities** identify the meanings that the subjects of acts or relations can have, namely **agent** or **zero**:

  – If the subject assumes the **agent** role, it is actively involved in performing some action on an object, optionally involving other objects.
  – If the subject assumes the **zero** role, it indicates a specific state for the subject.

- **Predicate role entities** identify the meanings that the predicate of acts or relations can have (note that the predicate on its own does not always fully define its role). In many cases, other parts of the modelling phrase, especially the link, are needed to define the predicate’s role. In that sense the role of the predicate can be seen as the role of the whole modelling phrase.

  – **Actions** indicate dynamic relations (acts) between entities. Actions can be one of the following:

    * **Transformation** or processing of one entity into another, either by assembly or by pure transformation
    * **Transportation** or distribution of an entity from one place to another, including **arranging** objects in new patterns
    * **Storing** or housing an entity in a place

        - **Exchanging** or trading an entity for another entity of value
        - **Control** or regulation of an entity
    - **Relationships** indicate static relations between entities. Relationships are mainly one of the following:
        - **Association**, where one entity is associated with another entity. For example, a client (human actor) can have contracts (informational objects). This is mostly indicated by the auxiliary verb *has*.
        - **Property**, where one entity owns another entity, for example, name is a property of client.
        - **Instance**, where one entity is an instance of another entity. For example, student John with student number 123456 is an instance of type student, mostly indicated by the verb phrase *is-a*.
        - **Recursion**, where an entity is associated with itself. For example, a course is related to other courses as prerequisite courses, mostly indicated by the verb *has*.
        - **Aggregation** and **composition**, where an entity consists of other entities. Composition is a "stronger" relationship than aggregation and must also comply with the following criteria: only one entity in the relationship represents the whole; the parts in the relationship exist only as long as the whole exists, and the whole is responsible for creating and destroying its parts; and a part may only belong to one whole at a time, but it can be attached to another whole (following UML). Aggregation is mostly indicated by the verb phrase *consists-of*.
        - **Inheritance**, where one entity is a type of another entity. For example, student (human actor) and staff (human actors) can both be seen as persons (human actors). Mostly indicated by the auxiliary verb phrase *is-a-type-of*.
        - **Location**, where one entity is located in a specific relationship to another location, e.g., "above", "in", "below".
    - **Events** indicate acts that trigger other acts.
    - **Conditions** indicate relations between entities that either allow or disallow actions from taking place.

- **Complement role entities** identify the meanings that the entities described in the complement of acts or relations can have. These complement roles are borrowed and adapted from Functional Grammar (Dik, 1997a, 1997b) and can be grouped into the following categories:
    - **Basic**
        - **Patient** is an entity that is directly affected or effected (produced) by an action.
        - **Zero** is an entity involved in a relation.
    - **Object (or what)**
        - **Instrument (or tool)** is an entity used to affect an action.

- **Location-related (or where)**
  - **Location** is a conceptual object of type of place where an action takes place or where an entity is located.
  - **Source** is an actor from which an entity originates.
  - **Direction** is an actor to which an entity moves.
  - **Path (or route)** is a conceptual object describing the way an entity moves or is oriented.
- **Time-related (or when)**
  - **Speed** is a conceptual object describing how fast an entity moves.
  - **Frequency** is a conceptual object describing how often an action takes place in a certain period of time.
  - **Time point** is a conceptual object describing a specific point in time.
  - **Duration** is a conceptual object describing two points in time during which an action takes place.
- **Stakeholder (or who)**
  - **Beneficiary** is the actor who benefits or is interested in an action.
  - **Receiver** is the actor who receives something as a possession as a result of an action.
  - **Company** is the actor who takes part in an action with the agent.

The relationships between base, structure and role modelling entities can be described by the following overview. Both **institutional actors** and information systems (which are specific types of **artificial actor**) consist of the following elements and relationships:

- **Events** that trigger **actors** to initiate **actions** if certain **conditions** are met
- **Actions** that involve **things** and other **actors**
- **Actions** and **things** that can cause **events**

In Table SM3 the constructs of the ISD modelling techniques are related to the modelling constructs of the integrative modelling technique developed in this research. This table illustrates the discovery and clarification phase of the grounded approach followed. As an example, Table SM2 (a subset of Table SM3) shows how various terminologies are synthesised (or filtered down) to the term “actor” in the integrative modelling technique. Table SM3 also confirms that the constructs in the technique can all be linked back to corresponding concepts in traditional business and ISD modelling techniques.

An in-depth, critical discussion of the existing techniques and the grounded process followed to distil the integrative modelling technique constructs from them, as well as a more detailed explanation of the technique with examples, are given in Joubert (2013). In the following section the application of the modelling technique is illustrated using part of a case study.

5 CASE STUDY

The following case study is an extract of a bigger case study found in the Business Rules Group's document on business rules (Hay & Healy, 2000). They developed the case study together with a number of companies, specifically to cover all possible types of business rules. The case study is of such detailed information that it can also be used to illustrate how to analyse business rules with the purpose of developing an information system from it. The proposed integrative modelling technique is applied to the case study to demonstrate and validate its applicability, serving as a proof of concept. The case study involves a car rental company called EU-Rent. For the purpose of referencing source statements in the case study, each sentence/paragraph in the case study has been numbered in the version below, unlike the original case study. The wording of the case study is exactly the same as the original wording by Hay and Healy (2000, pD.1–D.8). The complete case study can be found in Table SM4. The sections from the case study that will be modelled in this section are given in Table 2.

Table 2: Extract of the EU-RENT case study (Hay & Healy, 2000, pp. D.1-D.8)

1 EU-RENT car rentals

1.1.1 EU-Rent is a car rental company owned by EU-Corporation. It is one of three businesses—the other two being hotels and an airline—that each has its own business and IT systems, but with a shared customer base.

1.1.2 Many of the car rental customers also fly with EU-Fly and stay at EU-Stay hotels.

2 EU-RENT business

2.1.1 EU-Rent has 1 000 branches in towns in several countries. At each branch, cars, classified by car group, are available for rental. Each branch has a manager and booking clerks who handle rentals.

2.2 Rentals

2.2.1 Most rentals are by advance reservation; the rental period and the car group are specified at the time of reservation. EU-Rent will also accept immediate ("walk-in") rentals if cars are available.

2.2.2 At the end of each day, cars are assigned to reservations for the following day. If more cars have been requested than are available in a group at a branch, the branch manager may ask other branches if they have cars they can transfer to him/her.

2.3 Returns

2.3.1 Cars rented from one branch of EU-Rent may be returned to a different branch. The renting branch must ensure that the car has been returned to some branch at the end of the rental period. If a car is returned to a branch other than the one that rented it, ownership of the car is assigned to the new branch.

Sections omitted—see Table SM4.

3.2.4 A customer may have multiple future reservations, but may have only one car at any time.

Sections omitted—see Table SM4.

https://doi.org/10.18489/sacj.v30i1.413

The integrative modelling technique can be introduced fairly easily to non-technical users, because it is in essence just a more formalised extension of natural language. For instance, instead of allowing passive sentences, only active sentences are permitted, sentences must always follow the basic subject-predicate-object/complement form, and only predefined (by the users themselves) words can be used.

The analysis of EU-Rent using this modelling technique is used to illustrate the application of the technique to a typical ISD analysis. The application is illustrative and not exhaustive. Two main phases can be identified: a business modelling phase involving the business user and a systems modelling phase translating the business model into existing ISD modelling structures.

5.1 Phase 1: Business modelling

Step 1: Identify base entities

Question to users to determine actors: Give a list of all the human individuals, institutions and intelligent actors (like IS and smart devices) both inside and outside your organisation with which the proposed system will interface. The outcome of the question to determine actors is shown in Table 3.

Table 3: Identification of actors

Actors:	**Institutional**
	EU-Rent, car rental company, EU-Corporation, business, hotel, airline, EU-Fly, EU-Stay, branch, renting branch, return branch, pickup branch
	Individual
	Branch manager, clerk, customer, driver
	Artificial
	IT systems (1.1.1)

Question to users to determine objects: Give a list of physical objects (things that you can handle and touch), informational objects (objects that store, manipulate, input or display information on any media such as paper, spreadsheets or MSWord documents) and conceptual objects (any classification systems, anything that represents a place and any time-related aspect). The outcome of the question to determine objects is shown in Table 4.

At this stage no questions are asked about acts/relations. They can be determined when model sentences and phrases are developed. The outcome of the question is to determine acts or relations as shown in Table 5 (but this is left undetermined for now).

Step 2: Identify structure and role entities

It is not really possible during analysis to do structure and role entities separately. Role entities are identified during structure-entities analysis as a logical grouping to aid analysis.

Table 4: Identification of objects

Objects:	**Physical** Car **Conceptual** Town (place), country (place), car group, rental period (time), bad experience type (e.g. late return, problems with payment or damage to cars), pickup day (time) **Informational** Rental, transfer, advance reservation, immediate rental, bad experience

Table 5: Identification of acts/relations

Act/relations:	None for now

Question to users to determine models: Divide the proposed system into logical parts (according to any classification—there is no right or wrong). The outcome of the question to determine models is shown in Table 6.

Table 6: Identification of models

Models:	Customer interface model, branch management model

At this stage, no questions will be asked about model views, but during any stage specific views can be derived of the model to highlight just one aspect. Figures SM5-SM7 illustrate possible object, actor and place views.

Question to users to determine model sentences and phrases (per model): What are the static relationships between things (both actors and objects)? The easiest way is to handle each type of relationship separately. The outcome of the question to determine model sentences and phrases is shown in Table 7.

The information in Figures 1–3 were used to analyse the business rules and identify the static relationships above. The outcome of these steps are now used in Figure SM8 to compile a graphical representation of the static relationships between things (both actors and objects).

Table 7: Identification of model sentences and phrases

	Model sentences and phrases
Relationships:	**Client interface model**
	Synonyms
	Future reservation is-a-synonym-for advance reservation (3.2.4)
	Inheritance
	EU-Rent is-a-type-of car rental company (1.1.1)
	Renting branch is-a-type-of branch (2.3.1)
	Pickup branch is-a-type-of branch (2.3.1)
	Aggregation
	EU-Rent has 1 000 branches (2.1.1)
	Association
	Branch has one manager (2.1.1)
	Branch has many booking clerks for rentals (2.1.1)
	Customer has many advance reservations (3.2.4)
	Location (special kind of association involving places)
	Branch is-located-in town (2.1.1)
	Town is-located-in country (2.1.1)
	Ownership (special kind of association involving actors)
	EU-Rent is-owned-by EU-Corporation
	(*can also be expressed as*)
	EU-Corporation owns EU-Rent, EU-Fly, EU-Stay (1.1.1, 1.1.2)
	Properties
	Advance reservation has-properties rental period, car group (2.2.1)

Question to users to determine the dynamic relationships between things (both actors and objects): What are the dynamic relationships between things (both actors and objects)? Here the standard concept of a use case is one of the best ways to define the dynamic relationships between entities, where each use case is defined as a model block. The outcome of the question to determine the dynamic relationships between things is shown in Table 8.

Table 8: Identification of the dynamic relationships between things

	Model sentences and phrases
Actions:	**Client interface model** *Assign cars to reservations* (model block, see 2.2.2) When day ends (event) Then branch manager assigns cars to advance reservations for following day If (number of requested) cars is-greater-than available cars, per car group, at branch Then branch manager (may) request other branches for available cars to transfer

The information in Figures 1-3 were used to analyse the business rules and identify the dynamic relationships above. The outcome of these steps are now used in Figures SM9 and SM10 to compile a graphical representation of the dynamic relationships between things.

If needed, more comprehensive dynamic relationships can be created to show the logical and causal relationships between model phrases. For instance, the model block for rental is shown in Figure SM10.

5.2 Phase 2: Systems modelling

The results of phase 1 can be used to inform traditional ISD processes and techniques. For example, in ERDs objects can be expressed as entities, while static relationships can be expressed as entity relationships. Alternatively, these constructs can be expressed in traditional class diagrams as classes and class relationships (see below). This indicates that constructs and models in this technique can be 'translated' into corresponding existing techniques.

Step 1: Create a data model

Express intelligent and non-intelligent objects in a traditional class diagram as classes, and express static relationships as class relationships (see Figure SM11).

- **Identify entities/classes**. Everything (intelligent and non-intelligent object) identified is a potential entity or class.
- **Identify entity/class relationships**. Every static relationship is a potential entity/class relationship that can be expressed by means of class relationships in a class diagram.

The implementation above using the EU-Rent case study shows that, from an ISD perspective, the integrative modelling technique can be used to model any business aspect of ISD through all the phases of a typical SDLC. Two more perspectives are demonstrated in Joubert (2013):

- The **business rules perspective**, which tests that the modelling technique can model any business rule (or any other rule for that matter).
- The **requirements perspective**, which tests that the most commonly used requirement modelling tool, use cases, can be modelled using the integrative technique.

The proposed integrative modelling technique that was demonstrated above will be evaluated in the next section.

6 DISCUSSION AND EVALUATION

The case study used above serves as a typical example of the application of the proposed technique, which in turn is an improvement of a highly structured systems development approach. At this point it should be acknowledged that there is a turn away from this type of approach towards design thinking in software development research and practice (Luebbe & Weske, 2010; Schryen & Wex, 2012; Hellmuth & Stewart, 2014; Bork, Karagiannis, & Hawryszkiewycz, 2017). In fact, the proposed integrated modelling technique, which was developed using a design science research approach, indirectly contributes to this movement. By integrating business analysis with ISD (albeit in a highly structured way), the new technique may be regarded as a step towards a new, holistic way in which we approach business and information systems modelling. In future work, more exemplars or instantiations could be compiled to validate or improve the integrated modelling technique (Flyvbjerg, 2006). A discipline actually needs an encompassing set of exemplars that confirm (or falsify!) its theories and propositions. In this study a 'most-likely' case was selected strategically to serve as a paradigmatic example that illustrates a proposition:

> A paradigmatic case of how scientists do science is precisely such a prototype. It operates as a reference point and may function as a focus for the founding of schools of thought. (Flyvbjerg, 2006, p. 232)

The study used a grounded approach as a way of doing a qualitative data analysis to inform the integrative model that had to be developed. Except for a few techniques like SSM, no specific theory of ISD modelling entities exists per se, but most of the ISD modelling techniques studied have implicit theories from which these techniques are derived. The study does not consider extant theories as a primary concern, but has as primary concern the purpose of generating a set of fundamental ISD modelling entities with the purpose of creating an integrative modelling technique for further analysis. Substantive theory considers an empirical area of enquiry, while formal theory considers a conceptual area of enquiry (Glaser & Strauss, 1967). This research concentrates on the empirical area of fundamental business and ISD modelling entities and therefore makes an incremental contribution towards substantive theory. All of the base and structure entities come directly out of the grounded process. The detailed role entities do not directly come out of the underlying data, but emerge from the literature study done on linguistics.

The density (Glaser & Strauss, 1967) of the conceptual detail of the theory is difficult to quantify. One way to do it is to evaluate the extensiveness of the suggested codes and their relationships. The comprehensive set of base, structure and role entities, relationships between base and structure entities, and relationships between role and structure entities indicates that the theory is fairly dense in conceptual detail.

The researchers found the initial process of categorisation to be quite subjective and difficult. But once the technique was fairly well developed, one of the best ways to validate that the categories do fit the data (Glaser, 1978) was to test the categories against a paradigmatic case study. Relevance implies that the theory should be relevant to some stakeholders (Glaser, 1978). The integrative technique, and its implied theory, is relevant in that it currently addresses a major problem in business and ISD modelling (i.e., the identified gap between the two approaches).

To evaluate the design science research areas of this study, the eight components of an IS design theory by Gregor and Jones (2007) are used (see Table 9). They argue for design knowledge to be expressed as theory, because it will ensure that IS rises above the level of a craft and it will provide for a “sounder basis for arguing for the rigour and legitimacy of IS as an applied discipline” (Gregor & Jones, 2007, p. 314).

The research makes a contribution towards the integration of business, process, technical and enterprise modelling. It shows that without a clear understanding of ISD modelling entities, there cannot be an integrated approach to modelling, starting from the business side and going all the way to the technical side. Currently no techniques exist to really address modelling from business to technical aspects using only one modelling language. Linguistics is a rich but underdeveloped reference discipline for ISD modelling. Most ISD techniques mention linguistics, but do not really take it to its logical conclusion. The proposed technique relates all the fundamental ISD modelling entities in use today in various ISD techniques to each other, and synthesises these with relevant linguistic concepts. In this way, it provides a unifying explanation of the relationships between these entities, which does not really exist currently. The fundamental modelling entities are amended by procedural steps to complete the technique, thereby complementing the theoretical-methodological contribution.

7 CONCLUSION

Most current business modelling techniques suffer from one of two problems: they are either easily understood by business users (mostly textual), but too simplistic to be used in ISD modelling, or they are usable for ISD modelling (using some existing ISD modelling technique like UML), but too complex for the average business user.

To develop an integrative modelling technique to overcome the gap between business and ISD modelling, it was important to first understand what the fundamental modelling entities in business and ISD are. Using a grounded approach, the research found that these entities follow a linguistic structure and can be divided into three main categories:

Table 9: Evaluation of the integrative modelling technique using the eight components of an IS design theory (Gregor & Jones, 2007)

Components of IS design theories	**Integrative modelling technique**
Purpose and scope (the *causa finalis*)	The purpose and scope is related to business modelling that can be used as the input or source for further ISD modelling.
Constructs (the *causa materialis*)	The base, structure and role modelling entities form the basic constructs of the theory and are clearly defined.
Principle of form and function (the *causa formalis*)	It is shown how the base modelling entities are related in the structure modelling entities to form the building blocks of this technique. These modelling building blocks can be used to communicate business-related analysis and design information.
Artefact mutability	The form (structural aspects) is considered to be fairly immutable, while the function is considered to be very mutable. New base entities and structure entities are unlikely to be developed much more. Role entities can see some growth when used and could increase and improve to some extent. Applying the technique to specific situations is the area where almost unlimited change can occur.
Testable propositions	A certain set of specific base entities and their relationships are stated. These can be tested and if other entities and other relationships can be identified, it will show that the original theory was incorrect or incomplete and can be improved or amended.
Justificatory knowledge	The data analysis techniques of grounded theory and the fundamental structure of linguistics were used as the main contributors to the knowledge base of this study, forming its kernel theories.
Principles of implementation (the *causa efficiens*)	Some guidelines (procedural steps) are provided on how to use this technique in a number of typical business modelling situations.
Expository instantiation	The feasibility to implement the technique is illustrated in the exemplary case study.

- Base entities (corresponding to the morphological level in linguistics), representing the real-world objects that make up organisations and systems as the words of the proposed modelling language. The high-level base entities are things, actors and objects.
- Structure entities (corresponding to the syntactic level in linguistics), using the base entities plus specific language constructions to form model sentences and phrases. The structure entities are models, model views, model sentences and model phrases.
- Role entities (corresponding to the semantic level in linguistics), helping to define meaning to base and structure entities in various situations. Role entities can be classified as subject, predicate and complement roles.

An understanding of the properties and attributes of these entities, together with an understanding of the relationship between these entities, and amended by procedural guidelines, forms a basic method to model business situations both easily and expressively. The research done for this study can be seen as a step in integrating business, process, technical and enterprise modelling. Currently all of these areas of modelling have their own sets of modelling techniques.

In future work linguistics can be explored in more depth as an ISD analysis tool. An exciting possibility is creating software that can generate analyses and/or models from text input, which will simplify the process of business analysis and design. In a fully-fledged system many validations can be done and help provided. For instance, a thesaurus can warn of possible synonyms. Another avenue is to look at the application and adoption of the proposed technique for the problematic area of modelling organisations in enterprise architectures. A third option would be to develop conversion software between existing models. Fourth, there is an opportunity to formalise the findings into an ontology of business and ISD modelling fundamental entities. Such an ontology could define common words in language with respect to this technique where, for instance, all human nouns are indicated as human actors. Normal dictionary definitions of words can be provided as a base from which to create custom definitions for terms. An ontology of the syntactic functions (structure entities) could again be used to enforce consistency when formulating model sentences and phrases (cf. Kroeze (2009)). Lastly, the presentation of the model could be enhanced by the use of colour in a simplified version that is suitable for tuition purposes on undergraduate level. For example, different colours could be used to highlight and trail concepts that lead to the creation of a model block. If the text is displayed in the same colour in the proposed technique, a student or reader should have a better understanding of where the information comes from. Such a more accessible version of the integrated modelling technique could also facilitate the implementation of the proposed solution in industry, thereby improving and simplifying business rule models and facilitating the transformation of business rules into ISD models.

ACKNOWLEDGEMENTS

This article is based on a PhD thesis by Pieter Joubert, who passed away in 2013. The University of South Africa (UNISA) is acknowledged for granting a period of research and development leave to the third author during which time the article was completed. The research was supported in part by the National Research Foundation (NRF) of South Africa. Any opinion, finding and conclusion or recommendation expressed in this material is that of the authors, and neither the NRF, nor UNISA and the University of Pretoria, accepts any liability in this regard. The authors would like to thank the editors and reviewers for their invaluable feedback that guided us to improve the article.

References

Avison, D. E. & Fitzgerald, G. (2006). *Information systems development: Methodologies, techniques and tools* (4th). McGraw Hill.

Bork, D., Karagiannis, D., & Hawryszkiewycz, I. (2017). Supporting customized design thinking using a metamodel-based approach. In Proceedings of the 20th Australasian Conference on Information Systems (ACIS) 2017 (pp. 1–11).

Checkland, P. (2000). Soft systems methodology: A thirty year retrospective. *Systems research and behavioral science*, *17*, 11–58.

Chen, P. P.-S. (1976). The entity-relationship model—Toward a unified view of data. *ACM Transactions on Database Systems*, *1*(1), 9–36. https://doi.org/10.1145/320434.320440

Dik, S. C. (1997a). *The theory of functional grammar: Part 1. The structure of the clause.* Berlin: Mouton de Gruyter.

Dik, S. C. (1997b). *The theory of functional grammar: Part 2. Complex and derived constructions.* Berlin: Mouton de Gruyter. https://doi.org/10.1515/9783110218374

Flyvbjerg, B. (2006). Five misunderstandings about case-study research. *Qualitative Inquiry*, *12*(2), 219–245. https://doi.org/10.1177/1077800405284363

Geerts, G. L. (2011). A design science research methodology and its application to accounting information systems research. *International Journal of Accounting Information Systems*, *12*(2), 142–151. https://doi.org/10.1016/j.accinf.2011.02.004

Glaser, B. G. (1978). *Theoretical sensitivity: Advances in the methodology of grounded theory.* Sociology Press.

Glaser, B. G. & Strauss, A. L. (1967). *The discovery of grounded theory.* Aldine Publishing.

Gould, H. (2016). *Systems analysis and design.* Bookboon.

Gregor, S. & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association of Information Systems*, *8*(5), 312–335. https://doi.org/10.17705/1jais.00129

Hassan, N. R. & Mathiassen, L. (2017). Distilling a body of knowledge for information systems development. *Information Systems Journal*, *28*(1), 175–226. https://doi.org/10.1111/isj.12126

Hay, D. & Healy, K. A. (2000). Defining business rules: What are they really? Last accessed 30 Jun 2008. Retrieved from http://www.businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf

Hellmuth, W. & Stewart, G. (2014). Using enterprise information architecture methods to model wicked problems in information systems design research: Completed research. In Proceedings of the Pacific Asia Conference on Information Systems (PACIS 2014).

Joubert, P. (2013). *Towards an integrative modelling technique between business and information systems development* (Doctoral dissertation, University of Pretoria, Pretoria).

Joubert, P., Kroeze, J. H., & Villiers, C. D. (2013). A grammar of business rules in Information Systems. *TD: The Journal for Transdisciplinary Research in South Africa*, *9*(2), 241–276. https://doi.org/10.4102/td.v9i2.206

Kroeze, J. H. (2009). Bootstrapping an XML schema of syntactic functions into a skeleton ontology. *South African Journal of Information Management*, *11*(3). https://doi.org/10.4102/sajim.v11i3.410

Levy, M. & Hirschheim, R. A. (2012). Removing the positivist straight jacket from Information Systems design science research. In ECIS 2012 Proceedings—European Conference on Information Systems (p. 9).

Luebbe, A. & Weske, M. (2010). Electronic colloquium on design thinking research report no. 3. Last accessed 03 Jul 2018. Retrieved from http://ecdtr.hpi-web.de/report/2010/003/

Luukkonen, I., Korpela, M., & Mykkänen, J. (2010). Modeling approaches in the early phases of information systems development. In ECIS 2010 Proceedings—European Conference on Information Systems.

National Institute of Standards and Technology. (1993). Draft Federal Information Processing Standards Publication 183—IDEF0 Integration Definition for Function Modeling (IDEF0). Last accessed 30 Jun 2018. Retrieved from http://www.idef.com/wp-content/uploads/2016/02/idef0.pdf

Niehaves, B. & Bernd, C. S. (2006). Criticality, epistemology, and behaviour vs. design—Information systems research across different sets of paradigms. In ECIS 2006 Proceedings—European Conference on Information Systems (pp. 50–61).

Object Management Group. (2009). Business Process Modeling Notation (BPMN) version 1.2. Last accessed 30 Jun 2018. Retrieved from https://www.omg.org/spec/BPMN/1.2

Prescott, P. (2015). *Sql: Learn the structured query language for the most popular databases including Microsoft SQL Server, MySQL, MariaDB, PostgreSQL, and Oracle.* CreateSpace Independent Publishing Platform.

Purao, S., Baldwin, C., Hevner, A., Storey, V., & Pries-Heje, J. (2008). The sciences of design: Observations on an emerging field. *Communications of the Association for Information Systems*, *23*, 523–546. https://doi.org/10.2139/ssrn.1281643

Recker, J. (2010). Opportunities and constraints: The current struggle with BPMN. *Business Process Management Journal*, *16*(1), 181–201. https://doi.org/10.1108/14637151011018001

Salam, S. Z. B. A. & Yusof, M. B. M. (2015). Knowledge integration in determining user requirements. In Proceedings of the 9th Malaysian Software Engineering Conference, MySEC 2015 (pp. 112–116).

Schach, S. R. (2004). *An introduction to object-oriented systems analysis and design with UML and the unified process*. McGraw-Hill.

Schryen, G. & Wex, F. (2012). IS design thinking in disaster management research. In Proceedings of the 45th Hawaii International Conference on System Sciences (2012) (pp. 4102–4111). https://doi.org/10.1109/HICSS.2012.391

Sinha, A., Paradkar, A., Kumanan, P., & Boguraev, B. (2009). A linguistic analysis engine for natural language use case description and its application to dependability analysis in industrial use cases. In Proceedings of the 2009 IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2009, Estoril, Lisbon, Portugal, June 29 - July 2, 2009. IEEE Computer Society 2009 (pp. 327–336). https://doi.org/10.1109/DSN.2009.5270320

Umanath, N. S. & Scamell, R. W. (2014). *Data modeling and database design* (2nd). Course Technology.

Vaishnavi, V. & Kuechler, W. (2015). *Design science research methods and patterns: Innovating information and communication technology*. CRC Press. https://doi.org/10.1201/b18448

Vaishnavi, V., Kuechler, W., & Petter, S. (2017). Design science research in information systems. Last accessed 30 Jun 2018. Retrieved from http://www.desrist.org/design-research-in-information-systems/

Wand, Y. (1996). Ontology as a foundation for meta-modelling and method engineering. *Information and Software Technology*, *38*(4), 281–287. https://doi.org/10.1016/0950-5849(95)01052-1

Wand, Y. & Wang, R. Y. (1996). Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, *39*(11), 86–95. https://doi.org/10.1145/240455.240479

Wilcox, P. A. & Gurău, C. (2003). Business modelling with UML: The implementation of CRM systems for online retailing. *Journal of Retailing and Consumer Services*, *10*(3), 181–191. https://doi.org/10.1016/S0969-6989(03)00004-3

Wyssusek, B. (2006). On ontological foundations of conceptual modelling. *Scandinavian Journal of Information Systems*, *18*(1), 63–80.

ZurMuehlen, M. & Indulska, M. (2010). Modeling languages for business processes and business rules: A representational analysis. *Information Systems*, *35*(4), 379–390. https://doi.org/10.1016/j.is.2009.02.006