

## Using analogical model formulation with sketches to solve Bennett Mechanical Comprehension Test problems

Matthew Klenk\*, Ken Forbus, Emmett Tomai and Hyeonkyeong Kim

*Qualitative Reasoning Group, Northwestern University, 2133 Sheridan Road,  
Evanston, IL, USA*

*(Received 6 October 2008; final version received 1 October 2009)*

One of the central problems of artificial intelligence is capturing the breadth and flexibility of human common sense reasoning. One way to evaluate common sense is to use versions of human tests that rely on everyday reasoning. The Bennett Mechanical Comprehension Test consists of everyday reasoning problems posed via pictures and is used to evaluate technicians. This test is challenging because it requires conceptual knowledge spanning a broad range of domains, experience with a wide variety of everyday situations, and spatial reasoning. This article describes how we have extended our Companion Cognitive Architecture, which treats analogical processing as central, to perform well over a subset of the Bennett test. We introduce *analogical model formulation* as a robust method for reasoning about everyday scenarios, by analogy with cases that represent prior experiences. This enables a companion to perform qualitative reasoning (QR) without a complete domain theory, as typically required for QR. We introduce *sketch annotations* to communicate linkages between visual and conceptual properties in sketches. We introduce *analogical reference frames* to enable comparative analysis to operate over a broader range of problems than prior techniques. We show that these techniques enable a companion to score reasonably well on a difficult subset of the Bennett test.

**Keywords:** analogical reasoning; qualitative reasoning; sketch understanding; test-based AI

### 1. Introduction

Understanding how to create systems capable of the breadth and flexibility of human common sense reasoning is one of the central problems of artificial intelligence (AI). By its very nature, breadth is hard to evaluate. The approach of psychometric AI (Bringsjord and Schimanski 2003), where human intelligence tests are used to evaluate AI systems, provides a useful methodology for addressing this problem. If a test is considered reasonable for people, then a program's performance on those problems can be used as a way of evaluating progress. One relevant test for common sense reasoning about the physical world is the Bennett Mechanical Comprehension Test (BMCT; Bennett 1969), which is used to evaluate applicants for technical positions. BMCT problems consist of diagrams depicting physical situations, with multiple-choice questions about their qualitative properties. For concreteness, two examples of BMCT questions are illustrated in Figure 1.<sup>1</sup> The BMCT is broad in two ways. First, it involves a variety of domains,

---

\*Corresponding author. Email: m-klenk@northwestern.edu

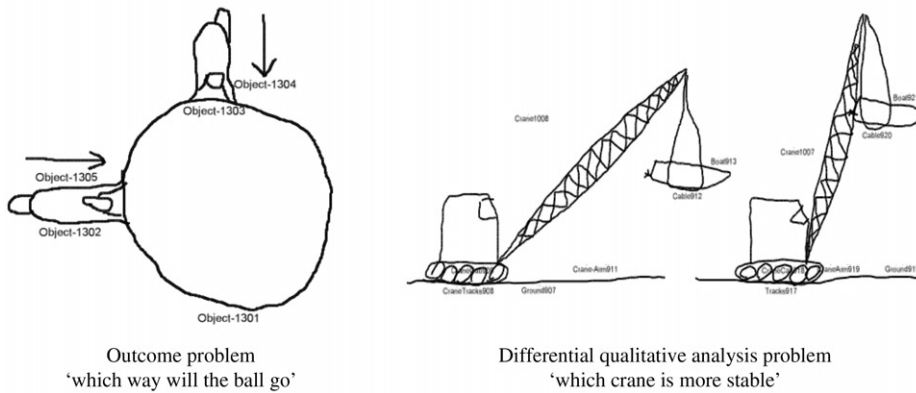


Figure 1. Example Bennett Mechanical Comprehension Test problems.

including statics, dynamics, acoustics, heat and electricity. We call this the *domain breadth* problem. Second, it involves a variety of everyday objects and systems: bicycles, railroad cars, cranes, hoists, boats and many others. We call this the *everyday breadth* problem. This makes it a valuable test for assessing everyday physical reasoning, which is an important part of human common sense.

Figure 1 illustrates the two general types of problems that appear on the BMCT. The ball problem is an example of an *outcome problem*, where a prediction about a property of a system is to be made. The crane problem is an example of a differential qualitative analysis (DQA) problem, where the question concerns a comparison between two (or three) situations, or two aspects of the same situation. To solve problems like these requires moving from a broad set of concepts used in everyday life to a model built from a concise vocabulary of formal abstractions and causal model fragments (Forbus 1984) that can be used to compute the answer. This is an example of *model formulation*, a central problem in qualitative reasoning (QR; Falkenhainer and Forbus 1991). In the case of the ball, the examinee must determine that the two people are applying forces to the ball, that the ball can be considered as a rigid object that is free to move, and so the forces will combine independently, leading to a motion that is to the right and down. In the case of the cranes, the examinee must realise that the stability of the crane depends inversely on the distance between the base of the crane and the boat – the farther out the boat, the less stable the crane is. With this causal model, the question boils down to comparing the corresponding distances on the two cranes, leading to the conclusion that the situation on the right is more stable. For both these examples, many other parameters were potentially relevant: What materials are involved, what kinds of surfaces are they resting on, and so on. But since there was no information in the problem about these differences, one must focus on the differences that are visible, which requires understanding how conceptual properties (like stability) depend on visual properties (like distances).

Existing QR techniques provide important functionality for solving BMCT problems, by enabling conclusions to be drawn without detailed numeric information. In QR terms, BMCT problems can be divided into two aspects: model formulation and computing the answer from the model. As indicated below, computing the answer can typically be done by existing QR techniques, with one or two extensions. The most serious difficulty is in formulating the model. The compositional modelling methodology (Falkenhainer and

Forbus 1991) assumes complete and correct domain theories, and says little about the mapping from descriptions of everyday scenarios to the structural abstractions of the domain theory. QR domain theories have been constructed for many technical domains, which provides some evidence that, in theory, the domain breadth problem could be handled within traditional QR. However, it is important to note that no domain theory of the breadth required to solve the entire BMCT, even factoring out the everyday breadth problem, has yet been constructed. On the other hand, traditional QR approaches are unlikely to provide a solution for the everyday breadth problem. The reason is that the number of entities that can potentially be involved is at least in the tens of thousands, with new ones being invented all the time. The concept of a WiiMote, for example, would not have been in any knowledge base 10 years ago.

The solution we propose is to use analogy to solve the everyday breadth problem and assist with the domain breadth problem. We do not mean cross-domain analogies, such as understanding electricity in terms of water. Instead, we focus here on within-domain analogies, where a new situation is understood in terms of a prior example (e.g. seeing a person pushing a wheelbarrow just like another person pushing a different wheelbarrow or a shopping cart). There are reasons to believe that within-domain analogical reasoning is common within human common sense reasoning (Forbus and Gentner 1997). When faced with a new problem, one is reminded of similar experiences. The explanations for these prior experiences can be used to formulate a model for the new situation. We call this process of using an analogy with a prior example to construct a model *analogical model formulation*.

To see how analogical model formulation can help to solve the domain breadth and everyday breadth problems, it is useful to consider two ways in which examples are understood. First, the prior explanation might have been derived entirely from a domain theory. In this case, the leverage provided by analogy is entirely in solving the everyday breadth problem: once the new problem is understood in terms of relevant abstractions, traditional model formulation can be used. Second, the explanation could be entirely specific to that example. That is, no abstraction of domain entities is performed, the causal models are simply stated in terms of the concrete objects. In this case, the leverage provided by analogy is in solving the domain breadth problem: a causal model that says this specific wheelbarrow is harder to move when the rock is bigger can be applied in a broader range of situations than just the example for which the explanation was given. Both types of explanations can be useful for solving new problems. In fact, the same analogy can be used both to find relevant abstractions for use with a partial domain theory, and for importing example-specific causal models into new situations.

Analogical model formulation fits naturally within the companion architecture (Forbus and Hinrichs 2004), since a primary scientific hypothesis of companions is that analogical processing is central to human intelligence. Analogical model formulation, unlike traditional model formulation methods, has a built-in learning account: learning by accumulating examples. It is well known in the knowledge acquisition community that getting domain experts to tell stories (i.e. concrete examples) is easy, while getting them to articulate complete and correct first-principles rules is extremely difficult. Since extending analogical model formulation requires only adding more concrete examples, this should be a very simple and natural way to learn by experience. We do not claim that learning by accumulating examples is sufficient for capturing the range of human intelligence. However, it is useful to understand how far one can go with a simple technique; so in this article companions only learn via accumulating examples.

This article describes how a combination of three ideas enables companions to perform well on a subset of the BMCT. The first idea of course is analogical model formulation. The second idea is *sketch annotations*, which introduce visual quantities into a sketch in a manner that can be used in causal theories and applied via analogy. The third idea is *analogical reference frames*, for solving DQA problems like the one above. Using analogical reference frames significantly extends the scope of comparative analysis over the traditional perturbed-system model (Weld 1988). While the original BMCT problems use well-drafted drawings, we use hand-drawn sketches as a way of factoring out the less interesting parts of the problem (i.e. low-level image processing, object recognition), while keeping what we view as the essential problems (i.e. spatial and conceptual reasoning). Section 2 reviews the components of the system: sKEA, the sketch understanding system used, the Companions Cognitive Architecture, our models of analogical matching and retrieval, and qualitative mechanics. Section 3 describes the idea of sketch annotations, and also describes how examples are created. Section 4 describes how analogical model formulation works, including a detailed example. Section 5 describes analogical reference frames for DQA. Section 6 puts everything together by showing how these operate within the companion architecture. Section 7 describes an experiment where a companion solves BMCT problems using examples created by others not familiar with the test set. We show that a companion can indeed do reasonably well on a subset of BMCT problems and that its performance improves as the number of examples grows. A detailed failure analysis illustrates the strengths and weaknesses of this approach. Section 8 describes other related work, and Section 9 provides some conclusions and ideas for future work.

## 2. System components

### 2.1. Sketch understanding with sKEA

Sketching is a powerful way to work out and communicate ideas. The *nuSketch* model (Forbus, Lockwood, Klenk, Tomai, and Usher 2004) takes sketching to be a combination of interactive drawing and conceptual labelling. While most sketch understanding systems focus on the problem of recognition, *nuSketch* systems are based on the insight that recognition is not necessary in human-to-human sketching. The sketching Knowledge Entry Associate (sKEA; Forbus and Usher 2002) is the first *open-domain* sketch understanding system. Anything that can be described in terms of sKEA's knowledge base can be used in a sketch. In this study, sKEA's knowledge base consists of a 1.2 million fact subset of Cycorp's ResearchCyc KB,<sup>2</sup> which includes over 38,000 concepts, over 8000 relations and over 5000 logical functions. We have added to that our own representations of qualitative physics, visual properties and relationships, spatial knowledge and representations to support analogical reasoning, but the vast majority of the content that we deal with was independently developed. The breadth of this KB makes it an excellent platform for exploring reasoning in a broad domains such as the BMCT, because the entity types and relations necessary to define problems, such as 'crane' and 'wheelbarrow', are already defined for us as opposed to having to generate them specifically for this project.

*Glyphs* are the basic constituent of sketches. A glyph consists of its *ink*, which represents its visual properties, and its *content*, an entity which represents the thing depicted by the glyph. Content entities can be instances of any of the concepts in the KB. Each sketch is divided into *layers*, which decompose a sketch into pieces. For example, two

systems being compared side by side would be drawn in the same sketch, but each system on a different layer. Sometimes, systems must be viewed at different levels of abstraction. In understanding how a wheelbarrow works, for example, it makes sense to draw the individual parts, since each contributes differently to how it functions. But if we are considering how hard it will be to lift a wheelbarrow, we need to consider the wheelbarrow as a single rigid object. sKEA includes *group glyphs*, which introduce a new entity to represent a selected set of entities, to handle such situations.

Using the ink, sKEA computes the following visual relationships between glyphs (Forbus, Tomai, and Usher 2003):

- Qualitative topological relationships: sKEA uses the RCC8 algebra (Cohn 1996) to describe the connectivity of glyphs in a sketch. Two glyphs might be disjoint (DC), touching exactly on their boundaries (EC), partially overlapping (PO), one completely inside the other (NTPP), or one inside the other but their boundaries touching (TPP). With their inverses, these eight relationships completely characterise the possible connectivity relationships between two 2D regions.
- Positional relationships: Relationships such as above/below and left/right are computed for pairs of glyphs that are adjacent. Adjacency is determined via a Voronoi diagram (Forbus, Tomai, and Usher 2003).
- Visual grouping relations: The RCC8 relationships naturally impose two visual grouping relationships. *Connected glyph groups* consist of sets of glyphs that are pairwise PO or EC with each other. *Contained glyph groups* consist of the set of glyphs which are TPP or NTPP with some larger glyph (called the *container* for the group).
- Orientations: An axis is computed for each glyph, which is characterised as primarily vertical or horizontal, as appropriate.
- Sizes: Each glyph in the sketch is classified as one of five qualitative categories, from tiny to very large, depending on the distribution of glyph sizes in the sketch. This is done by computing the minimum bounding rectangle of each glyph normalised against the minimum bounding rectangle of entire sketch.

Sometimes, the visual relationship between a pair of glyphs and the nature of their contents implies a conceptual relationship between their contents. For example, if a glyph representing a wheel is EC to a glyph representing the ground, then it is reasonable to assume that the wheel is touching the ground. These *visual-conceptual relationships* represent commonly used conventions for depicting situations in sketches (Forbus, Usher, and Tomai 2005). sKEA automatically infers a large candidate set of relationships, and provides an interface for users to select which of them, if any, is appropriate. Figure 2 contains a screen shot of a user selecting the *enmeshedGears-Adjacent* relationship for a pair of gears. The selected conceptual relationship is added to the predicate calculus representation for the sketch.

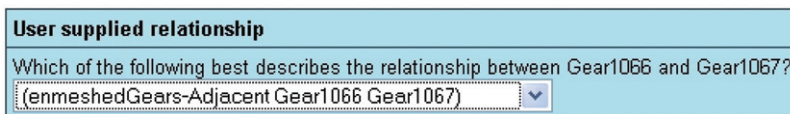


Figure 2. sKEA allows the user to specify the conceptual relationship between sketched entities.

## 2.2. Companion Cognitive Architecture and analogical processing

The Companion Cognitive Architecture is based on the hypothesis that analogical processes are central to human reasoning and learning (Forbus and Gentner 1997). Forbus and Hinrichs (2004) provides an overview of theoretical commitments of the companion architecture. To provide the background for this article, we first briefly outline the architecture and the computational models of analogical matching and retrieval used.

### 2.2.1. Companion architecture

Companions are designed to be software organisms. They are implemented via a distributed agent architecture. Each agent has its own knowledge base, whose contents are periodically updated and synchronised. Communication between agents occurs through KQML messages (Labrou and Finin 1997). In this study, the following configuration of agents was used:

- Facilitator: Manages sessions, starts up other agents and helps set up communication channels between agents.
- Session Manager: Provides generic facilities for user interaction, including startup, shutdown and making queries.
- Sketching Agent: Provides an interface between sKEA and other companion's agents.
- Session Reasoner: Responsible for the domain reasoning, in this case solving BMCT problems.
- Similarity-based Retriever: Provides analogical reminders to the session reasoner based upon the current contents of working memory.

The agent architecture allows specialised reasoning to be distributed to different agents across the entire companion. For example, when the session is started, subscriptions brokered by the Facilitator are set up between the Session Reasoner and the Retriever, so that the Retriever receives updates in the Session Reasoner's working memory. Similarly, when the user makes a change to the sketch altering the Sketching Agent's working memory, subscriptions update the working memory contents in the Session Reasoner. These brokered subscriptions basically become remote procedure calls once set up: If the Session Reasoner needs to know the distance between two points, the query is automatically forwarded to the Sketching Agent, which measures the distance on the sketch and sends the result back to the Session Reasoner.

### 2.2.2. Computational models of analogical processes

We use Gentner's (1983) structure-mapping theory, which postulates that analogy and similarity are based on structural alignment between representations. Given two structured representations (the *base* and *target*), the job of comparison is to find the maximal structurally consistent match. A structurally consistent match is one that satisfies the constraints of *tiered-identity*, *parallel connectivity* and *one-to-one mapping*. Tiered-identity means that there is a strong preference for matching identical predicates, but non-identical matches will be considered under some circumstances. Non-identical functions, for example, can be matched by default in cross-domain analogies, since they tend to represent dimensions of entities involved in the match. Similarly, *minimal ascension* (Falkenhainer 1988) allows non-identical relations to match if they are part of a larger



mapped structure and share a close common ancestor in the ontology. In this study, only identical predicates are allowed to match, since we are concerned with within-domain analogies. Parallel connectivity states that if two statements are matched then their arguments must also match. The one-to-one mapping constraint requires that each element in the base corresponds to at most one element in the target and vice versa. To explain why some analogies are better than others, structure mapping uses the principle of *systematicity*. Systematicity prefers mappings that are highly interconnected, with chains of higher order relations, over mappings with an equal number of relations that are disconnected from each other. The systematicity principle captures a tacit preference for coherence and causal predictive power in analogical processing.

The Structure-Mapping Engine (Falkenhainer, Forbus, and Gentner 1986) models analogical matching. SME takes two structured representations as input (the base and target) and produces one or more *mappings*. Each mapping is represented by a set of *correspondences* between *entities* and *expressions* in the base and target. Mappings also include *candidate inferences* which are conjectures about the target using expressions from the base which, while unmapped in their entirety, have subcomponents that participate in the mapping's correspondences. SME operates in polynomial time, using a greedy algorithm (Forbus, Ferguson, and Gentner 1994).

MAC/FAC (Forbus, Gentner, and Law 1994) models similarity-based retrieval given a case of facts, or *probe*, and a case library of named cases. The first stage uses feature vectors that are automatically computed from the structural representations of the probe and the case library, with each predicate or relation being a dimension of the (unit) vector, with its strength proportional to the number of statements using it in the description. The dot product of these vectors enables the first stage to very rapidly select a few (typically three) candidates from the case library. The second stage uses SME to compare the structured representation of these candidates to the probe, resulting in one (or more, if they are very close) reminding. Both SME and MAC/FAC have been used successfully in many domains, and as cognitive models, both have been used to model a number of psychological results (Forbus 2001).

### 2.3. Qualitative mechanics

Mechanics is traditionally concerned with forces, motion and how bodies interact. Qualitative mechanics provides a set of abstractions (e.g. rigid body, surface normal, qualitative descriptions of angle, etc.) that support QR about traditional mechanics phenomena. We assume the technical vocabulary and model fragments of qualitative mechanics as part of the starting endowment of the system, rather than as something to be learned. Our qualitative mechanics domain theory is drawn from Nielsen (1988) and Kim (1993). Specifically, we use their qualitative representations of objects, surfaces, force transfers and centres of rotation. We summarise the key aspects of the domain theory here.

The domain theory consists of five model fragments, *XForceTransfer*, *YForceTransfer*, *TorqueTransfer*, *TorqueEquilibrium* and *ForceDistribution*. Figure 3 contains the descriptions for *XForceTransfer* and *TorqueEquilibrium*, and the rest are defined in Appendix 1.<sup>3</sup> The *XForceTransfer* model fragment has five participants, the forcer, the object, each of their surfaces and the direction of the force. This model fragment is active when the forcer is applying a force on the object's surface

```

(defModelFragment XForceTransfer
:participants ((TheObject :type RigidBody)
              (TheForcer :type RigidBody)
              (TheSurface1 :type Surface :constraints ((hasSurface TheObject TheSurface1)))
              (TheSurface2 :type Surface
                  :constraints ((hasSurface TheForcer2 TheSurface2)
                               (surfaceContact TheSurface1 TheSurface2)))
              (TheXDir1 :type Sense))
:conditions ((xForceApplied TheSurface1 TheXDir1 TheForcer))
:consequences ((c+ ((QpQuantityFn XNetForce) TheObject)
                ((QpQuantityFn XForceAt) TheSurface1)))

(defModelFragment TorqueEquilibrium
:participants ((TheObject :type RigidBody)
              (TheForcer1 :type RigidBody)
              (TheForcer2 :type RigidBody)
              (TheSurface1 :type Surface :constraints ((hasSurface TheObject TheSurface1)))
              (TheSurface2 :type Surface :constraints ((hasSurface TheObject TheSurface2)))
              (TheRotDir1 :type RotDirection)
              (TheRotDir2 :type RotDirection
                  :constraints ((inverseRotDirection TheRotDir1 TheRotDir2))))
:conditions ((torqueApplied TheSurface1 TheRotDir1 TheForcer1)
            (torqueApplied TheSurface2 TheRotDir2 TheForcer2))
:consequences ((qprop ((QpQuantityFn ForceAppliedToSurfaceBy) TheSurface1 TheForcer1)
                      ((QpQuantityFn ForceAppliedToSurfaceBy) TheSurface2 TheForcer2))
              (qprop ((QpQuantityFn ForceAppliedToSurfaceBy) TheSurface1 TheForcer1)
                      ((QpQuantityFn DistanceToOrigin) TheSurface2))
              (qprop- ((QpQuantityFn ForceAppliedToSurfaceBy) TheSurface1 TheForcer1)
                      ((QpQuantityFn DistanceToOrigin) TheSurface1)))

```

Figure 3. Example model fragments from our qualitative mechanics domain theory.

and results in the force being added to the net force on the object. The `TorqueEquilibrium` model fragment has seven participants and is active when there are two applied torques in opposite directions. While this definition is a simplification<sup>4</sup>, it is sufficient for our purposes. When a `TorqueEquilibrium` model fragment is active, its consequences are believed. These consequences are qualitative proportionalities describing causal functional dependencies between quantities. Since the Cyc KB uses a relational form for quantities, we use a function, `QpQuantityFn`, to convert them to an equivalent fluent. Here two functions are used to represent the four quantities affected by the model fragment. The function `(QpQuantityFn ForceAppliedToSurfaceBy)` takes two arguments and denotes the amount of force applied to a surface (its first argument) by an object (its second argument). The function `(QpQuantityFn DistanceToOrigin)` denotes the distance from a surface to the point it rotates around. When instantiated, these model fragments provide causal relationships (i.e. qualitative proportionalities) needed to solve BMCT problems.

Notice that the qualitative mechanics domain theory is defined in terms of a technical vocabulary of abstract concepts, including `RigidBody`, `Surface` and `Fulcrum` as well as the relationships `xforceApplied` and `torqueApplied`. Our domain theory includes inference rules which, given conceptual relationships between sketched objects (e.g. `canPivotAround`, `touchesDirectly` and `on-Physical`), can conclude the `xforceApplied` and `torqueApplied` relationships. However, since problems on the BMCT are given in terms of everyday situations, the appropriate abstractions must be inferred in order to determine which model fragments are applicable. Section 4 describes how this is done via analogical model formulation.

### 3. Sketch annotations

In everyday sketching, people annotate sketches of physical entities with conceptual information that would not appear in the actual situation. In architectural drawings,



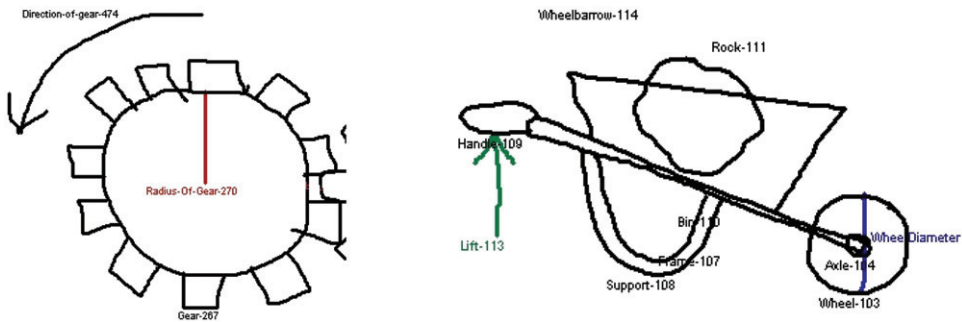


Figure 4. The gear has a rotational annotation indicating counter-clockwise motion and a linear annotation indicating its radius. The wheelbarrow has a force annotation indicating an applied force upward on the handle and a length annotation indicating the diameter of its wheel.

annotations indicate distances between walls and the widths of windows. In sketches explaining principles, annotations indicate important properties, such as physical quantities (e.g. the radius of a gear) and where forces are applied. *Annotation glyphs* provide this capability in sKEA. Like other glyphs, an annotation glyph consists of its ink and its content, i.e. the entity it is representing. However, annotation glyphs also refer to one or more other glyphs in the sketch, indicating the entity (or entities) about which they are providing information. We call these glyphs the *references* for the annotation glyph.

In this study, we use three types of annotation glyphs. *Force annotations* indicate the location and direction of an applied force on a reference. sKEA computes the direction and application surface of a force annotation from its ink and referenced glyphs. If there are two references, sKEA uses the direction of the arrow and the relative positions of the references to determine which object is applying the force. If there is only one reference (e.g. the handle from the wheelbarrow example), sKEA assumes a new object which is applying a force at the point of the arrow onto the reference. *Rotational annotations* indicate a reference's direction of rotation. sKEA assumes the qualitative rotational motion of the reference as either clockwise or counter-clockwise. *Linear annotations* indicate linear distances, either along a single reference or between two references. Two special subclasses of linear annotations are *X-coordinate* and *Y-coordinate annotations*, which refer to the projection of the measurement onto the appropriate axis. sKEA computes distance measurements using the closest *anchor points* on the reference(s) to the endpoints of the linear annotation. Anchor points are used to specify which parts of the reference(s) that the annotation is tied to. Each glyph has nine anchor points: the centroid, the rightmost top, leftmost top, top rightmost, and so on clockwise around the glyph. Anchor points provide symbolic descriptions that can be projected as candidate inferences from an example to a new situation (e.g. the distance from the left bottommost point to the right bottommost point of a reference). Figure 4 contains sketches that illustrate each type of annotation.

### 3.1. Creating examples using annotation glyphs

Examples in the companion's case library represent its experience. Users create examples of physical scenarios using sKEA and a concept map system. The process begins with the

user drawing the scenario, labelling their glyphs with the everyday concepts they would use to describe the entities depicted. Next, the user sketches force and rotation annotations where applicable in the sketch. They use sKEA's conceptual labelling interface to identify all of the appropriate qualitative mechanics abstractions (e.g. `RigidObject`) and sKEA's visual/conceptual relationships interface to identify the appropriate conceptual relationships between entities (e.g. `touchesDirectly`). At any time, the user can invoke a traditional model formulation algorithm (Forbus and de Kleer 1993) to see if the appropriate qualitative mechanics model fragments are instantiated. Once they are satisfied with the model fragments, the final step is to create causal models that describe conceptual quantities in terms of visual quantities. For each relevant conceptual quantity (e.g. the revolution rate of a gear), the user draws a linear annotation for the causally related visual quantity (e.g. the distance from the centre of the gear to its top right most point). Next, using the concept map system, the user adds a qualitative proportionality linking the conceptual quantity to the visual quantity (e.g. that the revolution rate is qualitatively proportional to the radius of the gear). This completes the process of constructing the scenario model for the example. Note that the model fragments, if any, are instantiated from an incomplete domain theory, while conceptual/visual quantity causal relationships are defined in an example-specific manner. All of the visual and conceptual representations for the example are stored as a case in a library, to be used for subsequent analogical model formulation.

To illustrate this process more concretely, we use the wheelbarrow in Figure 4 as an example. First, a user draws the wheelbarrow in sKEA. To do this, the user draws seven glyphs representing the wheel, axle, bin, frame, support, handle and rock. Next, a group glyph is created representing the wheelbarrow including all the glyphs except the rock. The wheelbarrow's handle is annotated with a force arrow indicating that there is an assumed force in the upward direction. Then, the user adds qualitative mechanics abstractions by labelling the wheelbarrow group glyph and the rock as instances of the type `RigidObject` and the axle as a `Fulcrum`. sKEA's visual/conceptual relations interface is then used to add some of the conceptual relationships needed by the qualitative mechanics domain theory. Here, the user stated that the rock is `on-Physical` the bin. The domain theory includes rules to determine that the `on-Physical` relationship results in a downward force from the rock onto the bin. A standard model formulation algorithm instantiates whatever model fragments are appropriate, based on these abstractions and relationships. This results in the instantiation of a `ForceDistribution` and two `TorqueEquilibrium` model fragments. The `ForceDistribution` consists of the rock pushing down on the wheel and the assumed object in contact with the handle. The two `TorqueEquilibrium` model fragments are symmetric. In one, `TheForcer2` is the assumed object in contact with the handle, and `TheForcer1` is the rock. In the other `TorqueEquilibrium` model fragment, the participants are reversed. The consequences of the first model fragment appear in Figure 5. The first qualitative proportionality states that the force applied by the rock on the surface between the rock and the bin is proportional to the force applied on the handle by the assumed force. Surfaces are defined using the function `ContactSurfaceFn` which takes two arguments. The term denotes the surface of the first argument that is in contact with the second argument. For example, `(ContactSurfaceFn Bin-110 Rock-111)` denotes the surface of `Bin-110` which is in contact with `Rock-111`. We use `(ContactObjectFn Handle-109)` to denote the assumed object in contact with the handle. The next two statements say that the force applied on the handle is proportional to

```

(qprop
  ((QpQuantityFn ForceAppliedToSurfaceBy) (ContactSurfaceFn Handle-109
                                             (ContactObjectFn Handle-109))
    (ContactObjectFn Handle-109))
  ((QpQuantityFn ForceAppliedToSurfaceBy) (ContactSurfaceFn Bin-110 Rock-111)
    (ContactObjectFn Handle-109))
(qprop
  ((QpQuantityFn ForceAppliedToSurfaceBy) (ContactSurfaceFn Handle-109
                                             (ContactObjectFn Handle-109))
    (ContactObjectFn Handle-109))
  ((QpQuantityFn DistanceToOrigin) (ContactSurfaceFn Bin-110 Rock-111)))
(qprop
  ((QpQuantityFn ForceAppliedToSurfaceBy) (ContactSurfaceFn Handle-109
                                             (ContactObjectFn Handle-109))
    (ContactObjectFn Object-109))
  ((QpQuantityFn DistanceToOrigin) (ContactSurfaceFn Handle-109
    (ContactObjectFn Handle-109))))

```

Figure 5. Resulting qualitative proportionalities from the torque equilibrium model fragment.

```

Automatically added by sKEA
  (visualRepresentationOfQuantity
    ((ConceptKnownAsFn "WheelDiameter")
      (GlyphFn Wheelbarrow-114 User-Drawn-Sketch-Layer-114))
    (AnnotationGlyphFn WheelDiameter-203 User-Drawn-Sketch-Layer-114))
  (startPointOf
    (AnnotationGlyphFn WheelDiameter-203 User-Drawn-Sketch-Layer-114)
    (RightmostBottomPointFn
      (GlyphFn Wheel-103 User-Drawn-Sketch-Layer-114)))
  (endPointOf
    (AnnotationGlyphFn WheelDiameter-203 User-Drawn-Sketch-Layer-114)
    (LeftmostTopPointFn
      (GlyphFn Wheel-103 User-Drawn-Sketch-Layer-114)))

Entered manually through the Concept Map
  (qprop ((QPQuantityFn RideSmoothness) Wheelbarrow-114)
    ((ConceptKnownAsFn "WheelDiameter")
      (GlyphFn Wheelbarrow-114 User-Drawn-Sketch-Layer-114)))

```

Figure 6. A subset of the facts (simplified) to represent the wheel diameter annotation and the causal relationship between the wheel diameter and the wheelbarrow's smoothness of ride.

the distance from the surface of the rock to the origin of rotation and inversely proportional to its distance from the origin of rotation.

Before the example is complete, any causal dependencies of conceptual quantities on visual quantities must be entered. All of the entities and quantities in the scenario model are automatically added to the concept map interface. Similarly, every time an annotation glyph is used to create a visual quantity, that quantity is automatically added to the concept map interface. In this example, the user might want to convey that the wheelbarrow's smoothness of the ride is determined in part by the diameter of its wheel. To do this, he first adds a linear annotation to the wheel. Next, using the concept map interface, he enters the qualitative proportionality between the quantity representing the wheel's diameter and the wheelbarrow's smoothness of ride. Figure 6 shows a portion of the symbolic representation for the annotation glyph and the internal representation of the qualitative proportionality entered via the concept map. The first fact links the visual quantity, named 'Wheel Diameter' by the user, to the annotation glyph. The next two facts indicate the start and end points of the annotation glyph, in terms of anchor points on the reference glyph. In this case, the start and end points are the bottom right point and the top left point of Wheel-103, respectively. The final fact shows the qualitative

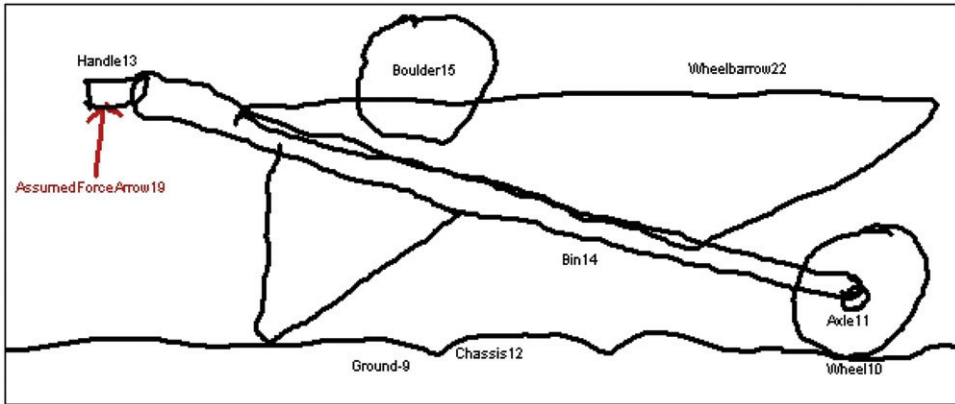


Figure 7. A wheelbarrow from a problem situation.

proportionality between the quantity representing the wheel's diameter and the wheelbarrow's smoothness of ride.

To summarise, examples include three types of information:

- (1) The everyday entities represented in the sketch.
- (2) Instances of model fragments, constructed automatically from the sketched entities and relationships, using the conceptual labelling and the visual/conceptual relationship interfaces to provide the necessary abstractions for model formulation.
- (3) Example-specific causal relationships between visual quantities (measurable in the sketch) and conceptual quantities.

Section 4 shows how these examples can be used via analogy to construct scenario models in new situations.

#### 4. Analogical model formulation

Analogical model formulation creates a scenario model for a new situation by analogy with a previously understood example. The process begins with MAC/FAC being used to retrieve a relevant example. SME then creates a mapping between the example and new situation. This mapping includes a set of candidate inferences which suggest modelling decisions for the new situation. For BMCT problems, candidate inferences provide qualitative mechanics abstractions and relationships, definitions for visual quantities and causal relationships. Together, this information provides the basis for a scenario model which can be used to solve BMCT problems.

Here, we show how analogical model formulation constructs a scenario model for the wheelbarrow shown in Figure 7 using the example wheelbarrow from Section 3.1. SME creates a mapping, a portion of which is shown in Table 1, between the predicate calculus representations of the problem and the example. Because SME is guided by common structure, the resulting mapping does not necessarily include all of the entities in the problem and example. Expressions from the example that do not participate fully in the

Table 1. Mapping between the problem entities and the example.

Example entities (base)	Problem situation entities (target)
Wheelbarrow-114	Wheelbarrow-22
Rock-111	Boulder-15
Handle-109	Handle-13
Frame-107	Chassis-12
Bin-110	Bin-14
Axle-104	Axle-11
Wheel-103	Wheel-10
Lift-113	AssumedForceArrow-19
Support-108	[unmapped entity]
WheelDiameter-203	[unmapped entity]
[unmapped entity]	Ground-9

mapping become candidate inferences, in which the mapped portions are replaced by the corresponding expressions in the problem. Analogical model formulation depends upon these candidate inferences to make modelling decisions in the problem scenario.

Analogical model formulation uses the example to infer three types of information about the problem scenario: causal models, qualitative mechanics abstractions and relationships and information regarding the measurement of visual quantities. Causal models are inferred from the example as follows. The qualitative proportionalities in Figures 5 and 6 become candidate inferences with the entities for `Wheelbarrow-114`, `Handle-109`, `Wheel-103`, `Bin-110` and `Rock-111` replaced with `Wheelbarrow-22`, `Handle-13`, `Wheel-10`, `Bin-14` and `Boulder-15`, respectively. Our system searches the candidate inferences for qualitative proportionalities and assumes their expressions into the problem representation. Qualitative mechanics abstractions and relations are inferred in the same way. Candidate inferences concerning abstractions and relations are assumed into the problem (e.g. (`isa Axle-11 Fulcrum`) and (`on-Physical Bin-14 Boulder-15`)). Visual quantity measurement information is imported in two ways. First, measuring `DistanceToOrigin` quantities requires qualitative mechanics knowledge about the centre of rotation. For example, calculation of the distance to the origin from the surface between the boulder and the bin requires the knowledge that the axle is the fulcrum. The second type of visual quantity concerns user defined annotations, such as `WheelDiameter-203`. In this case, the expressions in the example concerning the annotation become candidate inferences. Since the entity for the annotation `WheelDiameter-203` does not participate in the mapping, these candidate inferences contain `AnalogySkolemFn` expressions. These expressions represent entities which appear in the base but do not have a corresponding entity in the target. We use these candidate inferences to automatically create a corresponding annotation in the problem. Figure 8 contains the candidate inferences which define the wheel diameter quantity and provide instructions as to how to draw the annotation based upon anchor points. Using this automatically constructed annotation, the scenario model includes information concerning the measurement of a visual quantity that was defined only in terms of the example.

In Section 5, we describe how we use analogy to frame comparative analyses allowing for its application between scenario models necessary for solving BMCT problems.

```

(visualRepresentationOfQuantity
  ((ConceptKnownAsFn "WheelDiameter")
   (GlyphFn Wheelbarrow-22 User-Drawn-Sketch-Layer-114))
  (AnalogySkolemFn
   (AnnotationGlyphFn WheelDiameter-203 User-Drawn-Sketch-Layer-114)))
(startPointOf
  (AnalogySkolemFn
   (AnnotationGlyphFn WheelDiameter-203 User-Drawn-Sketch-Layer-114))
  (RightmostBottomPointFn
   (GlyphFn Wheel-103 User-Drawn-Sketch-Layer-114)))
(endPointOf
  (AnalogySkolemFn
   (AnnotationGlyphFn WheelDiameter-203 User-Drawn-Sketch-Layer-114))
  (LeftmostTopPointFn
   (GlyphFn Wheel-103 User-Drawn-Sketch-Layer-114)))

```

Figure 8. Candidate inferences concerning the wheel diameter visual quantity.

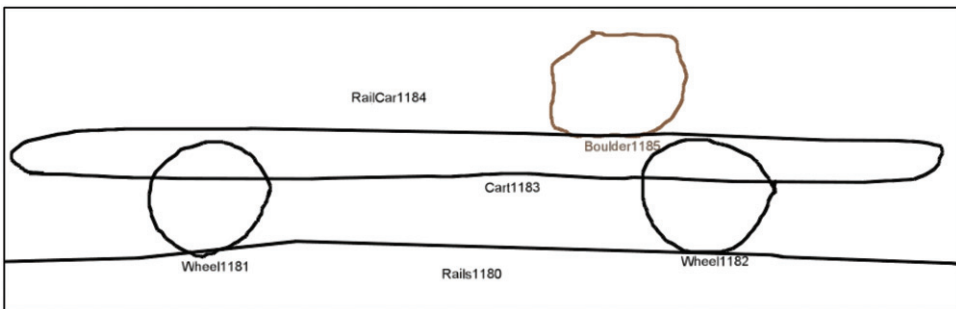


Figure 9. A DQA problem concerning aspects of the same system, ‘which wheel of a railcar presses harder on the rail?’

## 5. Analogical reference frames for comparative analysis

*Comparative analysis*, and in particular DQA, seeks to understand how and why the behaviour of a system will change given some changes to its parameters (Weld 1988). For example, comparative analysis can explain why the period of an oscillating block system would increase if the mass of the block was increased. A *DQ value* is a qualitative description of how one particular parameter in a system will change given other parameter changes. There are four possible DQ values: unchanged, increased, decreased and ambiguous. The DQA problems on the BMCT do not fit directly into the traditional perturbed system framework. First, they involved comparisons between scenarios, rather than describing a perturbation in a single scenario. Second, some of the problems require comparisons between different parts of the same system. For example, the BMCT problem in Figure 9 asks ‘which wheel of a railcar presses harder on the rail?’

We found that analogy provides a general mechanism for framing comparative analyses. Using SME, we create an *analogical reference frame* to determine correspondences between either two systems, or different aspects of the same system. These correspondences frame a comparative analysis problem by defining what each parameter is compared against. In problems with multiple systems, we use SME with one of them as the base and the other as the target. In the case of single system problems, the system is



compared with itself while constraining the entities being compared to match each other. For instance, in the rail car problem shown in Figure 9, SME matches the scenario to itself but requires that `Wheel-1181` corresponds with `Wheel-1182`.

The analogical reference frame lines up quantities whose differences can be reasoned about via standard DQA. Returning to Figure 9, we want to find the DQ value of the force applied to the rail at `Wheel-1181` through a comparison with its corresponding quantity, the force applied to the rail at `Wheel-1182`. Using our causal models to find the dependent parameters, we find that the force on a wheel is inversely qualitatively proportional to the distance between the surface of the boulder and that wheel. Since the two distances are aligned by the analogical reference frame, we determine that the DQ value for the distance concerning `Wheel-1181` is decreased. Since the relationship to force is inversely qualitatively proportional this distance, the DQ value for the force applied at `Wheel-1181` is increased, i.e. `Wheel-1182` is pressing harder on the rail. Analogical reference frames are important because they allow a wider class of systems to be analysed, since the correspondences between aspects of a problem are computed dynamically.

## 6. Solving BMCT problems via analogy

Here we describe how a companion uses sketch annotations, analogical model formulation and analogical reference frames to solve BMCT problems from examples. Problems are presented as sketches of the situation and a query. The session reasoner solves these problems using the AND/OR suggestion architecture from (Forbus and de Kleer 1993). The problem-solving knowledge consists of 19 methods and 136 backchaining rules. Solving BMCT problems using analogical model formulation involves three steps. First, the companion retrieves a relevant example. Second, the companion creates a scenario model based upon the example using analogical model formulation. Third, the companion uses the model to compute the answer.

Figure 10 shows how this process is implemented on the Companion Cognitive Architecture. The Session Reasoner performs the majority of the reasoning. It relies on the Retriever to find relevant analogues and the Sketching Agent to perform visual quantity measurements. The Sketching Agent also maintains the concept map, in addition to running sKEA.

As previously discussed, there are two types of questions on the BMCT: outcome questions and DQA questions. Figure 11 contains an example of each type and the associated query. The predicate of the query indicates the type of problem. Because the ball problem is a single situation, it is sketched on a single layer. The predicate `solveQMOneSketchProblem` designates that this is an outcome question. The first argument is the query for the outcome question, ‘what is the net force on the ball?’ In DQA problems with comparisons between scenarios, the sketch consists of two layers, one for each scenario (e.g. one layer for each wheelbarrow). The predicate of the query in the wheelbarrow problem, `solveDQProblem`, indicates that this is a DQA problem. The first argument is the context containing the facts representing the sketch. The next two arguments are the objects being compared, in the case the wheelbarrows. The next argument is the quantity being compared between these situations, the force applied to the handle of the top wheelbarrow. The last argument is the answer, which is the DQ value indicating the change in the quantity from one situation to the other. The correct answer to

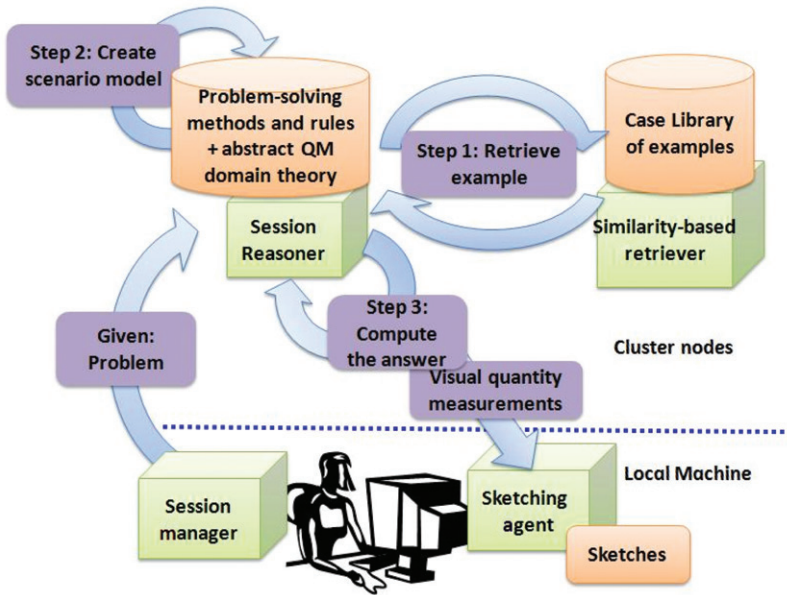


Figure 10. Solving BMCT problems on a companion.

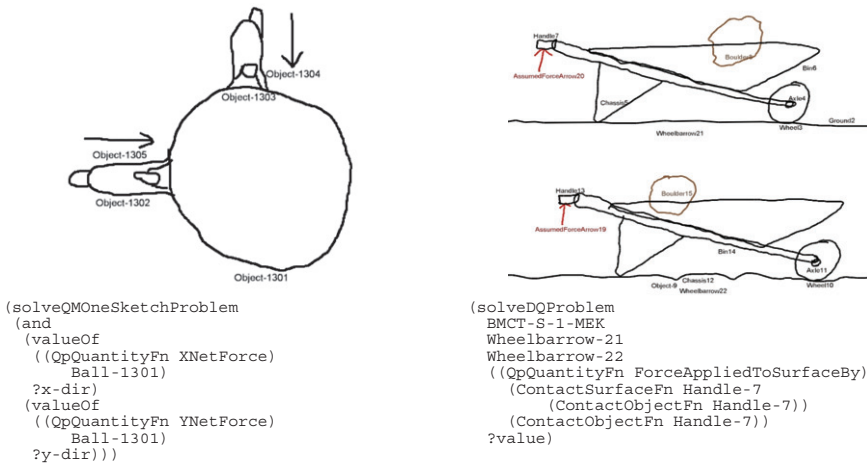


Figure 11. Sample BMCT problems – ‘Which direction will the ball travel?’ (outcome problem) and ‘Which wheelbarrow is easier to carry?’ (DQA problem).

this problem is increasing (i.e. IncreasedDQ) indicating that the force applied to the handle increases from the top wheelbarrow to the bottom wheelbarrow.

### 6.1. Retrieve analogous example

The first step of problem-solving is retrieving an analogous example. The companion does this for each layer in the problem sketch. To retrieve a relevant example, the Retriever uses

MAC/FAC to generate a reminding for the situation depicted by the layer. MAC/FAC determines the most similar example from its case library using the situation with the low level visual properties removed (i.e. visual groupings, glyph orientations and relative sizes) as the probe. For outcome problems, the first retrieval is used. For DQA questions, the retrieval must also contain candidate inferences that causally constrain the goal quantity. This is a useful heuristic, because without these candidate inferences, the companion will not be able to construct a useful scenario model for solving the problem. Should the first retrieval prove unsatisfactory, a second retrieval is performed. If that, too, fails, the low level visual properties are added back into the probe and up to two more retrieval attempts are made.

### **6.2. Perform analogical model formulation**

As described in Section 4, analogical model formulation creates a scenario model for the problem, consisting of causal models, qualitative mechanics abstractions and relationships and information regarding the measurement of visual quantities. Consider for example the ball problem. Analogical model formulation infers that the ball and the two people are instances of the collection `RigidObject`, and the ball `touches-directly` each of the people. These facts allow the companion to formulate a qualitative mechanics model of the problem using its domain theory and a standard model formulation algorithm. This scenario model consists of two model fragments: `XForceTransfer` and `YForceTransfer`. Turning to the wheelbarrow problem, the results of the analogical model formulation for each situation are described in detail in Section 4. For this particular problem, the qualitative proportionality between the force applied at the handle and the distance between the rock and the centre of rotation is a crucial aspect of each resulting scenario model.

### **6.3. Solving outcome problems**

Solving an outcome problem involves standard QR. For example, in the ball problem, the companion calculates the net force, down and to the right, from the consequences of the force transfer model fragments.

### **6.4. Solving differential qualitative analysis problems**

Solving a DQA problem requires the additional step of constructing an analogical reference frame. The companion uses the causal model and analogical reference frame to ascertain the relevant visual properties to measure. These visual properties are measured and their numerical values compared to produce DQ values for the causally independent parameters. These DQ values are then propagated through the causal model to derive a DQ value for the query parameter. These problems can be quite difficult: one problem on the BMCT involves comparisons between aspects of three situations. In the case, the companion sets up three reference frames, one for each pair of situations, and carries out the same analysis for each reference frame in order to derive the correct answer.

#### 6.4.1. *Creating the analogical reference frame*

As described in Section 5, the companion creates an analogical reference frame for the problem. It uses SME to create a mapping between the two scenarios, or the scenario with itself. In the latter case, the mapping is constrained by requiring a correspondence between the aspects of the scenarios being compared by the query. The resulting correspondences indicate how quantities should be compared. Specifically, the DQ value for a parameter refers to a comparison between its value and that of the parameter corresponding to it in the mapping. In the wheelbarrow problem, the companion sets up the analogical reference frame by mapping the top wheelbarrow onto the bottom wheelbarrow.

#### 6.4.2. *Backward chaining through the causal model*

Once the reference frame is set up, DQA proceeds by chaining backward from the sought quantity through the causal model. Non-visual quantities that are either not causally constrained by other parameters or are not known to be different are assumed to be the same across the scenarios, i.e. a DQ value of unchanged. For example, in the wheelbarrow problem in Figure 11, the companion assumes that the rocks apply the same amount of force on the wheelbarrow's bins because force applied is not a visual quantity. DQ values for visual quantities are computed by comparing measurements between the corresponding quantities in the sketch. In the wheelbarrow problem, the companion determines that the force applied on the handle is proportional to the distance from the rock to the wheelbarrow's origin of rotation. Therefore, since the DQ value for distance from the surface of the bin touching the rock to the wheelbarrow's origin of rotation increases, the force applied on the handle also increases from the top wheelbarrow to the bottom wheelbarrow.

#### 6.4.3. *Measuring visual quantities*

The companion uses the sketching agent to measure visual quantities. In this study, there are three types of visual quantities: `DistanceBetweenSurfaces`, `DistanceToOrigin` and example-specific visual quantities defined by a linear annotation. The `DistanceBetweenSurfaces` quantity represents the distance between two surfaces. sKEA reduces each surface to a point by averaging the X and Y coordinates of the surface's endpoints and computes the distance between these points. The `DistanceToOrigin` quantity represents the distance from a surface to the centre of rotation of the object the surface is on. The centre of rotation is determined by two methods, each of which depends upon the results of analogical model formulation. First, the object may participate in the conceptual relationship, `canPivotAround`, with another object. In this case, the surface between these objects would be the centre of rotation. Second, the object may be part of a conceptual glyph group. In this case, if there is another glyph also in the group who is an instance of the collection `Fulcrum`, the centroid of this glyph is the centre of rotation. Example-specific visual quantities created by annotations are measured from the anchor points transferred via candidate inferences. Linear annotations measure the distance between anchor points, while X-coordinate and Y-coordinate annotations measure distance between the anchor points along the appropriate axis.

In the wheelbarrow problem, the companion determines that the distance to the origin of rotation from the surface on the wheelbarrow's bin, defined by the contact with the

rock, is a relevant visual quantity using the causal model. sKEA uses the ink of the bin and rock glyphs to determine the line segment of the bin which represents the surface. sKEA averages the endpoints of this line segment to calculate one end of the distance measurement. The centre of rotation of this wheelbarrow depends upon the qualitative mechanics abstraction *Fulcrum*. Because the wheelbarrow is a group glyph and the axle is an instance of the *Fulcrum* collection, the companion selects the centroid of the axle as the centre of rotation for the wheelbarrow. Next, sKEA computes the distance between these points and provides the companion with the results. The same process occurs for the other wheelbarrow. Recall that not only was the relevance of this quantity established via analogy, its measurement also depended upon the qualitative mechanics abstraction inferred during analogical model formulation.

Next, we evaluate a companion's performance using this algorithm across a range of problems from the BMCT.

## **7. Experiment**

We conducted an experiment to see how well a companion performs on problems from the BMCT using analogical model formulation. We chose a subset of the test because we wanted to minimise domain encoding efforts, to focus better on seeing what could be achieved via analogical model formulation. We selected 13 of the 68 problems on the BMCT, focusing on problems involving net force, revolution rate, stability and smoothness of ride. Since some of these phenomena can be handled by pre-existing qualitative mechanics theories, we test the domain breadth problem posed by the BMCT by including problems in our evaluation set without providing the companion with a corresponding domain theory.

A total of 11 problems involved differential qualitative analyses, six of which involved phenomena not covered by the companion's domain theory. The other two problems were outcome problems. The experiment and analyses below provide evidence concerning three questions:

- (1) Can a companion using analogical model formulation solve BMCT questions?
- (2) How does a companion perform when the number of explanations increases? This is important for assessing how well learning by adding examples scales.
- (3) How well do the retrieval and mapping mechanisms perform? That is, when there are errors, how often are these the cause, as opposed to some other part of the system?

### **7.1. Method**

To model the experiences that analogical model formulation draws upon, we created a list of 18 example situations. A total of 15 examples were intended to be good analogues for specific test questions, with each problem from the BMCT subset having at least one relevant example. The other three examples provided additional distracters. To create the examples, we recruited three graduate students, with varying degrees of familiarity with sKEA, to serve as knowledge enterers (KEs). Each example was described to the KEs by a short English phrase, e.g. 'a tricycle'. They were instructed to draw each example in two dimensions, avoiding perspective projections. They were also instructed to break objects into separate glyphs whenever they were going to be referring to a named part in

describing how that example worked. For example, because a wheelbarrow is lifted at its handle and rotates around its axle, the handle and axle would be separate glyphs. While drawing, they used sKEA's conceptual labelling interface to apply appropriate domain abstractions from a list provided to them. They also used sKEA's visual/conceptual relationship interface to select relevant relationships. KEs were also given a list of physical quantities that were relevant in this subset of the BMCT (i.e. smoothness of ride, revolution rate and stability). When one of these quantities was relevant, they were instructed to include it in their example, and explain what other properties of the system that it depends on, using annotation glyphs as necessary to define physical quantities in terms of visual measurements. To evaluate their progress, they ran a standard qualitative mechanics model formulation system to derive model fragment instances. If there were missing or inappropriate instances of model fragments, they were encouraged to modify their sketch until they were satisfied with the model fragments generated. For example, in a sketch depicting two meshed gears where one of the gears is rotating (as indicated by an annotation), the KE would know that something was wrong if there was no mention of torque transfer in the active model fragment list. Once finished, each example sketch was stored in a case library for that particular KE.

The 13 problems were drawn by a fourth graduate student, an experienced sKEA user. The problem sketches did not include any qualitative mechanics structural abstractions or conceptual relationships. Thus, all the problems required analogical model formulation to arrive at the correct answer. No guessing by the companion was allowed. The 13 problems were presented to the companion in a series of seven trials. In the first three trials, the companion had access to each KE's case library individually. In the next three trials, the companion was given each pair wise combination of case libraries, and in the final trial, the companion had all the examples from the three cases libraries.

While the 13 problems represent a subset of the test, they still cover a broad range of situations. The predicate calculus generated for the sketches of these 13 problems contains 164 entities of 84 different types. These entities are related to each other by 37 unique relations. The problem representations contain on average 182 facts, with the largest and smallest problems having 397 and 40 facts, respectively. Because annotations and conceptual relationships were added to the examples, the example representations are slightly larger, ranging from 52 to 467 facts with an average of 201. They include 100 conceptual types and 212 unique relations.

## 7.2. Results

A summary of the results appears in Table 2. The correct retrieval column lists the number of times the system retrieved one of the appropriate analogues for the problem sketch. All the retrieval results are statistically significant ( $p < 0.001$ ). The correct answer column lists the number of times the system provided the correct answer. In four of the seven trials, the companion produced the correct answer on a statistically significant number of problems ( $p < 0.05$ ). Every problem was solved in at least one of the trials. KE2 had the most experience with sKEA, leading to similar representations to the problems, and KE3 had the least experience, providing some serious variability. Table 2 demonstrates that companions can indeed solve BMCT problems via analogical model formulation: 77%



Table 2. Problem solving results versus case libraries of examples.

Library	Number of correct retrievals (out of 13)	Number of correct answers (out of 13)
KE1	7	6 ( $p < 0.24$ )
KE2	10	10 ( $p < 0.001$ )
KE3	5	2 ( $p < 0.96$ )
KE1 + 2	11	9 ( $p < 0.008$ )
KE1 + 3	9	6 ( $p < 0.24$ )
KE2 + 3	10	10 ( $p < 0.001$ )
KE1 + 2 + 3	12	10 ( $p < 0.001$ )

Table 3. Companion's performance by question type.

Question type (number)	Number of answers produced (%)	Number of correct retrievals (%)	Number of answers correct (%)
BMCT questions (91)	71 (78)	65 (71)	53 (58)
Outcome questions (14)	14 (100)	10 (71)	10 (71)
DQA questions (77)	57 (74)	55 (71)	43 (56)
Net force questions (35)	24 (68)	23 (65)	15 (42)
Other questions (42)	33 (78)	32 (76)	28 (66)

correct, under the best conditions. Furthermore, as the number of available examples grows, the companion's performance improves. Notice that every combination of KEs except for KE2 + KE3 provides an improvement in correct retrievals. This is important because it means greater breadth can be achieved to some degree by increasing the system's experience. A close inspection of the results reveals that in each of the combination trials, example sketches from at least two of the case libraries were used to formulate correct answers. This indicates that the methods have some degree of robustness across examples entered by multiple people.

Table 3 looks at the same data, but broken down by question type, to get a better understanding of the companion's performance. Overall, the companion answered correctly 58% of the time across all problem/memory conditions. However, since we have ruled out errors in the fixed components of the system via by-hand analysis, the difference between the number of answers produced (78%) and correct retrievals (71%) suggests that there are occasional problems in mapping or in using candidate inferences. This was not the case for the outcome questions – if the retrieval was correct, the companion derived the correct result. Recall that outcome questions use whatever analogue is first retrieved. Finding a criterion for testing the analogically derived model would make a difference here. This problem was worse in the DQA problems, despite the use of a relevance heuristic to filter retrievals. In 20 of the DQA questions (26%), the companion was unable to find an example that causally constrained the sought quantity, and thus was unable to produce an answer. Also, when the companion retrieved a

Table 4. DQA results by retrieval number and strategy.

DQA strategy type	Number of uses	Number of correct retrieval	Number of correct answer
1st retrieval: no spatial relations, full case library	33	32	28
2nd retrieval: no spatial relations, case library – 1st retrieval	5	5	3
3rd retrieval: including spatial relations, full case library	11	11	8
4th retrieval: including spatial relations, case library – 3rd retrieval	8	6	4
Total	57	54	43

relevant example, it still missed 12 out of 55 problems (22%). We examine the failures more closely below.

We evaluate analogical model formulation’s performance without a complete domain theory by distinguishing between different types of DQA questions. In net force questions, the companion transferred causal models from examples that were generated from our qualitative mechanics domain theory. The companion performed slightly worse on these problems averaged across all memory conditions. It retrieved a relevant example 65% of the time and answered the problem correctly on 42% of the problems across all memory conditions. The other questions concern the phenomena not covered in the companions domain theory: stability, revolution rate and ride smoothness. The causal models required to solve these problems were defined in an example-specific manner via linear annotations. The companion answered 66% of these problems correctly across all memory conditions, supporting the hypothesis that analogical model formulation is a promising approach to addressing the domain breadth problem posed by the BMCT.

To better evaluate the retrieval heuristic used in DQA problems, Table 4 organises the DQA results based upon which retrieval strategy produced the answer. While the majority of the answers were based upon the first retrieval, 24 answers across all memory conditions required additional retrievals. Of these, the companion retrieved an appropriate analogue 22 times and answered the question correctly 15 times. These results support our hypothesis that our retrieval method is useful in for solving BMCT problems from examples.

### 7.3. Analysis of failures

It is useful to understand why systems fail. We first analyse failures in outcome problems, and then analyse failures in DQA problems.

The four failures on outcome problems occurred because the companion failed to retrieve the correct example from memory. The companion confused a gear rotating inside another gear with two gears rotating side by side, due to annotation glyph placement. This results in incorrect qualitative mechanics abstractions and relationships being assumed in the problem, which in turn leads to the companion constructing an incorrect scenario model. Currently, we do not have a method for evaluating the retrieval on

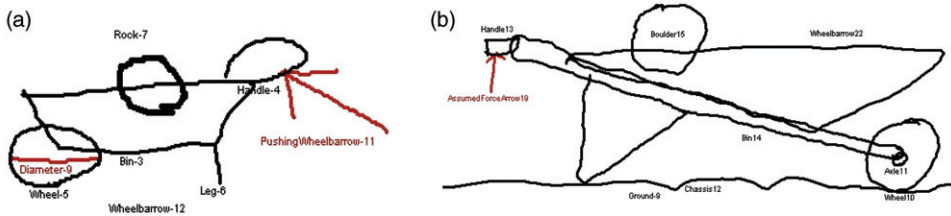


Figure 12. (a) Example wheelbarrow, (b) problem wheelbarrow.

outcome problems. People seem to handle this problem by recognising contradictions in their reasoning. Recognising contradictions is difficult in analogical model formulation because the mapping with the example is assumed to provide the correct structural abstraction and conceptual relationships. We plan to address this in future work by exploring ways of combining reasoning from multiple examples.

The failures on DQA problems occurred during both the retrieval and mapping stages of the algorithm. As noted above, the retrieval must yield a mapping which causally constrains the quantity in question. For 20 of the 77 problem/memory condition pairs, such an example could not be found. Frequently, this was because the KEs sketched the situation at a different level of abstraction than the problem. Even when an example is found that constrains the goal quantity, there are two failure modes for mapping. First, the causal model may include surfaces or objects which do not exist in the problem sketch. Figure 12 illustrates this problem. The problem sketch contains glyphs for the ground and the axle, but the example does not. Furthermore, the chassis in the problem sketch is conceptually labelled as a leg in the example. While SME handles incomplete matches, significant differences in depiction, such as divergence in number of glyphs, can cause mapping failures. In this case, the bin in the example maps to the chassis in the problem. The candidate inferences postulate surfaces and/or glyphs that do not exist (e.g. the surface between the boulder and the chassis), which sKEA is unable to reason about. The companion fails whenever a candidate inference for a necessary causal relationship includes references to either surfaces or glyphs that do not exist.

The second kind of mapping failures in DQA problems are errors in measurements of user defined visual quantities. When measuring the wheel diameter in the problem scenario of Figure 12, if the wheel in the example was mapped to the axle in the problem, then the visual measurement for the wheel diameter would be the width of the axle in the problem. These mapping failures lead the companion to produce incorrect answers.

People seem to have several methods for dealing with such problems. First, they try other examples, going back to memory to find an example that is more productive. A simple version of this is already implemented in our retrieval method. Also, people use *rerepresentation* (Yan, Forbus, and Gentner 2003) to bring the base and target into closer alignment. Knowledge about depiction seems crucial: If two sketches are misaligned, simplifying the more complex one, or postulating new glyphs in the simplified one, seems to be a promising strategy.

As illustrated in Table 4, the companion performed slightly worse on net force DQA problems which relied on causal models generated by its qualitative mechanics domain theory than on problems which relied upon example-specific causal models. An analysis of these causal models indicates differences in size which increased the likelihood

of mapping failures. The net force causal models contained more entities than the example-specific causal models. This is evident in the example wheelbarrow in Figure 12a. The causal model concerning the force applied at the handle references the following entities: Handle-4, (ContactObjectFn Object-4), Rock-7, Bin-3 and Wheel-5. All five of these objects must map appropriately to the problem to create the necessary scenario model. On the other hand, the example-specific explanation for the wheelbarrow's ride smoothness references only two entities: Wheel-5 and Wheelbarrow-12. While the causal models concerning net force DQA problems are more complex, the companion does have access to a qualitative mechanics domain theory. One avenue for future work is to develop a theory which uses the first-principles explanation to verify and repair these analogically inferred causal models.

## 8. Related work

Another example of psychometric AI is Project HALO (Barker et al. 2004), which explored how well the state of the art in knowledge-based systems could be used to create systems that solve AP Chemistry problems. Like our work, HALO focused on a subset of the test, basing their problems on several pages of textbook knowledge. The approach used by all three teams was to encode general-purpose knowledge by hand, using teams of domain experts and AI experts. The everyday breadth problem was handled by hand-encoded rules, which is less general than our analogical model formulation approach. The HALO systems had to handle quantitative knowledge, including equation solving, whereas the BMCT requires us to do spatial reasoning.

The compositional modelling methodology (Falkenhainer and Forbus 1991) developed in the QR community has led to efficient algorithms for model formulation using first-principles reasoning from domain theories (Nayak 1994; Rickel and Porter 1994). These algorithms assume that the input descriptions are already in an abstract structural vocabulary, and do not handle the problem of mapping from everyday entities to abstract vocabularies, as we do. The use of experience in model formulation was first proposed by Falkenhainer (1992) for improving compositional modelling by choosing appropriate perspectives and levels of detail. Falkenhainer's work, like other compositional modelling efforts, still started with a narrow vocabulary for describing structural descriptions. Our study, by contrast, exploits the breadth of the ResearchCyc knowledge base to provide a vastly broader range of input possibilities, and analogical model formulation to determine which abstractions should apply to them.

Some researchers use high-resolution inputs or CAD systems to produce structural descriptions containing metric information which can also be exploited by the system (Joscowicz and Sacks 1993). Aside from annotations and using quantitative information to compute low-level qualitative visual properties, we do not exploit metric information at all. Stahovich, Davis, and Shrobe (2000) used sketched input, but allowed only a handful of abstract types to be drawn. Archytis (Yaner and Goel 2008) uses a method called compositional analogy to construct structural models from unlabelled two dimensional line drawings. Their system recognises shapes based upon similarities between the lines and intersections in the problem and the labelled example. These shapes then guide the transfer of component types and connections from the example to the problem. Our system differs in a number of important ways. First, they use vector-graphics inputs, which are easier to process than the hand-drawn sketches we use with sKEA. On the other hand, we require

users to segment their glyphs. While both systems rely to some degree on hand-labelled conceptual information, we use automatic model formulation during example creation to automatically add information, thereby reducing tailorability. Archytis does not address retrieval, and their analogies focus only on similarities in depiction, unlike our use of MAC/FAC for retrieval and the use of both visual and conceptual information in mapping.

In analogical problem solvers and case-based reasoning (CBR) systems, inferences are made about problems based upon previous cases. Most of today's CBR systems are based on feature-vectors, and hence lack the representational capacity to handle everyday physics problems. In CBR systems which use relational representations (as we do), typically a heavy emphasis is placed on adaptation (Kolodner 1993). This frequently requires domain specific heuristics. In analogical model formulation, the adaptation is completely handled via structure mapping. Analogical problem solvers (VeloSo and Carbonell 1993; VanLehn 1998; Melis and Whittle 1999; Ouyang and Forbus 2006) typically solve new problems by transferring plans from previous problem-solving episodes. Thus analogy is used as a means of guiding the problem solver, but it could, with more effort, solve the problems without analogues. In contrast, companions use analogical model formulation to infer crucial pieces of domain knowledge, i.e. qualitative mechanics abstractions and relationships, definitions for visual quantities and causal relationships. Without these inferences, the companion would not be able to solve any of these problems.

## **9. General discussion**

We have argued that QR combined with analogical processing is a promising avenue for addressing the problems of domain breadth and everyday breadth that must be solved for human-like reasoning. The companion's performance on a subset of the BMCT provides important evidence for this claim. Analogical model formulation enables a companion to build scenario models over a broad range of input descriptions, even with an incomplete domain theory. Sketch annotations provide a means for defining visual quantities in examples, which can be used in example-specific causal explanations and applied via analogy to new situations. This extends the companion's reach by enabling it to reason about phenomena for which it does not have a corresponding domain theory. Analogical reference frames extend the scope of comparative analysis to include the types of problems found on the BMCT.

An important benchmark for psychometric AI is having the same system perform well on a spectrum of intelligence tests (Bringsjord and Schimanski 2003). In this spirit, we have also successfully tested the companions architecture with mechanics problems similar to those on the AP Physics exam (Klenk and Forbus 2007), using the same strategy of learning by accumulation of examples and analogical model formulation. The success of companions on these two very different kinds of problems provides support for the generality of these ideas.

There is much future work ahead: after all, the BMCT includes statics, dynamics, heat, electricity and even orbital mechanics. There is no QR system that comes close to handling the breadth of phenomena covered in the exam, let alone the breadth of everyday concepts involved. A system that can learn to perform well on this exam would be an important milestone for AI. Our long-term goal is to enable a companion to solve the entire test, but

with as little hand coding as possible. Our use of examples to describe new phenomena to the system is a first step in that direction. One line of investigation is to move beyond learning via accumulating examples, using the SEQL model (Kuehne, Forbus, Gentner, and Quinn 2000) to learn generalisations from examples. Another is to explore the debugging of reasoning. During problem solving, we need to incorporate strategies for verifying and repairing analogical inferences made from explanations generated by first-principles reasoning. After problem solving, we have to be able to give the companion feedback and advice. For example, when a companion gets a problem wrong, it could figure out how to change its knowledge so that it does better in the future. We suspect that model-based diagnosis techniques (e.g. de Koning, Bredeweg, Breuker, and Wielinga 2000) would be useful for this. Recall that for more than half of the incorrect answers, the companion knew in some sense that it could not provide a good answer. This is an excellent opportunity to explore interactive learning during problem solving. For example, given an incorrect mapping or retrieval, providing corrective feedback could give a companion evidence about how to reorganise its encoding strategies and/or re-representation methods. We are developing an interactive interface for this, using simplified English for input, which we hope will also help expand the range of materials we can use in testing them and helping them learn.

### Acknowledgements

This study was supported by DARPA IPTO. The authors would like to thank Kate Lockwood and Patricia Dyke for their sketches and Tom Hinrichs and Jeff Usher for their work on the companion architecture.

### Notes

1. For the security of the test, we cannot provide a full list of the problems used in this evaluation.
2. The majority of the content in our KB is drawn from ResearchCyc ([www.research.cyc.com](http://www.research.cyc.com)), plus our own material on QR and analogy. The conventions for Cyc-style knowledge bases (Lenat and Guha 1989) are documented on that website.
3. The compound form shown in Figure 3 is translated into a set of backward chaining rules for use with our reasoning engine.
4. Torque equilibrium also requires that the opposing torques be equal. Our system currently assumes this by default.

### References

- Barker, K., Chaudhri, V., Chaw, Yi, S., Clark, P., Fan, J., Israel, D., Mishra, S., Porter, B., Romero, P., Tecuci, D., and Yeh, P. (2004), 'A Question Answering System for AP Chemistry: Assessing KR Technologies', in *Proceedings of the 9th International Conference on Knowledge Representation and Reasoning*, pp. 545–553.
- Bennett, G.K. (1969), *Bennett Mechanical Comprehension Test*, San Antonio, TX: The Psychological Corporation.
- Bringsjord, S., and Schimanski, B. (2003), 'What is Artificial Intelligence? Psychometric AI as an Answer', in *Proceedings of IJCAI03*, pp. 887–893.
- Cohn, A. (1996), 'Calculi for Qualitative Spatial Reasoning', in *Artificial Intelligence and Symbolic Mathematical Computation*, LNCS 1138, eds. J. Calmet, J.A. Campbell and J. Pfalzgraf, Berlin: Springer Verlag, pp. 124–143.



- de Koning, K., Bredeweg, B., Breuker, J., and Wielinga, B. (2000), 'Model-Based Reasoning about Learner Behavior', *Artificial Intelligence*, 117, 173–229.
- Falkenhainer, B. (1988), 'Learning from Physical Analogies', Unpublished Ph.D. thesis, University of Illinois at Urbana-Champaign.
- Falkenhainer, B. (1992), 'Modeling Without Amnesia: Making Experienced-Sanctioned Approximation', in *The Sixth International Workshop on Qualitative Reasoning in Physical Systems*, pp. 44–55, Edinburgh, Scotland, Heriot-Watt University.
- Falkenhainer, B., and Forbus, K. (1991), 'Compositional Modeling: Finding the Right Model for the Job', *Artificial Intelligence*, 51, 95–143.
- Falkenhainer, B., Forbus, K., and Gentner, D. (1986), 'The Structure-Mapping Engine', in *Proceedings of the Fifth National Conference on Artificial Intelligence*, pp. 272–277.
- Forbus, K. (1984), 'Qualitative Process Theory', *Artificial Intelligence*, 24, 85–168.
- Forbus, K. (2001), 'Exploring Analogy in the Large', in *Analogy: Perspectives from Cognitive Science*, eds. D. Gentner, K. Holyoak and B. Kokinov, Cambridge: MIT Press.
- Forbus, K., and de Kleer, J. (1993), *Building Problem Solvers*, Cambridge, MA: MIT Press.
- Forbus, K., Ferguson, R., and Gentner, D. (1994), 'Incremental Structure-Mapping', in *Proceedings of the Cognitive Science Society*, pp. 613–618.
- Forbus, K. and Gentner, D. (1997), 'Qualitative Mental Models: Simulations or Memories?' in *Proceedings of QR97*, Cortona, Italy.
- Forbus, K., Gentner, D., and Law, K. (1994), 'MAC/FAC: A Model of Similarity-Based Retrieval', *Cognitive Science*, 19, 141–205.
- Forbus, K. and Hinrichs, T. (2004), 'Companion Cognitive Systems: A Step Towards Human-Level AI', in *AAAI Fall Symposium on Achieving Human-level Intelligence through Integrated Systems and Research*, October, Washington, DC.
- Forbus, K., Lockwood, K., Klenk, M., Tomai, E. and Usher, J. (2004), 'Open-Domain Sketch Understanding: The nuSketch Approach', in *Proceedings of the AAAI Fall Symposium on Making Pen-based Interaction Intelligent and Natural*, October, Washington, DC.
- Forbus, K., Tomai, E., and Usher, J. (2003), 'Qualitative Spatial Reasoning for Visual Groupings in Sketches', in *Proceedings of the 17th International Workshop on Qualitative Reasoning*, pp. 131–138, Brasilia, Brazil.
- Forbus, K. and Usher, J. (2002), 'Sketching for Knowledge Capture: A Progress Report', *IUI'02*, San Francisco.
- Forbus, K., Usher, J., and Tomai, E. (2005), 'Analogical Learning of Visual/Conceptual Relationships in Sketches', in *Proceedings of AAAI-05*, Pittsburgh, PA.
- Gentner, D. (1983), 'Structure-Mapping: A Theoretical Framework for Analogy', *Cognitive Science*, 7, 155–170.
- Joskowicz, L. and Sacks, E. (1993), 'Configuration Space Computation for Mechanism Design', in *IEEE-94*, pp. 1080–1087.
- Kim, H. (1993), 'Qualitative Reasoning About Fluids and Mechanics', unpublished Ph.D. dissertation and ILS Technical Report, Northwestern University, Evanston, IL.
- Klenk, M. and Forbus, K. (2007), 'Measuring the Level of Transfer Learning by an AP Physics Problem-Solver', in *Proceedings of AAAI-07: Twenty-Second Conference on Artificial Intelligence*, Vancouver, BC, pp. 913–918.
- Kolodner, J.L. (1993), *Case-Based Reasoning*, San Mateo, CA: Morgan Kaufmann Publishers.
- Kuehne, S., Forbus, K., Gentner, D. and Quinn, B. (2000), 'SEQL: Category Learning as Progressive Abstraction Using Structure Mapping', in *Proceedings of Cognitive Science*, pp. 750–775.
- Labrou, Y. and Finin, T. (1997). 'A Proposal for a New KQML Specification', Technical Report TR CS-97-03, University of Maryland, Baltimore County.
- Lenat, D.B., and Guha, R.V. (1989), *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*, Reading, MA: Addison-Wesley.

- Melis, E., and Whittle, J. (1999), ‘Analogy in Inductive Theorem Proving’, *Journal of Automated Reasoning*, 22, 117–147.
- Nayak, P. (1994), ‘Causal Approximations’, *Artificial Intelligence*, 70, 277–334.
- Nielsen, P. (1988), ‘A Qualitative Approach to Rigid Body Mechanics’, Unpublished Ph.D. dissertation, University of Illinois at Urbana-Champaign.
- Ouyang, T. and Forbus, K. (2006), ‘Strategy Variations in Analogical Problem-Solving’, in *Proceedings of AAAI-06*.
- Rickel J. and Porter, B. (1994), ‘Automated Modeling for Answering Prediction Questions: Selecting the Time Scale and System Boundary’, *AAAI-94*, pp. 1191–1198, Cambridge, MA: AAIT/MIT Press.
- Stahovich, TF., Davis, R., and Shrobe, H. (2000), ‘Qualitative Rigid Body Mechanics’, *Artificial Intelligence*, 119, 19–60.
- VanLehn, K. (1998), ‘Analogy Events: How Examples are Used During Problem Solving’, *Cognitive Science*, 22, 347–388.
- Veloso, M., and Carbonell, J. (1993), ‘Derivational Analogy in PRODIGY: Automating case acquisition, storage, and utilization’, *Machine Learning*, 10, 249–278.
- Weld, D. (1988), ‘Comparative Analysis’, *Artificial Intelligence*, 36, 333–374.
- Yan, J., Forbus, K., and Gentner, D. (2003), ‘A Theory of Rerepresentation in Analogical Matching’, in *Proceedings of CogSci03*.
- Yaner, P., and Goel, A. (2008), ‘From Design Drawings to Structural Models by Compositional Analogy’, *International Journal of AI in Engineering Design, Analysis and Manufacturing, Special Issue on Multimodal Design*, 22, 117–128.

### Appendix 1. Additional qualitative mechanics model fragments

```
(defModelFragment YForceTransfer
:participants ((TheObject :type RigidOb)
(TheForcer :type RigidOb)
(TheSurface1 :type Surface
:constraints ((hasSurface TheObject TheSurface1)))
(TheSurface2 :type Surface
:constraints ((hasSurface TheForcer2 TheSurface2)
(surfaceContact TheSurface1 TheSurface2)))
(TheYDir1 :type Sense))
:conditions ((yForceApplied TheSurface1 TheXDir1 TheForcer))
:consequences ((c+ ((QpQuantityFn YNetForce) TheObject)
((QpQuantityFn YForceAt) TheSurface1)
YForceTransfer)))

(defModelFragment TorqueTransfer
:participants ((TheObject :type RigidOb)
(TheForcer :type RigidOb)
(TheSurface1 :type Surface
:constraints ((hasSurface TheObject TheSurface1)))
(TheSurface2 :type Surface
:constraints ((hasSurface TheForcer2 TheSurface2)
(surfaceContact TheSurface1 TheSurface2)))
(TheRotDir1 :type RotDirection))
:conditions ((torqueApplied TheSurface1 TheRotDir1 TheForcer))
:consequences ((c+ ((QpQuantityFn NetTorque) TheObject)
```

```

      ((QpQuantityFn TorqueAt) TheSurface1)
      TorqueTransfer)))
(defModelFragment ForceDistribution
 :participants ((TheObject :type RigidOb)
 (TheForcer :type RigidOb)
 (TheSurface1 :type Surface
 :constraints ((hasSurface TheObject TheSurface1)))
 (TheSurface2 :type Surface
 :constraints ((hasSurface TheObject TheSurface2)
 (notDirectlyForceApplied TheSurface2 TheDir1 TheForcer)))
 (TheForcerSurface :type Surface
 :constraints ((hasSurface TheForcer TheForcerSurface)
 (surfaceContact TheSurface1 TheSurfaceOfForcer)))
 (TheDir1 :type 2DQVector))
 :conditions ((forceApplied TheSurface1 TheDir1 TheForcer))
 :consequences ((qprop- ((QpQuantityFn ForceAppliedToSurfaceBy)
 TheSurface2 TheForcer)
 ((QpQuantityFn DistanceBetweenSurfaces) TheSur-
 face2 TheSurface1))))

```

Copyright of Journal of Experimental & Theoretical Artificial Intelligence is the property of Taylor & Francis Ltd and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.