Title Page

Title: An Extended Conceptualisation and Formal Ontology for Conjecture Mapping

Authors:

1. Dr Peter Reimann
Professor of Education
School of Education and Social Work
THE UNIVERSITY OF SYDNEY
R.439, Education Building A35 | The University of Sydney | NSW 2006
Australia
T +61 2 9351 6365
E peter.reimann@sydney.edu.au

ORCID: http://orcid.org/0000-0002-3869-2239

2. Dr Kate Thompson
Associate Professor of Digital Pedagogies
Faculty of CI, Education & Social Justice,
School of Teacher Education & Leadership
QUEENSLAND UNIVERSITY OF TECHNOLOGY
2 George Street
Brisbane QLD 4000
Australia

T +61 7 3138 3283
E kate.j.thompson@qut.edu.au

ORCID: https://orcid.org/0000-0002-2269-1215

ABSTRACT

Background

As a technique for linking learning theory to learning design, Conjecture Mapping (Sandoval, 2014) plays an important role in Design-based Research. Our goal is to strengthen the main function of conjecture mapping as a boundary object between the designers and researchers.

Methods

Based on a literature review of modifications and extensions to conjecture mapping, we developed an extended conceptualisation, which in turn was formalised as the Design Conjecture Ontology (DCON). Finally, the ontology served as the information model for sharing conjecture maps online.

Findings

The literature review yielded evidence of the widespread use of CM and substantial overlap in the suggestions made for how to extend and refine the technique. Linking to argumentation theory proved productive as it gave conjectures a well-understood epistemic form and function. The DCON ontology gives precision to the terms used and can support the practice of Design-based Research.

Contribution

This synthesis of methodological research on conjecture mapping, taking the form of a formal ontology enables the Learning Sciences to represent one of its pivotal techniques in a coherent form. It paves the way for the sharing and collaborative development of conjecture maps in a consistent digital format and for further advancing the method.

An Extended Conceptualisation and Formal Ontology for Conjecture Mapping

Introduction and Background

In an influential paper published in this journal in 2014, William Sandoval introduced conjecture mapping—and the conjecture map— as a *technique* for Design-Based Research, as "…a means for specifying theoretically salient features of a learning environment design and mapping out how they are predicted to work together to produce desired outcomes" (Sandoval, 2014, p. 19). To this day, it remains one of the few techniques—the practical methods—that explicitly target the nexus of design and research. Since connecting the designing of learning environments to advancing theories about learning is the defining element of Design-based Research, methods such as conjecture mapping are of pivotal importance (Barab & Squire, 2004; Collins, Joseph, & Bielaczyc, 2004).

The technique of conjecture mapping (CM) involves creating a visual display of how *design elements*—that together form the *embodiment* of a high-level *design conjecture*—lead to the emergence of *mediating processes*, which in turn lead to *outcomes*. The high-level design conjecture expresses the central pedagogical claim. *Design conjectures* articulate expectations about how design elements will contribute to the rise of mediating processes, while *theoretical conjectures* express claims about how mediating processes produce learning outcomes. In Sandoval's view, design and theoretical conjectures express claims about causal relations.

It is important to keep in mind what a conjecture map is not: It is neither a fully developed learning design nor a fully-fledged research design. It is, rather, a technique to build bridges between the practices of learning design and learning research. We see CM as akin to what is known in evaluation methodology as formulating the program logic, as unpacking the “black box” between intervention and outcomes (Astbury & Leeuw, 2010). Thus, the technique fits well with other methods for theory-based evaluation (Weiss, 2007). Indeed, it can be seen as aligned with the principles of realist evaluation (Pawson & Tilley, 1997) in cases where the theoretical conjectures state mechanisms for transformations, not just predictions.

This paper has three goals: (1) To demonstrate how providing a formal ontology for CM helps to integrate the various suggestions that have been made over the years to modify/extend CM into one framework, thus enabling the Learning Sciences to represent one of its pivotal techniques in a coherent form. (2) To connect CM to argumentation in a systematic manner, by modelling conjectures as (Toulmin-style) arguments. (3) To pave the way for the sharing sharing and collaborative development of conjecture maps in a consistent digital format.

In the next section we review some of the studies that have used conjecture mapping in the design research process and summarize three papers that proposed extensions to the technique. The middle part of the paper contains a suggestion for how to synthesise the modification and extension suggestions with the original model. In the third section we present an ontology that formalizes the extended conceptualization and argue for its epistemic and practical benefits. The paper closes with suggestions for next steps.

Examples for the Use of Conjecture Mapping

The technique has been steadily used over the years; Google Scholar has its citation rate at 503 (on 8 August 2021). For instance, Wozniak ‘s (2015) paper illustrates the retrospective analytical potential of CM by re-constructing three phases of designing a university course in the health sciences. Lee, Recker, and Phillips (2018) used CM in the context of participatory design with librarians, also as a device to prompt reflections on design constraints. In the study by Boelens, De Wever, and McKenny (2020), the focus of mapping was on formulating and testing design conjectures in the context of designing for open-ended tasks. The mapping was performed by pre-service teachers specialising in vocational education. Another recent study (Moser, Guggemos, & Seufert, 2021), also employed the technique for pre-service teacher education. CM can also be combined with other design methods, as demonstrated in (Tawfik, Schmidt, & Hooper, 2020) for designing a serious game.

As this small sample shows, the CM technique has been used for a wide variety of learning design projects. In addition to application, a (small) number of contributions aim to advance the technique itself. Ma and van Aalst (2014) extended the conjecture map in two ways: by including information about iterative design and research refinements and by including information about the extent to which conjectures are empirically supported in the conjecture map. They suggest a way to include these extensions in the maps, using two different kinds of arrows and a new node type: *challenges*. Both extensions point into the direction of documenting the *trajectory* of design research, not only the starting point or endpoint. This does not only allow to evaluate the credibility of design decisions but also "may inform the broader community's understanding of the research process, which is essential for distinguishing educational design research from educational design" (Ma & Van Aalst, 2014, p. 78). In (Wu & Thanq, 2018) it is also suggested to enable locating evidence in the map, as well as providing a way to represent design principles and design patterns. Finally, Thanq and Wu (2019) argue that CMs should represent two different kinds of evidence: "with-in intervention evidence"—that is, data gathered as part of the design project a CM describes—and "with-out intervention evidence", i.e., findings from prior studies.

Another form of extending conjecture mapping is the integration with learning design methods. For instance, Thompson, Gouvea and Habron (2016) combined CM with design-for-learning (Carvalho & Goodyear, 2014) for designing and analysing a college-level environmental science course. There are many frameworks to help instructors and researchers make sense of the complexity that encapsulates design, learning and teaching (see Muñoz-Cristóbal et al., 2018 for a review of design frameworks). Goodyear & Carvalho (2014) describe the Activity Centred Analysis and Design (ACAD) framework. They state that there are four elements to consider while designing: The social, set and epistemic can be designed (roles and rules; tools and digital and physical learning environment; and the processes of knowledge building, tasks). The fourth element occurs during the enactment of the design (which they refer to as *learn time*). The element referred to as the co-configuration and co-creation of the learning environment describes what learners and teachers actually do. The use of ACAD as a design framework assumes that learning (measured by outcomes or changes over time) is mediated by activity. Activity is then the focus of the design and can be considered to be emergent, and influenced by the three designable elements that situate it (Alhadad & Thompson, 2017).

Researchers have also used this framework to guide the design and the establishment of research and design questions, that guide the collection of data, to inform practice (e.g., Alhadad & Thompson, 2017; Thompson et al., 2016; Thompson & Kanasa, 2017). To do this, Sandoval's (2014) conjecture mapping was added to the ACAD framework to provide a method to identify (1) design conjectures that link designed elements of a learning environment to the desired activity of learners (conjectures about how the design will be enacted); and (2) theoretical conjectures that link the activity taking place in a learning environment with anticipated learning outcomes (conjectures about how people learn).

For the ACAD method, a visual design aid called the ACAD cards has been developed (Yeoman & Carvalho, 2019). The ACAD cards are different colours that correspond to the ACAD framework: green (epistemic design), orange (social design) and yellow (set design). There are also blue cards, and these correspond to learning theories (see Figure 1). A full list of all cards is included in Yeoman & Carvalho (2019).

Figure 1: A typical display of ACAD cards during an design workshop.

Since learning design is in general rich in graphical notations and visual representations (Luckin et al., 2013), embedding conjecture mapping into more specialised design methods, such as ACAD, requires a flexible visual representations of conjecture mapping. This, in turn, necessitates to represent the logic of the technique independent of the (visual and/or textual) notation. The conceptualisation of Conjecture Mapping outlined next aims for separating logic from notation and for including additional elements, such as evidence and challenges, in the technique.

An Extended Conceptualisation

The extended version of conjecture mapping was developed in three iterations: First, some of the naming conventions suggested in the original version were changed. Then the suggestions made in (Ma & Van Aalst, 2014) and in (Wu & Thanq, 2018) were integrated. Finally, the concept of conjecture was elaborated. The result is depicted in Figure 2. Note also that the conceptualisation separates the modelling language from the modelling notation, following best practice (Fill & Karagiannis, 2013). That is to say, the conjecture mapping technique is no longer tied to a particular visual format, be that boxes or tables or trees. Any notation system, visual or otherwise, that is powerful enough to express the distinction made in the model could be used to create a representation of the model. This is also to say that the graphical notation used in Figure 2 is not normative.

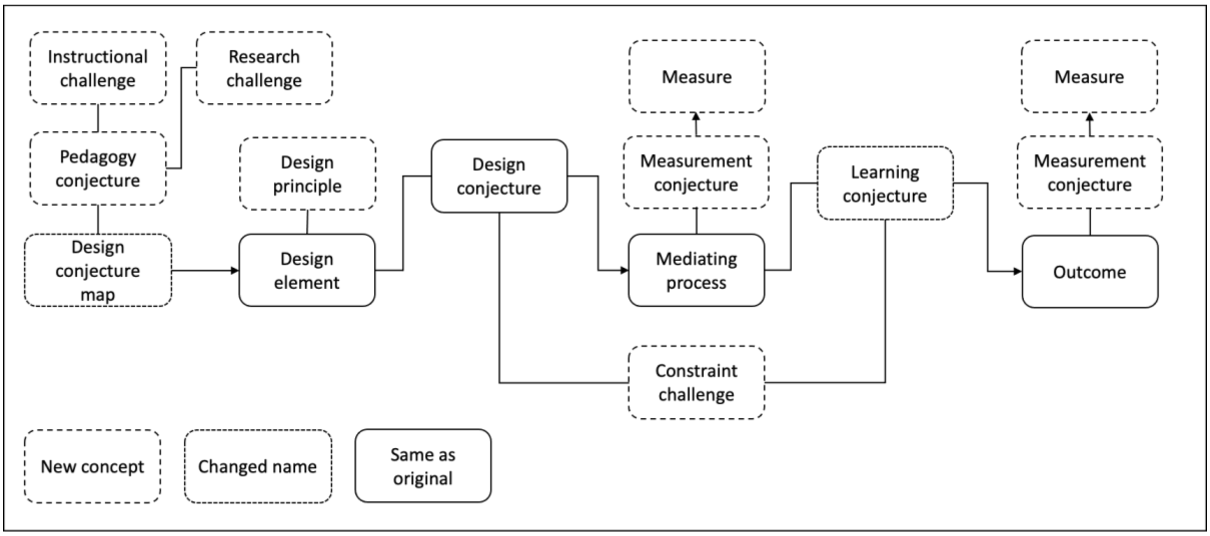

Figure 2: The revised and extended conceptualization.

Changes to terminology

Firstly, the label "embodiment" was dropped; it might lead to confusion with embodied cognition (Shapiro, 2011) and only was a container for design elements. Secondly, *learning conjecture* replaces the term "theoretical conjecture", which was deemed too generic a name.

New elements

Both Ma and van Aalst (2014) and Wun and Thanq (2018) argue for expressing evidence for the validity of conjectures in a CM. This is realised by adding the new elements *measure* and *measurement conjecture* to the vocabulary of conjecture mapping (see Figure 2). A measurement conjecture allows to provide warrants for a why a particular measurement method—a motivation scale, say—is a valid measure for a mediating process or an outcome. As mentioned above, Thanq and Wu (2019) argue that CMs further should distinguish with-in intervention from with-out intervention evidences. With-in intervention evidence can be modelled as measurements, while with-out intervention evidence is modelled as the warrant component of design and learning conjectures. More on this below in the section on argumentation.

The suggestion by Wu and Thanq (2018) to also represent *design principles* in the map is taken up; they enter into relations with design elements. Ma and van Aalst (2014) make a convincing case for representing *challenges* in the CM. Further consideration of their argument led us to distinguish three types of challenges: (1) *instructional challenges*, which motivate the practical side of DBR; (2) *research challenges*, which drive the theoretical work; and (3) *constraint challenges*, which refer to the factors that are largely outside of the designer's and researcher's control but are nevertheless important—for instance, because they can block the intervention from achieving its purpose. Constraint challenges are closest to what Ma and van Aalst were suggesting.

The remaining new elements are all related to the notion of conjecture. This requires its own section because it involved a conceptual elaboration as well as extensions of Sandoval's conceptualisation.

Conjectures as Arguments

Sandoval includes conjecturing in the concept mapping technique because "…the design of learning environments is a theoretical activity, that learning environments intrinsically embody hypotheses about how learning happens in some context and how to support it" (Sandoval, 2014, p. 20). According to the Collins English Dictionary, a conjecture is "is a conclusion that is based on information that is not certain or complete" (2021). Using the concept of 'conjecture' rather than 'hypothesis' is fortunate because it leaves space for design claims that are not all in the form of a 'scientific' hypothesis.

Because conjectures are built on incomplete knowledge, judgment about their correctness is deferred until more is known. Until then—and in general even then—conjectures can be subject to debate: About their correctness, about their usefulness. Not all conjectures have to take the form of hypotheses; at times, conjectures act as *assumptions* based on what is known already, what represents current consensus or is in some form warranted. And not all hypotheses have to be predictions; a hypothesis can be abductive in nature: suggesting a (mechanistic) cause rather than predicting a future state of affairs (Peirce, 1931).

However, *conjecturing* as a form of scholarly practice is not well understood. It is broader than abductive (“how brought about?”) and predictive (“what if?”) reasoning and not subject to normative criteria (“a mere conjecture”). We therefore suggest conceptualising a design research conjecture as an *argument*: The practices of scientific as all as everyday argumentation have been extensively analysed (Schwarz & Baker, 2017; Walton, Reed, & Macagno, 2008), hence there is much more agreement about what makes well-formed arguments than well-formed conjectures. While scientific hypotheses can be seen as a special kind of argument, arguments can also pertain to normative and ethical issues, thus providing the scope needed to keep design decisions ‘open for debate’. By treating conjectures as arguments, it can be left to the conjecture map modeller which interpretation they want to give to their conjectures: rhetorical-persuasive, causal-predictive, causal-abductive, or ethical-normative.

Specifically, we are suggesting to conceive of a conjecture as an argument of the form suggested originally by Toulmin (Toulmin, 2003). In Toulmin notation, a template for a basic conjecture could look like this:

```
Claim:      The <decision for design element d> is justified
Warrant:    because there is previous evidence supporting it from
            studies s1 and s2,
Data:       and because a pilot study showed an increase in x.
Qualifier:  The positive effects may only occur to students with a
            <capacity/aptitude> at <level>.
```

With respect to facilitating the editing, storage and exchange of design conjectures with digital technology, the ontology developed by Rahwan and Sakeer (2006) is useful. It is an extension of the Argument Interchange Format AIF, an ontology widely used to describe arguments in a machine-readable format (Rahwan, Banihashemi, Reed, Walton, & Abdallah, 2011). The elements are described in Table 1.

Table 1: Components of the Toulmin argument scheme as suggested in Rahwan & Sakeer (2006).

Label	**Meaning**
Claim I	The assertion that the argument backs.
Data (D	The evidence (e.g. fact, an example, statistics) that supports the claim.
Warrant (W)	Linking the evidence to the claim; a kind of inference schema.
Backing (B)	The backing supports the warrant; it acts as evidence for the warrant.
Rebuttal I:	A rebuttal is an argument that might be made against the claim
Qualifier (Q):	Qualifies the conditions under which the argument holds
Qualifier-Application (Qapp)	An inference schema linking a Qualifier to a claim
Rebuttal-Application (Rapp)	An inference schema linking a Rebuttal to a claim.

Inheriting from AIF, this ontology for Toulmin-style arguments distinguishes information nodes (C, D, B, R, Q) from nodes that perform inferences (W, Qapp, Rapp). In Figure 3, this is indicated by the nodes' different dashing styles. While inference nodes are described declaratively in AIF, just like information nodes, they are different from information nodes in that in principle they can be executed by an inference engine.

Figure 3 provides an example of what a design conjecture in this format looks like. Note that the Qualifier is the 'logical' place for representing what Sandoval calls "participant structures" and other forms of pre-requisites. For instance, most design elements will only 'work' if the participants have certain literacies developed. Thus, the Qualifier node provides the means to link to what is known about prerequisites, indicators and counter-indicators, based on experience and research. It is also the place to model *constraint challenges*.

More generally, the argument structure makes salient how to link design ideas to research and to scientific argument. Specifically, in the Data node on can describe what would need to be captured in the design research to test the validity of the relation stated in the Claim. Like the Qualifier node, the Backing and Rebuttal nodes are the points where previous research can be brought to bear on the argument, what Chen and Wu (2019) call “without-intervention evidence”.

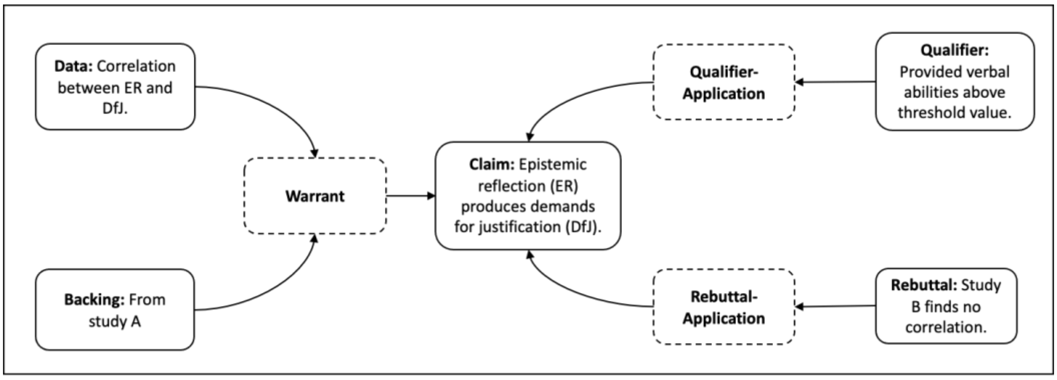

Figure 3: A design conjecture as a Toulmin-style argument. The dashed nodes refer to inference schemes. See text for explanation.

Conjectures, then, are rather ubiquitous in our conceptualisation of a design conjecture map, as can be seen in Figure 2. They come in are four types:

- *Pedagogy conjecture* is new and links what the original method calls “High level conjecture” (what in the new conceptualisation is called *design conjecture map*) to an instructional challenge (also new).
- *Design conjecture* keeps the name and meaning from the original proposal.
- *Learning conjecture* is a new term, replacing the “theoretical conjectures”.
- *Measurement conjecture* is a new term.

This concludes our suggestions for extensions to the original conjecture mapping technique. The new conceptualisation is faithful to the ‘spirit’ of the original one and integrates suggestions for extensions made in the last years. Moving conjecturing closer to argumentation can, we believe, help to increase the rigor of the technique and the quality of debating design and research suggestions. In the next section we suggest a first version of an ontology based on the extended conceptualisation.

A First Formalisation

Describing the design conjecture map in a formal ontology achieves a number of things: For one, we gain greater clarity as to the meaning of the concepts and how they should be used. This in turn supports the (social) process of deliberation that aims at establishing an agreement on the ontology. Secondly, through formalisation we bring conjecture maps closer to being understandable for computers. Thus, computers can be used to provide consistency checks (e.g., are there any logical inconsistencies?), satisfiability checks (such as, are there classes that cannot have instances?), and classify instances automatically into types (Sicilia, Lytras, Sánchez-Alonso, García-Barriocanal, & Zapata-Ros, 2011). Even just the basic functionality of storing CMs in databases rather than in a variety of document formats (Word, PowerPoint, Keynote…) will make it much easier to share, re-use and improve them.

Semantic technologies have been applied to learning design in a number of forms. The most comprehensive endeavour so far has been work on a semantic standard for (e-)learning design, appropriately called Learning Design (Koper & Tattersall, 2005). Of direct relevance to the Learning Sciences is Mizoguchi's and colleagues' work (Mizoguchi & Bourdeau, 2015), in particular the research on theory-aware authoring of CSCL designs (Isotani, Inaba, Ikeda, & Mizoguchi, 2009; Isotani, Mizoguchi, Inaba, & Ikeda, 2010; Isotani et al., 2013). For a systematic overview of semantic web technologies in education see Jensen (2019) and for ontology modelling see Stancin (2020). To our knowledge, no work has been done so far on an ontology for design conjecture mapping.

Background: RDF and OWL

The Resource Description Framework (RDF)[i] is a general framework for describing Internet resources. In RDF, a resource is any object that is uniquely identifiable by a Uniform Resource Identifier (URI). Properties (or attributes) of resources are defined by statements: an object-attribute-value triple, or equivalently a subject-predicate-object triple. RDF statements can be represented in many forms: as 3-tuples, as directed graphs, in an XML-based syntax, or in Turtle notation. Since RDF's data model is based on arbitrary graphs (in the mathematical sense), it is well suited for creating conceptual domain models, including rich semantic descriptions of resources. Despite its openness, it provides for efficient representation, querying and processing of domain models, including automated reasoning. RDF is the data model for the semantic web (Allemang, Hendler, & Gandon, 2020).

OWL (pronounced like the bird) stands for Web Ontology Language. It can be used to author specific domain ontologies. Ontologies are a formal way to describe objects, their properties and relations in a domain. An ontology typically specifies (i) types of objects (or classes), such as vehicles and trucks; (ii) the relation between classes, such as that truck is a sub-class of vehicle; and (iii) properties of objects, such that a vehicle has an engine. Ontologies also can specify the properties of properties; such as data properties (number, text,…) and relational properties (functional, symmetric, transitive, inverse…). Ontologies have been developed for all sorts of domains, from car parts to museum collections to academic publications. They play an important role in the Internet because they provide the—if not standardised than at least widely shared—taxonomies for naming resources (objects and processes) and their properties. OWL is expressed in RDF triples and at the same time is a meta-language for RDF. Since it has formal semantics, it can be used for powerful forms of automated reasoning (a subset of First Order Logic, to be precise). For a non-technical introduction see Uschold (2018).

For developing the two ontologies, we used OWL2[ii] with the Protégé editor[iii].

The Argumentation Ontology

We created a simple version of the upper ontology for arguments, following (Rahwan & Sakeer, 2006). To stay aligned with the AIF language for argument exchange on the Internet (Rahwan et al., 2011), it includes the classes S-Node, I-Node, and RA-Node from AIF, plus the elements from Table 1. The object hierarchy is displayed in Figure 4.

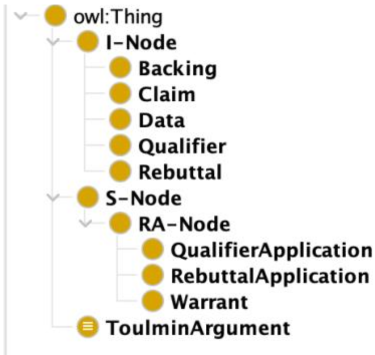

Figure 4: OWL class hierarchy for Toulmin argument ontology. Indentation is used to visualise the sub-class relation.

Not visible in Figure 4 is the property `beliefStatus`; a conjecture can have its current degree of confirmation expressed. Wu and Thanq (2018) suggested as values "fully supported", "partially-supported", and "rejected". They are suitable for empirical conjectures. For non-empirical conjectures a vocabulary like "accepted/rejected" could be used.

The Design Conjecture Map Ontology

The DCON (for *d*esign *con*jecture) ontology is displayed in Figure 5. Since a conjecture is modelled as a Toulmin-style argument, `Conjecture` is sub-classed under `ToulminArgument`. We do not need to represent the different conjecture types that are part of the conceptualization above because there is no difference to their form, and their meaning can be deduced from the context, i.e., to which design element they are attached.

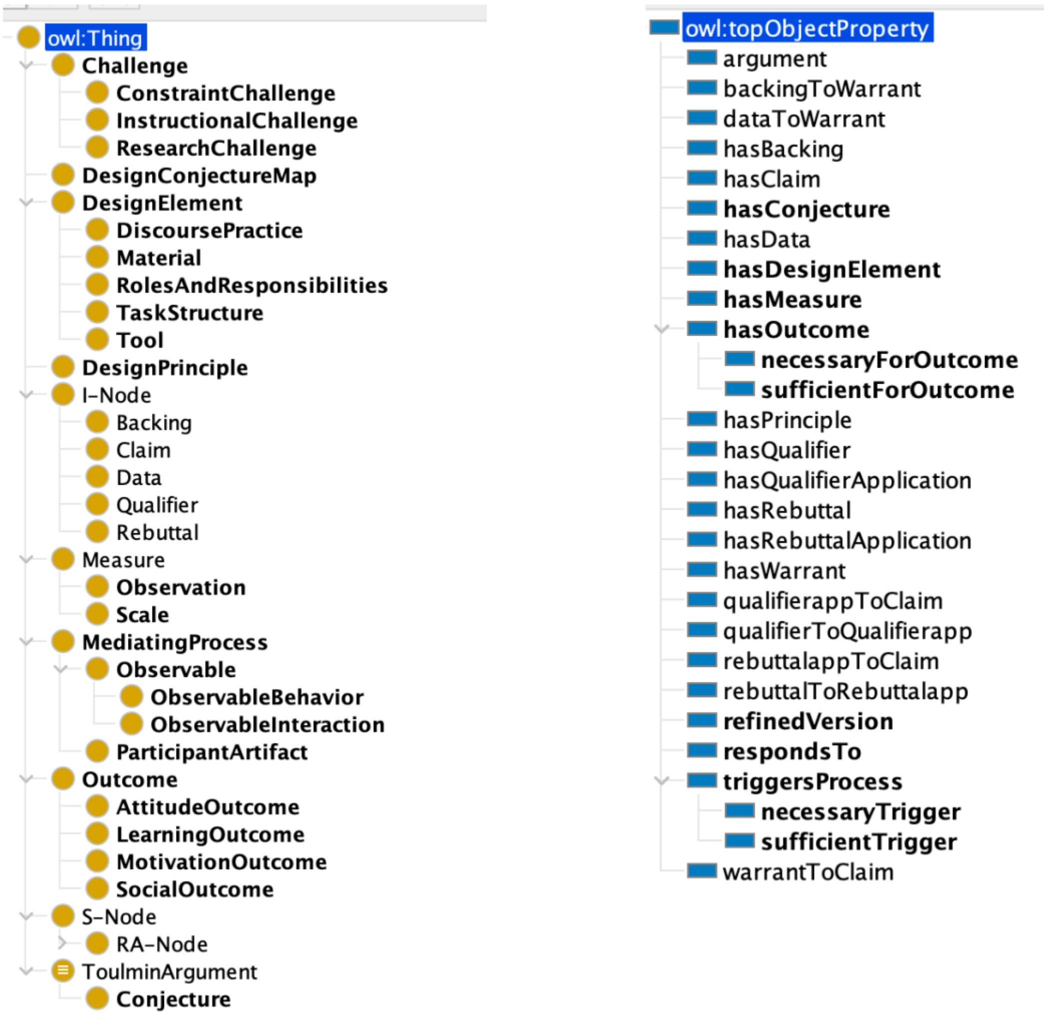

Figure 5: The complete conjecture map ontology, classes on the left, properties on the right

The classes should be pretty much self-explanatory, given that they are modelled around the conceptualization above (Figure 2, Figure 3) and Sandoval's original proposal. Most of the properties shown on the right-hand panel of Figure 5 have to do with the Toulmin-style argument scheme as explained above. Let us look at the properties that apply specifically to CM. There are seven of them:

- `hasConjecture`: Via this property, any element of the design conjecture map can have a conjecture attached. If the element is a learning design node, then the conjecture is about the whole design; if it is a specific design element, the conjecture pertains to that element.

- `respondsTo`: A learning design responds to an instructional and/or research challenge. Using this property, multiple learning designs can be linked to the same challenge(s), constituting alternative or additive solutions to the challenge(s).

- `hasDesignFeature`: This property links a learning design to an element that is part of the design. A learning design must have a least one element. Subclasses of elements are discourse practices, materials, roles and responsibilities, task structures, and tools. (See top of left panel in Figure 5.)

- `triggersProcess`: This property links a design element to a mediating process—it "triggers" that process; at least, that is the expectation (the conjecture).

- `hasOutcome`: Links a mediating process to an expected outcome.

- `hasMeasure`: This property can be used to connect an outcome or a mediating process to one or more measurement methods (e.g, observation, scale, analytics).

- `refinedVersion`: Conjectures and DesignElements can have refined versions. There can be chains of refined versions. This together with the data properties `creator` and `creationTime` allows to model the history of design advancement suggested in (Ma & Van Aalst, 2014).

This concludes the description of the ontology's classes and properties[iv]. While clarifying the meaning of the elements and relations that constitute the process of conjecture mapping and the components of a conjecture map, by itself it does not do any actual work. It is solely the template or the schema for how to structure information. How can this become practically useful?

The Ontology in Action

We can distinguish two kinds of action: Actions *on* the ontology as a knowledge object and actions *with* the ontology as an information model.

DCON as a Knowledge Object

First and foremost, as a formalisation of a conceptualisation DCON is a knowledge object; it is intended as representing the consensus in a field (the Learning Sciences) and in a community of practice (practitioners of Design-based Research) about what conjecture mapping is and what a conjecture map is. Such a consensus needs to be achieved through deliberation amongst the stakeholders and community members; it also needs to be open to revision. We can think of this as a collective process of knowledge production similar to Gerhard Fischer's Seeding-Growth-Reseeding cycle (Fischer et al., 2001), with a fourth stage "Freezing" added. (1) In the Seeding phase, an initial conceptualisation or design is suggested, like done here with DCON. (2) In the Growth phase, the seed is collectively debated and advanced by the stakeholders or in a community of practice. (3) The emerging suggestions are consolidated in the Reseeding phase.

(4) The Freezing stage is needed because for the ontology to be practically useful as an information model it needs to be relatively stable; revision cycles should not be shorter than five years.

DCON as an Information Model

An information model in software engineering "is a representation of concepts and the relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse." (Wikipedia contributors). Used as an information model, DCON provides guidelines for how to store and how to process conjecture maps. This raises the *competency question* (Noy & McGuinness, 2001): What questions do we want to have answered from the information the ontology models? In the design of DCON so far, we were guided by these competency questions:

1. What solutions have been developed for a particular instructional challenge? What answers have been suggested for a particular research challenge?
2. How have these solutions and answers been advanced over time?
3. Where are contributions to design and research most needed?

These questions indicate that we are now in the realm of data; conjecture maps are no longer abstract entities but individual (digital) objects: specific conjecture maps developed to address specific design research problems. To make them machine-processable, these objects are stored in a database as *instances* of one or more of the classes specified in the DCON and the argumentation ontology. They are also given properties that align with their class membership. This is how we ensure that all information in the database—all instances— are described in the language defined by the respective ontologies used. Table 2 shows the RDF triples describing part of a specific conjecture map. (As a non-normative convention, names of instances have a leading underscore for distinguishing them from classes.)

Table 2: Parts of conjecture map described in RDF.

```
:_ConjectureMap01 rdf:type owl:NamedIndividual .
:_ConjectureMap01 rdf:type :DesignConjectureMap .
:_ConjectureMap01 :hasConjecture :_PedagogyConjecture01 .
:_ConjectureMap01 :hasDesignElement :_DesignElementCollaborativeInquiry .
:_ConjectureMap01 :hasDesignElement :_DesignElementReflectiveJournal .
:_ConjectureMap01 :respondsTo :_InstructionalChallenge01 .
```

With this in place, we can demonstrate how to answer the three competence questions. Since conjecture maps are stored in a (RDF) database, to get answers it needs a language for querying these kinds of databases, such as SPARQL[v]. SPARQL queries are straightforward to read because they take the form of matching patterns against data stored as RDF triples. Patterns appear in the WHERE clause while the SELECT clause specifies which variables will be displayed.

Question 1: What solutions have been developed for a particular instructional challenge? What answers have been suggested for a particular research challenge?

For answering this question, we first need to find all challenges in the database. Since challenge is a class we defined in the ontology with three sub-classes (see left pane in Figure 5), we first search for all these sub-classes, then for all the instances of these classes, and finally we gather the links to the conjecture maps that respond to these challenges. Hence, a basic query to answer this question is:

```
Select ?challenge ?challengeType ?cmap
where
{
    ?challengeType rdfs:subClassOf :Challenge .
    ?challenge a ?challengeType  .
    ?cmap :respondsTo ?challenge .
}
```

Question 2: How have these solutions and answers been advanced over time?

To answer this question in a basic form, we simply follow the `refinedVersion` paths in the database:

```
Select ?oldversion ?newversion
where
{
    ?oldversion :refinedVersion+ ?newversion
}
```

This query illustrates a powerful feature of SPARQL, called property graphs: The plus sign after the predicate name tells the query engine to follow the predicate relation to other nodes; in this way, chains of nodes linked by a predicate can be found. In our case, this allows us to find revisions and revisions of revisions.

Question 3: Where are contributions to design and research most needed?

This is not a question that has just one answer; "need for research" can be interpreted in many ways. A simple answer can be provided based on metrics for how well empirically supported a conjecture map is. A learning scientists could then decide to contribute to the research on those maps that have least support, or on those that are just a few steps away from being fully supported. The way we answer this question here is demonstrative of the simplest form of answer at best.

For the case where the belief status for a conjecture is one of "fully-supported", "partially-supported", and "unsupported", we get a ranked list of the conjectures like so:

```
Select ?cmap (ROUND(AVG(?statusScore)) AS ?sumStatus)
where
{ ?cmap a :DesignConjectureMap .
  ?cmap ?element ?status.
  VALUES ?status {"fully-supported" "partially-supported" "unsupported"}
  BIND (IF(?status = "fully-supported", 3, (IF(?status = "partially-
        supported", 2, 1))) As ?statusScore )
}
```

The query re-codes the three belief status categories into the integers 1 (unsupported), 2 (partially-supported) or 3 (fully-supported), and calculates the support score for the whole conjecture map by averaging over the values for its conjectures. A better metric would be the difference to the maximum score, but we want to keep things simple while showing the principles.

The last two queries also illustrate that with SPARQL one can also calculate over, group, and sort RDF data, like one is used from relational databases for tabular data. But this is not the place to dive deeper into the many ways RDF databases can be queried. It is worth pointing out, though, that RDF data travel easily; after all, RDF is a standard for data on the World Wide Web. They can be stored as a file, opened and edited with any text editor, and exchanged via email attachment or shared on a (cloud) server such as Dropbox or Github. In the same format, statements can be imported into RDF database engines and then queried and transformed with languages such as SPARQL. Further, RDF database engines can easily be set up to provide their query functionality to human users on web pages and to other computers (e.g., semantic search engines) as a web service.

Conjecture Map Visualisation

The reader might ask what happened to the compact visualisation used in (Sandoval, 2014) and extended in (Ma & Van Aalst, 2014): the boxes, nodes and arrows? Indeed, since DCON is not modelling the graphical notation of conjecture mapping, but the meaning of the elements and relations used in the mapping technique, the original visual representation is not included. However, we can easily reconstruct it, in principle, because the original graphical notation also forms an abstract graph, nodes connected by directed arcs. ‘In principle’ because it would require a bit of programming to reconstruct the layout and the graphical elements for displaying the map. This is not done here because it is not in the realm of ontology development.

A slightly different kind of visualisation can be produced without programming, exploiting the fact that RDF triples can be represented visually in form of a directed graph. Figure 6 shows an example of a specific conjecture map visualised as a directed graph. Again, to make the comparison with the original conjecture mapping paper easier, the example is similar to the one visualised in Figure 2 of the 2014 Sandoval paper. Reading from the left to the right, there’s the high-level design idea (now encapsulated in a pedagogy conjecture). To the right of it are two design features mentioned (formerly “embodiments”), one of which (Reflective Journal) is expanded. This feature is thought to trigger a mediating process, which in turn leads to a specific learning outcome. Also displayed are three conjectures, one of which (the design conjecture) is expanded.

The graph representation can also serve as an interface for interactively exploring RDF databases. Using applications such as Gruff[vi], the user can navigate through a database by clicking on a node, which will render the immediately connected nodes on the screen, thus making it salient which paths to explore.

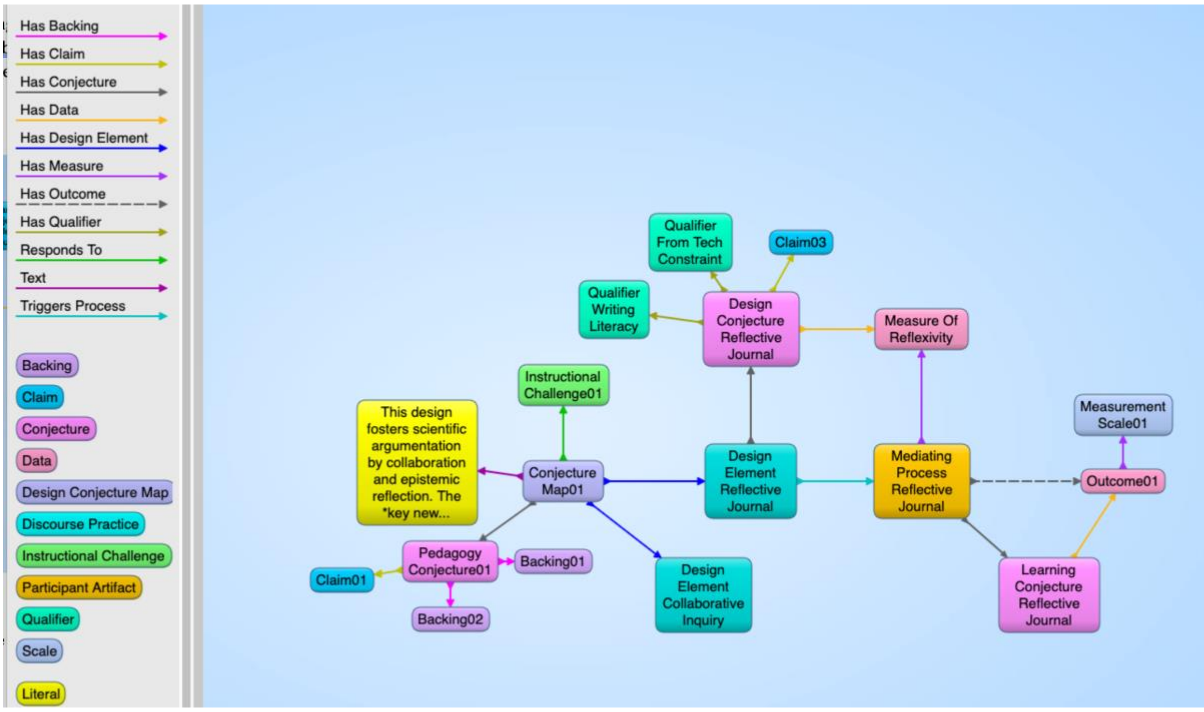

Figure 6: A design conjecture map with one design element expanded. Produced with Gruff™ and AllegroGraph™.

Discussion

To summarize: The first contribution of this paper is a new conceptualisation of the conjecture mapping technique. The conceptualisation integrates a number of modifications and extensions suggested in the literature with the original description of the technique in (Sandoval, 2014); it also connects conjecture mapping with evidentiary argumentation. The second contribution is a suggestion for a formal ontology—the Design Conjecture ONtology (DCON)— that expresses what the elements of the extended conjecture map mean. The third contribution is a demonstration of how to put the ontology to use, using the SPARQL query language.

The intention of this paper is to advance the main function of conjecture mapping: to articulate relations between learning design and learning research—to strengthen the role of the conjecture map as a boundary object (Nicolini, Mengis, & Swan, 2012) between the disciplines of instructional/learning design, educational technology, and the Learning Sciences. Explicating the semantics of conjecture mapping paves the way for, on the one side of the bridge, linking conjecture maps to models of learning designs (Koper & Tattersall, 2005) and, more generally, to semantic education technologies (Jensen, 2019; Mizoguchi & Bourdeau, 2015). On the other side of the bridge, in this form conjecture maps can also be linked to semantic models of research methods and research data (Chalk, 2016). A first step in this direction has been made with DCON, by linking to an upper-level ontology for evidentiary argumentation (Rahwan et al., 2011). More work is needed here, though, because at the time of writing there were no formal models available for the research methods used in the Learning Sciences, nor for the data these methods yield. Also, while there is no shortage of formal models for learning designs and learning content, models of the learning design process and of design methods are rare (Vidal-Castro, Sicilia, & Prieto, 2012).

Conjecture mapping can also serve as a bridge for newcomers to DBR, such as PhD students in the Learning Sciences. While this has not been considered in the literature so far, from personal observations we know that students find it challenging to construct conjecture maps. For instance, students are confused about the distinction between design and theoretical conjectures; they don't understand what is meant by mediating process; and the conjecture maps created are always islands on their own: they never connect to previous research. There is room for making the conjecture mapping technique 'learnable'. Clarity of terminology and a notation that is expressive enough to capture its nuances should be helpful, as is the opportunity to search for conjecture maps in databases connected to each other via the Internet.

While the benefits of formal modelling have been mentioned repeatedly, there are also costs and drawbacks. An important concern is the tension between the creativity needed for good design (as well as for good research!) and the strictures of formal languages. This is, of course, a kind of tension inherent in any design or research method. The more important it becomes to share the outcomes of one's work with others and establish trust in these outcomes, the more sensible it is to rely on trusted methods; they tend to come with an element of rigor. Formal syntax and articulated semantics are also needed to the extent that one wants to use the information infrastructure available today, such as online databases. In general, as one moves from individual practice to collective practice, formal articulation of practice becomes more important and useful.

A more subtle concern is 'factorization': To see the world of learning (design) as a 'bundle of factors', each 'factor' acting independently from others (save for statistical 'interactions') in producing 'learning outcomes'. Sandoval was aware of this:

> *"The final thing to notice about the map is that it claims that the hypothesized mediating processes jointly produce all desired outcomes. (...) ... the complexity of naturalistic learning environments implies that they cannot realistically be decomposed into particular parts that have particular effects."*
> *(Sandoval, 2014, p. 26).*

While a single factor may never be active solely on its own in naturalistic learning environments, and open systems in general, this does not mean that it cannot be isolated for analytical purposes. Fundamentally, parts need to exist for the concept of the whole (and its decomposition) to be possible. DCON is built on parts—design elements, conjectures, etc.—but does not entail strong interdependence assumptions. The concept of Mediating Process allows to model a combination of design elements to produce one or more outcomes. And claims can be made on any level of granularity. The decision of how detailed or comprehensive a claim should be depends on the design/research purpose and context, on theory, and on measurement considerations.

We mentioned already that there are no formal descriptions of research methods used in the Learning Sciences. An important part of future work on conjecture mapping is an extension to model causal relations. The Claim element could be refined to differentiate cause-effect relations as a specialisation of the Claim class, and this Cause class would need to further distinguish between necessary and sufficient causes. Also, in this first suggestion for a design conjecture ontology we are not exploiting the potential that the expression of conjectures in the AIF argumentation ontology is offering. For instance, the relation between conjectures could be much richer than just then refinement relation, depicting the argumentative relations between them (Rahwan et al., 2011).

Further extensions would be needed to account for the forms of design and research discussions that are not argumentative in nature; for those that are, for instance, issues-based and deliberative rather than argumentative (Tempich, Studer, Simperl, Luczak, & Pinto, 2007). Issue-based interactions are more representative of design-thinking, whereas arguments are more typical for research-thinking. And since discursive practices of argumentation and deliberation are sequential in nature, this suggests to model the history of conjecture maps using an event-based ontology of time. (AllegroGraph™, for instance, provides for this[vii].). This would be a rigorous way to realize Ma and van Aalst's (2014) suggestion to capture the trajectory of the design research process in the map.

An ontology is of little use if it doesn't get used, deliberated, and refined. All the materials described here are available on Github[viii]. This repository also can be used for discussion and collaborative development. We are planning to run workshops on semantic modelling of conjecture mapping and other methods pivotal to the Learning Sciences in future conferences of the Learning Sciences Society.

References

Alhadad, S. S. J., & Thompson, K. (2017). Understanding the mediating role of teacher inquiry when connecting learning analytics with design for learning. *Interaction, Design, & Architecture(s), 33*, 54-74. Retrieved from http://www.mifav.uniroma2.it/inevent/events/idea2010/doc/33_3.pdf

Allemang, D., Hendler, J., & Gandon, F. (2020). *Semantic web for the working ontologist*: Assocation for Computing Machinery (ACM).

Astbury, B., & Leeuw, F. L. (2010). Unpacking black boxes: Mechanisms and theory building in evaluation. *American Journal of Evaluation, 31*(3), 363-381. doi:https://doi.org/10.1177/1098214010371972

Barab, S., & Squire, K. (2004). Design-based research: Putting a stake in the ground. *The Journal of the Learning Sciences, 13*(1), 1-14.

Boelens, R., De Wever, B., & McKenney, S. (2020). Conjecture mapping to support vocationally educated adult learners in open-ended tasks. *Journal of the Learning Sciences, 29*, 430-470. doi:10.1080/10508406.2020.1759605

Carvalho, L., & Goodyear, P. (2014). *The architecture of productive learning networks*. New York, NY: Routledge.

Chalk, S. J. (2016). SciData: A data model and ontology for semantic representation of scientific data. *Journal of Cheminformatics, 8*, 1-24. doi:10.1186/s13321-016-0168-9

Collins, A., Joseph, D., & Bielaczyc, K. (2004). Design research: theoretical and methodological issues. *The Journal of the Learning Sciences, 13*(1), 15-42.

Fill, H.-G., & Karagiannis, D. (2013). On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform *Enterprise Modelling and Information Systems Architectures - An International Journal, 8*, 4-24. doi:10.1007/BF03345926

Fischer, G., Grudin, J., McCall, R., Ostwald, J., Redmiles, D., Reeves, B., & Shipman, F. (2001). Seeding, evolutionary growth and reseeding: the incremental development of collaborative design environments. In G. M. Olson, T. W. Malone, & J. B. Smith (Eds.), *Coordination theory and collaboration technology* (pp. 447-471). New York: Erlbaum.

Isotani, S., Inaba, A., Ikeda, M., & Mizoguchi, R. (2009). An ontology engineering approach to the realization of theory-driven group formation. In *International Journal of Computer-Supported Collaborative Learning* (Vol. 4, pp. 445-478).

Isotani, S., Mizoguchi, R., Inaba, A., & Ikeda, M. (2010). The foundations of a theory-aware authoring tool for CSCL design. *Computers and Education, 54*, 809-834. doi:10.1016/j.compedu.2009.09.010

Isotani, S., Mizoguchi, R., Isotani, S., Capeli, O. M., Isotani, N., De Albuquerque, A. R. P. L., . . . Jaques, P. (2013). A Semantic Web-based authoring tool to facilitate the planning of collaborative learning scenarios compliant with learning theories. *Computers and Education, 63*, 267-284. doi:10.1016/j.compedu.2012.12.009

Jensen, J. (2019). A systematic literature review of the use of Semantic Web technologies in formal education. *British Journal of Educational Technology, 50*(2), 505-517. doi:10.1111/bjet.12570

Koper, R., & Tattersall, C. (Eds.). (2005). *Learning design. A handbook on modelling and delivering networked education and training*. Berlin: Springer.

Lee, V. R., Recker, M., & Phillips, A. L. (2018). *Conjecture mapping the library: Iterative refinements toward supporting maker learning activities in small community spaces*. Paper presented at the Proceedings of International Conference of the Learning Sciences, ICLS

Luckin, R., Puntambekar, S., Goodyear, P., Grabowski, B., Underwood, J. D. M., & Winters, N. (Eds.). (2013). *Handbook of design in educational technology*. New York, NY: Routledge.

Ma, G., & Van Aalst, J. (2014). *Facilitating design research by mapping design research trajectories*. Paper presented at the Proceedings of International Conference of the Learning Sciences, ICLS https://repository.isls.org/bitstream/1/1194/1/78-85.pdf

Mizoguchi, R., & Bourdeau, J. (2015). Using Ontological Engineering to Overcome AI-ED Problems: Contribution, Impact and Perspectives. *International Journal of Artificial Intelligence in Education, 26*(1), 91-106. doi:10.1007/s40593-015-0077-5

Moser, L., Guggemos, J., & Seufert, S. (2021). Improving a MOOC to foster information literacy by means of a conjecture map. *International Journal of Learning Technology, 16*(1).

Muñoz-Cristóbal, J. A., Hernández-Leo, D., Carvalho, L., Martinez-Maldonado, R., Thompson, K., Wardak, D., & Goodyear, P. (2018). 4FAD: A framework for mapping the evolution of artefacts in the learning design process. *Australasian Journal of Education Technology, 34*, 16-34.

N.A. (2021). Conjecture. In *Collins English Dictionary*.

Nicolini, D., Mengis, J., & Swan, J. (2012). Understanding the Role of Objects in Cross-Disciplinary Collaboration. *Organization Science, 23*(3), 612-629. doi:10.1287/orsc.1110.0664

Noy, N. F., & McGuinness, D. L. (2001). *Ontology Development 101: A guide to creating your first ontology*. Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. Stanford University.

Pawson, R., & Tilley, N. (1997). *Realistic evaluation*. Los Angeles: Sage.

Peirce, C. S. (1931). *Collected papers of Charles Sanders Peirce, vols. 1-8, edited by C. Harthshorne and P. Weiss.* Cambridge, MA: Harvard University Press.

Rahwan, I., Banihashemi, B., Reed, C., Walton, D., & Abdallah, S. (2011). Representing and classifying arguments on the Semantic Web. *Knowledge Engineering Review, 26*, 487-511. doi:10.1017/S0269888911000191

Rahwan, I., & Sakeer, P. V. (2006). Towards Representing and Querying Arguments on the Semantic Web. *Frontiers in Artificial Intelligence and Applications, 144*, 3-14.

Sandoval, W. (2014). Conjecture mapping: An approach to systematic educational design. *The Journal of the Learning Sciences, 23*(1), 18-36.

Schwarz, B. B., & Baker, M. (2017). *Argumentation Theory for Education*. Cambridge: Cambridge University Press.

Shapiro, L. (2011). *Embodied cognition*. London: Routledge.

Sicilia, M. Á., Lytras, M. D., Sánchez-Alonso, S., García-Barriocanal, E., & Zapata-Ros, M. (2011). Modeling instructional-design theories with ontologies: Using methods to check, generate and search learning designs. In *Computers in Human Behavior* (Vol. 27, pp. 1389-1398).

Stancin, K., Poscic, P., & Jaksic, D. (2020). Ontologies in education – state of the art. In *Education and Information Technologies* (Vol. 25, pp. 5301-5320): Education and Information Technologies.

Tawfik, A. A., Schmidt, M., & Hooper, C. P. (2020). Role of conjecture mapping in applying a game-based strategy towards a case library: a view from educational design research. *Journal of Computing in Higher Education, 32*, 655-681. doi:10.1007/s12528-020-09251-1

Tempich, C., Studer, R., Simperl, E., Luczak, M., & Pinto, H. S. (2007). Argumentation-based ontology engineering. *IEEE Intelligent Systems, 22*, 52-59. doi:10.1109/MIS.2007.103

Thanq, V. C. D., & Wu, J. (2019). *Further refinements of conjecture mapping for design-based research*. Paper presented at the Proceedings of EdMedia + Innovate Learning, Amsterdam, Netherlands:. https://www.learntechlib.org/primary/p/210085/.

Thompson, K., Gouvea, J. S., & Habron, G. (2016). *A design approach to understanding the activity of learners in interdisciplinary settings: Environment and diversity*. Paper presented at the Proceedings of International Conference of the Learning Sciences, ICLS, Singapore. https://45.55.127.102/bitstream/1/331/1/124.pdf

Thompson, K., & Kanasa, H. (2017). *Learning in the STEAM Room*. Paper presented at the AERA, San Antonio, TX.

Toulmin, S. E. (2003). *The uses of argument (updated edition)*. Cambridge: Cambridge University Press.

Uschold, M. (2018). *Demystifying OWL for the enterprise*: Morgan & Claypool.

Vidal-Castro, C., Sicilia, M. Á., & Prieto, M. (2012). Representing instructional design methods using ontologies and rules. *Knowledge-Based Systems, 33*, 180-194. doi:10.1016/j.knosys.2012.04.005

Walton, D., Reed, C., & Macagno, F. (2008). *Argumentation schemes*. Cambridge: Cambridge University Press.

Weiss, C. H. (2007). Theory-Based Evaluation: Past, Present, and Future. *New Directions for Evaluation, 114*, 68-81.

Wikipedia contributors. (01 March 2021). Information model. In Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/w/index.php?title=Information_model&oldid=1009685973

Wozniak, H. (2015). Conjecture mapping to optimize the educational design research process. *Australasian Journal of Educational Technology, 31*(5), 597-612. doi: https://doi.org/10.14742/ajet.2505

Wu, J., & Thanq, V. C. D. (2018). *Refining conjecture mapping for design-based research*. Paper presented at the ICCE 2018 - 26th International Conference on Computers in Education, Main Conference Proceedings. https://repository.nie.edu.sg/bitstream/10497/22839/1/ICCE-2018-767.pdf

Yeoman, P., & Carvalho, L. (2019). Moving between material and conceptual structure: Developing a card-based method to support design for learning. *Design Studies, 64*, 64-89.

---

[i] http://www.w3.org/RDF

[ii] https://www.w3.org/TR/owl2-overview/

[iii] https://protege.stanford.edu/

[iv] For more details see https://github.com/prei007/design-conjecture-mapping/blob/main/design-conjectures.owl.ttl.

[v] https://www.w3.org/TR/sparql11-query/

[vi] https://allegrograph.com/products/gruff/.

[vii] https://franz.com/agraph/support/documentation/current/magic-properties.html#sparql-magic-temporal

[viii] https://github.com/prei007/design-conjecture-mapping