INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15 – 18, 2005

REPRESENTING CK THEORY WITH AN ACTION LOGIC

Filippo A. Salustri

Abstract

CK theory is an interesting and unique theory of design. This paper introduces ALX3d, a formal version of CK based on the action logic ALX3, which is able to represent aspects of the actions, preferences, beliefs, and knowledge of collaborating, imperfect agents (such as human designers). It is shown that all the basic notions of CK can be rendered in the logic of ALX3d with only one relatively minor change in how the CK terms *concept* and *knowledge* are defined and related. Beyond this, ALX3d provides a richer framework for describing design activities in formal terms, including alternation between synthesis and analysis tasks, and the definition of goals that trade off design requirements against one another. A case study of CK from is used to show how ALX3d can also be used to describe some "real-world" situations. The advantages of ALX3d are that they recast CK in a form more readily understood by those accustomed to expert, knowledge-based, and formal systems; provide a "scientific" vehicle for reasoning about the design activities it can describe; and define a possible basis for the development of new, computer-based designers' aids.

Keywords: CK theory, design theory, formal system, action logic

1 Introduction

CK theory [1,2] presents an interesting and unique theory of design, but the available literature does not cast CK in a sound logic. In this paper, the author will demonstrate that CK can be captured by an *action logic*, ALX3. CK must be adapted to follow the conventions of formal systems, but the spirit and benefits of CK are preserved. Some extensions to CK are also presented and related to other work by the author [3]. Having a formal representation of CK yields new benefits that will be described.

Action logics are formal systems that address the activities undertaken by reasoning agents. ALX3 [4] is especially well suited because it is the only sound and complete action logic of which the author is aware that assumes the agents (i.e. human designers) exhibit *bounded rationality* – they are imperfect reasoners having imperfect/incomplete knowledge (per Simon [5]). CK also assumes bounded rationality.

The author bases this work on [1]. Each element of CK is translated in turn into ALX3. Then, some extensions drawn are added from the author's own work. The resulting new theory is called ALX3d (ALX3 for design).

2 Overview of ALX3

ALX3 is a sound and complete 1st order action logic that incorporates knowledge, belief, preference, and action operators to represent the activities of multiple agents working with bounded rationality (per Simon [5]). ALX3 is completely documented in [4]. It assumes the usual apparatus of 1st order logic: constants, variables, functions, and relations, conjunction

($\wedge$), disjunction ($\vee$), negation ($\neg$), material implication ($\Rightarrow$), and universal ($\forall$) and existential ($\exists$) quantification over variables. We also use the notation $x \Leftrightarrow_{def} y$ to indicate that *x* is defined by *y*.

In action logics, an *agent* is an entity that takes actions to achieve certain *states*. An agent *a* can know ($\mathbf{K}_a\psi$) or not know ($\neg\mathbf{K}_a\psi$) a proposition ψ; the agent may also believe ($\mathbf{B}_a\psi$) or not believe ($\neg\mathbf{B}_a\psi$) a proposition ψ. ALX3 defines knowledge typically for formal systems as *true, justified belief*. Within the system, knowledge and belief are treated as two separate operators related by a definition of the former with respect to the latter. Other definitions of knowledge are possible without necessarily affecting the soundness of the logic. In the language of ALX3, $\mathbf{K}_a\psi \Leftrightarrow_{def} \mathbf{B}_a\psi \wedge \psi$, but $\neg(\mathbf{B}_a\psi \Rightarrow \mathbf{K}_a\psi)$. That is, for a proposition ψ, knowledge is equivalent to true, justified belief, but it is not the case that belief implies knowledge. Consider the statement "It rained in Tokyo yesterday." The statement is either true or false, whether or not an agent knows it. The agent may believe the statement is true, but will not *know* until the agent takes appropriate actions to verify the belief.

One might question our definition of knowledge with regards to determining truth. A belief is true if it is true in an absolute sense, which means technically that all we have are beliefs. We would have to have access to some omniscient agent to guarantee that true things are in fact true. There are various ways out of this conundrum. The author prefers to assume that truth is determined relative to a context. So long as the context does not substantively change, then the truth of a statement is "guaranteed". This is a typical scientific approach: until incontrovertible evidence is found that denies a statement assumed to be true, then accepting the truth of the statement is considered reasonable. This has been a working principle in science for centuries and appears quite robust.

The other important difference between knowledge and belief is that knowledge is not "forgotten" – once you have it, you cannot lose it (losing knowledge is not the same as forgetting it). Beliefs, on the other hand, can be *retracted* if a validation activity proves the belief incorrect. This makes beliefs similar to *assumptions* – statements that we take to be true for the sake of achieving some goal until they are demonstrated to be false.

ALX3 also assumes a *many-worlds interpretation* based on action logic semantics. This means that the current state (or "world") is a collection of propositions, and that alternative (or future, or past, or just other) states are accessible from the current state through *actions* that cause propositions to be added or withdrawn. Thus, actions allow one to represent how states of knowledge and belief, and the propositions they include, change. Actions are written as $\langle a\rangle\psi$, where *a* is the action and ψ is a proposition that is true in any state that can occur because of executing the action. In other words, $\langle a\rangle\psi$ means that executing action *a* will make ψ true.

Associated with actions is the notion of *accessibility*. A state *t* is accessible from another state *s* if there exists an action or a sequence of actions that can be described by propositions that are true at *t*, *s*, and all intermediate states. There are two accessibility relations in ALX3. *Direct accessibility* (**DA**) is defined as $\mathbf{DA}_i\psi \Leftrightarrow_{def} \exists a\langle a_i\rangle\psi$; that is, agent *i* can reach a state where ψ holds directly from the current state by executing action *a*. *General accessibility* (**A**) is defined as $\mathbf{A}_i\psi \Leftrightarrow def\ \mathbf{DA}_i\psi \vee (\mathbf{DA}_i\phi \wedge (\phi\therefore\psi))$; that is, agent *i* can reach a state where ψ holds either directly through one action or indirectly through a sequence of actions (the causation operator $\therefore$ is explained below).

A key feature of ALX3 is a special implication operator to describe causal relations. Here we write it $\phi\therefore\psi$; that is, in all states that are *closest* to current state and where ϕ holds, ψ also holds. A *closest state* is one that (a) is accessible from current state (i.e. has an action allowing an agent to move to it from the current state) and (b) has the smallest possible changes from the current state. The semantics of ALX3 [4] provides a complete formal description of this operator, but that is beyond the scope of this paper. We note that the ALX3 causal operator is broader than the usual "scientific" sense of "cause and effect." In ALX3, cause can arise from simple preference (see below – i.e. something is caused because an agent prefers it to an alternative). It can also capture the relation between dependent and independent variables – i.e. the values of a set of independent variables cause the values of a set of dependent variables. As such, a ALX3 causal relation can also be regarded as an *explanation*.

The sequence of actions that occurs between some initial and final states constitutes a *process*. ALX3 agents can remember this sequence thereby having knowledge of the *history* of what they have done to attain some goal.

Finally, agents may sometimes choose to *prefer* one state to another for no reason that one can explain at that moment. ALX3 supports this with a binary *preference operator*. We write this as $\psi\mathbf{P}_a\phi$; that is, agent *a* prefers closest states where ψ is true to closest states where ϕ is true. Preference is a useful concept because it decouples identifying what agents prefer, which one can represent with simple assertions, from the rationale for that preference, which can be far more difficult to capture.

3 Applying ALX3 to design: ALX3d

The fundamental departure in ALX3d from CK is the notion and application of *logical status*. In CK, only *knowledge* has logical status. In CK, *concepts* are not knowledge and have no logical status.

In ALX3d, on the other hand, we distinguish between the logical status of a proposition and whether an agent knows that logical status. That is, we use *belief* (a proposition that has a logical status not known to an agent) as the equivalent notion. We therefore make the following equivalences. CK's *knowledge space* (K-space) contains all propositions "that have logical status" [1]; we interpret this as saying that K-space contains all *known* propositions ψ (i.e. $\mathbf{K}_a\psi$). Similarly, CK's *concept space* (C-space) contains all propositions "that have no logical status" [1]; we interpret this as saying that C-space contains all *believed* propositions ϕ (i.e. $\mathbf{B}_a\phi$).

In CK as in ALX3d, a design is complete when its description is entirely knowledge (no beliefs left), $\mathbf{K}_a D$ (where D stands for the design being developed). We know the design is sufficient because we know it satisfies the requirements (R); in other words, we know the requirements imply at least one design,

$$\mathbf{K}_a(R \Rightarrow D). \tag{1}$$

Initially, in ALX3d, we know some of the requirements, $\mathbf{K}_a R_i$, and may have some beliefs about the design, $\mathbf{B}_a D_i$. We would also believe that there exists a design that satisfies the (final) requirements, $\mathbf{B}_a(R \Rightarrow D)$. We would also believe that we could possibly reach that state, $\mathbf{B}_a\mathbf{A}_a(R \Rightarrow D)$. Both these beliefs are necessary – else why are we even trying to execute the design? We summarise this by saying that the goal of designing is to move from a state of belief to one of knowledge. This is consistent with CK, and we write this as:

$$\mathbf{K}_a R_i \wedge \mathbf{B}_a(D_i \wedge R \Rightarrow D \wedge \mathbf{A}_a(R \Rightarrow D)) \therefore \mathbf{K}_a(R \Rightarrow D \wedge D). \tag{2}$$

In CK, K-space and C-space include propositions that capture design information. In ALX3d we impose a little more detail. The requirements *R* of a design problem is a conjunction of individual requirement propositions, $R \Leftrightarrow_{def} \wedge^{i} r_i$. *R* can be a conjunction of different *r*'s at each different state. Furthermore, a design *D* is a conjunction of individual design propositions, $D \Leftrightarrow_{def} \wedge^{i} d_i$. *D* can be a conjunction of different *d*'s at each different state.

We identify an appropriate design by going from $\mathbf{B}_a(R \Rightarrow D)$ to $\mathbf{K}_a(R \Rightarrow D)$. This is done with a *validation action* where in *D* is evaluated with respect to *R*. (This is a C→K operator in CK.) Once we know $R \Rightarrow D$, then $\mathbf{K}_a(R \Rightarrow D) \Rightarrow \mathbf{K}_a D$ and we know the design too. That is, we know the design is "right" because we have validated it against *R*.

We use the causal implication operator in ALX3 to define the following:

$$\mathbf{K}_a R \wedge \mathbf{B}_a D \therefore \mathbf{K}_a(R \Rightarrow D) \vee \mathbf{K}_a(\neg(R \Rightarrow D)) \vee \neg \mathbf{K}_a(R \Rightarrow D). \quad (3)$$

This says that in all closest states to the current one, if we know the requirements and believe to have a valid design, then one of three conditions holds.

1. $\mathbf{K}_a(R \Rightarrow D)$ – The validation was successful and we now know that *D* satisfies *R*. We have found a solution.
2. $\mathbf{K}_a(\neg(R \Rightarrow D))$ – The validation failed; *D* does not satisfy *R*. *D* and/or *R* will have to be changed to continue.
3. $\neg \mathbf{K}_a(R \Rightarrow D)$ – The validation could not be performed. This would be the case if either *R* or *D* were lacking in sufficient detail.

We will examine these three conditions in further detail below.

CK's definition #1 of "design" [1] can now be written as: design is the process of expanding *R* and *D* to go from $\mathbf{B}_a(R \Rightarrow D)$ to $\mathbf{K}_a(R \Rightarrow D)$.

The CK notion of *K-relativity* is consistent with ALX3. If there is no K-space (even if the K-space is empty), nothing can be done. $R \Rightarrow D$ is evaluated with respect to K-space, which in ALX3d is simply the collection of propositions known to agents. There also exist *context logics* [6] that can further formalize the notion of relativity of knowledge propositions, but this is beyond the scope of this paper.

In CK, "the formulation of the requirements is a first concept formulation which is expanded by the designer in a second concept that is called the proposal." There is a problem, however, if we accept requirements as a *concept* structure. We must *know* at least some of the requirements, i.e. know the logical status, because in CK, one can only reason about things with defined logical status. Other requirements may begin as beliefs (in C-space) and migrate through design actions to K-space, but we cannot create the design without actually *knowing* (some of) what the product must do (the requirements). In ALX3d, we can reason about *beliefs* because they are assumed to have a "temporary" logical status. This means that ALX3d is a richer representation than CK, without violating the intention of CK.

Since CK concepts have no logical status in K-space, and since one needs logical status to choose one element from a set, the Axiom of Choice is not used in CK. This is unwarranted in ALX3d. Beliefs are assumptions regarding the logical status of propositions. The collection of beliefs (the *belief structure*) allows provisional reasoning. A belief can be retracted, which would then require validating all inferences that include the retracted belief. Nonetheless, reasoning is possible. Therefore, we do not need to exclude the Axiom of Choice; indeed, at this time, the author finds no conclusive argument either for or against the Axiom of Choice.

The importance of reasoning about beliefs is an important element of ALX3d that CK does not provide. What designer would pursue a concept he did not *believe* would be fruitful? What company would pursue a product development project that its members did not *believe* would be successful? Designers cannot wait for the certainty of *knowledge* (beliefs that are verified as true), so they must make assumptions to proceed, and backtrack if those assumptions turn out to be wrong. The belief system support in ALX3d lets us do this.

In CK, the logical status of concepts is changed by adding or subtracting properties. In logic, we represent properties with propositions. For example, *weight(motor, 5kg)* asserts the *motor* has the property of *weight* with value *5kg*. Description logics [7] provide further depth to the logical description of properties and how they can be used to develop ontologies (formal descriptions of bodies of knowledge), but this is beyond the scope of this paper.

Therefore, in ALX3d, we add and subtract propositions that ascribe properties to R and D, such that subsequent application of validation actions turn beliefs into knowledge. From this, all four kinds of CK operators (C→C, C→K, K→C, and K→K) translate easily to ALX3d. This is explained below.

Consider the example in [1] about bicycles with pedals and "effective wings." "Bicycles with Pedals" (denoted by the predicate *bp*) leads to a ALX3d belief $\mathbf{B}_a(\exists x\ bp(x))$, while "Bicycles with Effective Wings" (denoted by the predicate *bew*) leads to $\mathbf{B}_a(\exists x\ bew(x))$. "Bicycles with pedals and effective wings" leads to $\mathbf{B}_a(\exists x[bp(x) \wedge bew(x)])$. The real question is not whether such a design is possible but rather whether R⇒x; that is, does there exist a situation wherein a bicycle with pedals and effective wings is appropriate. If there is no such situation, then even considering the issue is vacuous.

The answer depends on what is *known* (the content of CK's K-space). For example, in a dome with an atmosphere on the Moon or some other very low gravity setting, *bew(x)* might be perfectly reasonable. The reason why *bew(x)* seems silly is because of the *situation* (context) we assume in the absence of specific knowledge of ⇒. Context logics [6] and work on situated design [8] might also help here, but again this is beyond the scope of the current paper.

Let us now consider some of the CK operators in more detail.

In CK, K→C operators add or subtract properties written as propositions in K-space to or from concepts, creating *disjunctions* in C-space [1]. This corresponds to the design stage of generating alternatives. These operators expand C-space with elements coming from K-space. The disjunction arises from considering that adding a property partitions the set of all concepts into those that satisfy the property and those that do not. Furthermore, C→C operators expand or "flesh out" a concept by adding other propositions without logical status.

In ALX3d, we can add a new design proposition *d'* to *D*:

$$\mathbf{B}_a D' \therefore \mathbf{B}_a[D \wedge d'], \text{ or } \mathbf{B}_a D'' \therefore \mathbf{B}_a[D \wedge \neg d']. \qquad (4)$$

There is no real distinction in ALX3d between what CK calls K→C and C→C operators, because in ALX3d, beliefs have provisional logical status that allows whatever reasoning "engine" works on knowledge to work on beliefs as well. The actions associated with these state transitions are actions by which a designer proposes new aspects of a design. This partitions states into those where *d'* and those where *¬d'*. This leads to a possible reasoning process as follows.

1. An initial state is assumed of $\mathbf{K}_a R \wedge \mathbf{B}_a D$.

2. The agent finds that validation cannot be done: $\mathbf{K}_a R \wedge \mathbf{B}_a D \therefore \neg\mathbf{K}_a(R \Rightarrow D)$.
3. The agent expands *D* with a new proposition *d'*: $\mathbf{K}_a R \wedge \mathbf{B}_a D \therefore \mathbf{K}_a R \wedge \mathbf{B}_a[D \wedge d']$.
4. Now, validation fails: $\mathbf{K}_a R \wedge \mathbf{B}_a[D \wedge d'] \therefore \mathbf{K}_a(\neg(R \Rightarrow [D \wedge d']))$. Since the previous validation could not be done, but including *d'* causes validation to fail, the only alternative left is *¬d'*.
5. The error is corrected: $\mathbf{K}_a(\neg(R \Rightarrow [D \wedge d'])) \therefore \mathbf{K}_a R \wedge \mathbf{B}_a[D \wedge \neg d']$.

If validation of $\mathbf{K}_a R \wedge \mathbf{B}_a[D \wedge \neg d']$ cannot be carried out, we know that $D \wedge \neg d'$ is not a sufficient solution and that more expansion must be done to the design (and possibly the requirements).

If the validation in step 4 yielded $\neg\mathbf{K}_a(R \Rightarrow [D \wedge d'])$, we would not be able to choose between *d'* or *¬d'* because neither led to a completion of the design process. Alternative courses of action here could include (a) seeking a different validation action – since it might be the validation action itself that cannot operate on the available information in *R* and $D \wedge \neg d'$, (b) continuing to pursue both $D \wedge d'$ and $D \wedge \neg d'$ as design alternatives until validation does give a distinct answer, (c) changing *R* and trying to validate again, or (d) arbitrarily choosing *d'* or *¬d'* by means of the designer's preferences (e.g. $d' \mathbf{P}_a \neg d'$). Changing *R* is done just as changing *D*, by adding *r'* or *¬r'* to give *R'* or *R''* respectively.

If validation of $\mathbf{K}_a R \wedge \mathbf{B}_a[D \wedge \neg d']$ *fails* in step 5 above, then neither *d'* nor *¬d'* is a suitable solution. This means that there is an error in *R*, since one of *d'* or *¬d'* *must* be true. In this case, one must use some sort of strategy to backtrack to earlier states until one finds a state in the history of changes to *R* where either *d'* or *¬d'* *does* hold. Details of such strategies constitute future work; here it is sufficient to recognize that such representations are possible in ALX3d.

CK's *restricting partitions* correspond to the addition of a proposition *known* by the agent to a design ($D' \Leftrightarrow_{def} D \wedge \mathbf{K}_a p$). Similarly, CK's expanding partitions correspond to the addition of a proposition that the agent *believes* to a design ($D' \Leftrightarrow_{def} D \wedge \mathbf{B}_a p$).

In CK, the current state can be "backtracked" by returning to a previous state, but the theory itself does not formally describe this (for example, by some appropriate operator). In ALX3d, however, we can use belief retraction to formalize backtracking directly. We can write this as $D' \Leftrightarrow_{def} D \backslash d'$ – that is, *D'* is like *D* but without *d'*. Backtracking in this manner does not apply to knowledge, because it is a principle of action logics that knowledge cannot become unknown once it is known. (Note: this is *not* the same as backtracking in logic programming languages like Prolog.)

Let us now consider C→K operators, which turn a concept into knowledge in CK. In ALX3d, these are validation actions. Once a (design) concept becomes knowledge in CK, it is a sufficient design solution. In ALX3d, validation actions validate a belief, turning it to knowledge. As in CK, such actions include conducting mathematical analyses, experimental tests, etc. The only substantive difference is that in ALX3d the key belief that must be validated is the implication $R \Rightarrow D$, rather than the design *D* itself. In ALX3d, knowing *D* follows from knowing (validating) the implication.

Finally, CK's K→K operators expand knowledge space through logical/scientific reasoning. Any such operator is available within the 1^{st}-order logic underlying ALX3d.

Beyond representing the fundamental features of CK, ALX3 provides the descriptive (not prescriptive) apparatus to represent other aspects of design activities. Though discussed in detail in a previous publication [3], some of those results are reproduced (in slightly modified form) here as potential extensions to a CK-like theory written in ALX3.

First, we will represent three ways of deciding when to change between tasks that advance the design (we shall call these *synthesis* tasks) and tasks that expand the requirements (we shall call these *analysis* tasks). We need to do this when the current state indicates an incomplete design. We can write this as

$$\mathbf{K}_a(\neg(R \Rightarrow D)) \Rightarrow [(R' \wedge D) \vee (R \wedge D')]\mathbf{P}_a \psi. \tag{5}$$

That is, changing only one of either the requirements or the design is preferred by the agent to any other changes. Furthermore, we use the following abbreviations.

$$^{R}P_a \Leftrightarrow_{def} (R \wedge D) \wedge ((R' \wedge D)\mathbf{P}_a(R \wedge D')), \tag{6a}$$
$$^{D}P_a \Leftrightarrow_{def} (R \wedge D) \wedge ((R \wedge D')\mathbf{P}_a(R' \wedge D)). \tag{6b}$$

The first statement means that means that the agent prefers to change the requirements rather than change the design, and the second statement means the converse.

One condition for alternating between synthesis and analysis tasks is that the agent believes there are no further activities of the current type (synthesis or analysis) that can be done. This can be written with two ALX3d statements:

$$^{R}P_a \wedge \mathbf{B}_a(\neg \mathbf{DA}_a R' \wedge R'\mathbf{P}_a R) \Rightarrow \langle a_a \rangle {}^{D}P_a, \tag{7a}$$
$$^{D}P_a \wedge \mathbf{B}_a(\neg \mathbf{DA}_a D' \wedge D'\mathbf{P}_a D) \Rightarrow \langle a_a \rangle {}^{R}P_a. \tag{7b}$$

Alternatively, we can represent an *opportunistic* approach [9], in which an agent will change between synthesis and analysis as soon as the opportunity to do so presents itself. We can write this as follows.

$$^{R}P_a \wedge \mathbf{B}_a(\mathbf{DA}_a D' \wedge D'\mathbf{P}_a D) \Rightarrow \langle a_a \rangle {}^{D}P_a, \tag{8a}$$
$$^{D}P_a \wedge \mathbf{B}_a(\mathbf{DA}_a R' \wedge R'\mathbf{P}_a R) \Rightarrow \langle a_a \rangle {}^{R}P_a. \tag{8b}$$

Finally, we can also represent a situation where changing between analysis and synthesis tasks is left to the preference of the designers. Such changes will occur if the designer is currently doing synthesis (or analysis) tasks, has a choice of changing to the other type of task, and has a preference to change. The rationale for the preference is not pertinent in this perspective; the agent merely asserts the preference. We can write this with the following two statements.

$$^{R}P_a \wedge \mathbf{B}_a[(\mathbf{DA}_a D' \wedge D'\mathbf{P}_a D) \wedge (\mathbf{DA}_a R' \wedge R'\mathbf{P}_a R) \wedge (D'\mathbf{P}_a R')] \Rightarrow \langle a_a \rangle {}^{D}P_a, \tag{9a}$$
$$^{D}P_a \wedge \mathbf{B}_a[(\mathbf{DA}_a D' \wedge D'\mathbf{P}_a D) \wedge (\mathbf{DA}_a R' \wedge R'\mathbf{P}_a R) \wedge (R'\mathbf{P}_a D')] \Rightarrow \langle a_a \rangle {}^{R}P_a. \tag{9b}$$

All three approaches shown above can occur at different points in the same design process and indicate the conditions that must logically exist for designers to undertake certain actions. It does not prescribe the actions to take; it only indicates when opportunities for certain actions exist. In “real life,” there are many other possible guidelines that can be represented; which ones are available in fact will vary from one design situation to the next.

These kinds of guidelines relate to CK in that they show some of the conditions under which expansive partitioning (of the design space) versus restrictive partitioning can occur. Put another way, the guidelines can help designers recognize opportunities to change (and hopefully improve) the way they design, but the guidelines do not enforce any one kind of design process.

A second feature of ALX3d is its capacity to represent some kinds of design principles, such as minimizing the number of parts in a design. Details of the representation are given in [3]; here we will review the results and indicate how it relates to CK. Many early design activities are based on trading off values for different variables based on the preferences of designers. Achieving such goals is a human activity, but defining these trade-off goals can be written in ALX3d as follows.

Let *x*, *y*, and *z* be three designs. Let φ(d) and ψ(d) be functions that map a design *d* to the value of some characteristic of the design; e.g. the number of parts in design *d* could be written *np(d)*.

$$\mathbf{G}^t[\phi(x)] \Leftrightarrow_{def} \phi(x)\mathbf{P}_a\phi(y) \wedge \phi(x) \therefore \psi(u) \wedge \mathbf{B}_a[\neg \exists z\, (\phi(z)\mathbf{P}_a\phi(x) \wedge \phi(z) \therefore \psi(v) \wedge \psi(u)\mathbf{P}_a\psi(v)\,], \tag{10a}$$

$$\phi(x)\mathbf{P}_a\phi(y) \Rightarrow \mathbf{B}_a(x\mathbf{O}_\phi y). \tag{10b}$$

The first statement defines a trade-off goal for the value of some characteristic φ (e.g. number of parts) of a design *x* thus: an agent prefers designs where φ(x) to design where φ(y), so long as the designer believes that more φ-preferred designs are such that some other characteristic ψ (e.g. size, weight, manufacturability, etc.) will obtain a less preferred value. The second statement gives a physical basis for the agent's preference, by associating preference with a belief in a partial ordering $\mathbf{O}_\phi$ with respect to the characteristic φ. Indeed, the second statement indicates a *rationale* for the preference used in the first statement. We note that the rationale is strictly a one-way implication; that is, while a preference implies a belief, the belief does *not* imply a preference.

For example, then, the principle of preferring the lowest possible number of parts can be written by substituting < for $\mathbf{O}_\phi$ and some predicate *np(d)* for φ, such that *np(d)* is the number of parts in design *d*. The rule then captures the notion that a designer will seek to lower the part count only until doing so will cause some other characteristics ψ (e.g. manufacturability) to obtain a non-preferred value.

The partial ordering need not be a conventional mathematical one such as <; it can be any ordering that can be represented in 1st-order logic, including variants such as fuzzy logic, without detracting from the logical soundness of the first statement. In fact, we can omit the partial ordering altogether, so long as the design agents can assert the appropriate preferences.

This demonstrates only one of many kinds of goals that can be represented by ALX3d. The author is currently studying other kinds of goals. The point of this presentation is to demonstrate that one can submit a CK-like theory of designing to logical representation.

4 An example

In [10], Hatchuel et al present examples of the application of CK theory. In this section, the author will discuss how ALX3d can achieve at least the same level of description as CK. We will use one of the examples in [10]: the design of a new chemical (Mg-CO2) rocket motor for use in Mars exploration missions. In [10], the case study is divided into four phases; we will also follow this layout.

The initial state (Phase 0) is the proposal that a Mg-CO2 engine would be "better" than the conventional solution. Per [10], we label this proposal C0. In CK, the proposal is a concept because it has no logical status. In ALX3d, the proposal is a belief, a statement that we assume to be true and then reason with it until we can either prove or disprove it. We write it as $\mathbf{B}_a C0$.

In Phase 1 of the case study, an attempt was made to use the Mg-CO2 concept for a sample return mission to Mars (labelled A1 in [10]). We would write this as $\mathbf{B}_a(C0 \wedge A1)$. An "evaluation" was then carried out by comparing the new motor to existent ones with respect to the criterion of minimum landed mass on Mars. This constitutes a validation action in ALX3d. It was found that the new motor failed the validation. In ALX3d as in CK, this only means that $\mathbf{K}_a \neg(C0 \wedge A1)$. As in CK, and by the fundamental properties of 1st-order logic, this does not necessarily imply that $\neg C0$. So we can preserve our core belief, $\mathbf{B}_a C0$, by contending that $\mathbf{B}_a \neg A1$, which would account for the validation result. The new belief is then $\mathbf{B}_a(C0 \wedge \neg A1)$. Note that in CK, it is assumed at this point that A1 is false; i.e. that the new motor will not work for a sample return mission. This is in fact incorrect; all we can infer *logically* is that we *believe* the Mg-CO2 motor will not work in this kind of mission because all we *know* is that the combination $C0 \wedge A1$ will not work. At this point, CK would have us accept the validity of our main proposal C0, but the whole point of the exercise is to determine if the concept has any merit at all. We see then that ALX3d is more expressive of the actual state of affairs in this case.

In Phase 2 of the case study, it is reported that a study conducted of mission profiles excluding sample return missions (i.e. $\mathbf{B}_a(C0 \wedge \neg A1)$) yielded no positive results, but that this was due to an excessive number of attributes placed on the problem during evaluation. It is also suggested that CK provides a key insight here – that those excess attributes must be removed in order to discover other possible solutions. However, the current author has been unable to find a clear indication of how CK *itself* accommodates this. Indeed, the current author contends that this is a feature of an ontological representation of design problems as a composition of facts. This is how logic works in general, and is not a feature particular to CK. There is an old adage: *always question your premises*. In this case, the premises are the "excess" attributes. Questioning them involves determining whether they are necessary or simply accepted by fiat, convention, or error.

In the case of the Mg-CO2 motor, it is evident that all scenarios had at least one key attribute in common: that the motor would be used during the transit to Mars. This is the premise that is questioned in [10]. In fact, then, the belief (the CK concept) $\mathbf{B}_a(C0 \wedge \neg A1)$ was interpreted incorrectly because the premise of using the motor in transit is not part of the concept; that is, A1 (use for sample return missions) does not necessarily imply use in transit. Formally, we could have written in ALX3d $\mathbf{B}_a(C0 \wedge \neg A1 \wedge A1')$ where A1' stands for "using the motor in transit."

The logical alternative here, regardless of the use of CK or ALX3d, is to use the complement of the premise: use the Mg-CO2 motor for purposes *other than* the transit to Mars. Practically, this is equivalent to using the motor *on* Mars, labelled A2 in [10], which we can represent in ALX3d as $\mathbf{B}_a(C0 \wedge A2)$, so long as we also accept that $A2 \Rightarrow \neg A1$.

One may then continue through the case study, identifying four other attributes that constitute possible uses of the Mg-CO2 motor on Mars: A3 – "used for mobility," A4 – "unplanned mobility," A5 – "emergency lift-off," and A5' – "additional distance." The systematic appearance of these alternatives follows from the use of CK only insofar as CK implies the use of breadth-first searches, which is our only *logical* course of action. A new concept is then specified in [10], which can be written in ALX3d as $\mathbf{B}_a(C0 \wedge A2 \wedge A3 \wedge A4 \wedge A5)$. However, there is a problem here. Technically, both this statement and its CK variant mean that the agent believes an appropriate design is a Mg-CO2 motor used for unplanned emergency lift-off mobility on Mars; that is, the mission involves the *simultaneous* occurrence of A2 through A5, because of the use of conjunctions.

The current author believes the intention was that the new motor could be used in *any combination* of the situations denoted by A2-A5. At very least, a disjunction should have been used, i.e. $\mathbf{B}_a(C0 \wedge (A2 \vee A3 \vee A4 \vee A5))$, to correctly represent that any of A3, A4, or A5 could constitute an appropriate use of the Mg-CO2 motor. However, not even this is the best possible representation, because three important facts are missing: (a) that A2 is a generalisation of A3-A5, (b) that A3 is a generalisation of A4 and A5, and (c) that some design activities occurred to get from A2 to A3 and then to A4 and A5.

The current author therefore suggests the following ALX3d representation for situation reported in [10]:

$$((\mathbf{B}_a(C0 \wedge A2) \therefore \mathbf{B}_a A3) \therefore \mathbf{B}_a A4) \therefore \mathbf{B}_a(A5 \vee A5'). \qquad (11)$$

This statement captures a great deal about the situation:

1. Initially, the agent believes the Mg-CO2 motor is a viable alternative for use on Mars ($C0 \wedge A2$).
2. There is a causal relation leading from "use on Mars" to "mobility on Mars" (A3).
3. To achieve A3, there must exist some design action (the "conceptual expansion" in CK theory) that moves the agent there. This is a human cognitive act connecting a means (the motor) to a desirable capability (mobility).
4. Similarly, once the agent believes A3, there is an expansion action that will lead the agent to a new state where the mobility is unplanned (A4).
5. Finally, once the agent is in a state of believing A4, there is an expansion action that will lead the agent to believe either emergency lift-off or additional distance (A5 or A5') as alternatively suitable situations.

We note that once we reach a state where A5 or A5' is true (and *only* in such states), then we can also say that $\mathbf{B}_a(A5 \vee A5') \Rightarrow A4 \Rightarrow A3 \Rightarrow A2$, which gives a causal chain back to the original propositions. Again, this demonstrates that ALX3d provides a richer representation than CK, while remaining consistent with the intent and general principles of CK.

Finally, in Phase 3 of the case study, a comparison of the Mg-CO2 concept and an alternative design, the ExoMars Rover, is reported. The concept used is that of Mg-CO2 combustion for unplanned mobility on Mars. (Note: A5 and A5' are not used.) The ExoMars performance constraints are given as (a) motor weighing less than 60kg, (b) mission life of no more than 180 days, (c) maximum power consumption of 200W, and (d) minimum 10km range. These constraints are used to limit a performance domain that the Mg-CO2 concept must satisfy. Based on existent knowledge (e.g. principles of rocket propulsion), two key design *parameters* for the Mg-CO2 concept are discovered: motor mass (m_m), and mass of the CO2 acquisition plant (m_p). These two parameters can be used to calculate values for *performance characteristic* of lifetime (t), power (p), and range (r).

The values of the parameters are found to exist within a bounded domain; any value set within the domain constitutes a possible solution, i.e. where the Mg-CO2 motor concept can compete against the ExoMars alternative. The authors argue [10] that this opens up new possibilities for mission concepts and design alternatives that would not have been noticed otherwise.

Phase 3 is described in [10] using text and diagrams, and it is not necessarily clear what activities derived from CK versus the use of rational, logical reasoning in general, or innovative thinking about the problem. No matter which is actually the case, the current author will show that stages of development that occurred in Phase 3 can be represented directly in the language of ALX3d and consistently with CK.

First, let $\Delta\psi$ be true only for design concepts. We would therefore assert ΔC0 to mean that C0 is a design concept. The set of all known design alternatives satisfying some propositions ψ is given in ALX3d by:

$$\mathbf{D}_a(\psi) \Leftrightarrow_{def} \{x: \mathbf{K}_a\Delta x \Rightarrow \mathbf{B}_a\psi\}. \tag{12}$$

The agent *knows* that Δx because the agent asserted it. Note that $\mathbf{K}_a\Delta x$ does not imply $\mathbf{K}_a x$; that is, knowing that we believe a thing is different than knowing the thing itself. For example, consider the previous example of bicycles with pedals and effective wings. Let B0 be "bicycle", P1 be "with pedals", and P2 be "with effective wings". Furthermore, assume we were interested in finding alternatives that have wings (P2) to "bicycles with pedals" ($B0 \wedge P1$). The set of alternatives is given by $\mathbf{D}_a(P2) \Leftrightarrow_{def} \{x: \mathbf{K}_a\Delta x \Rightarrow \mathbf{B}_a P2\}$, which would include bicycles with pedals and any other design concept satisfying "with effective wings." Alternately, $\mathbf{D}_a(P1) \Leftrightarrow_{def} \{y: \mathbf{K}_a\Delta y \Rightarrow \mathbf{B}_a P1\}$ would contain the alternatives to bicycles with effective wings that also satisfy "with pedals."

Now, let $C1 \Leftrightarrow_{def} C0 \wedge A2 \wedge A3 \wedge A4$; i.e. C1 is the concept of using Mg-CO2 motors for unplanned mobility on Mars. The designer can assert ΔC1 as a possible design. The designers' state thus includes $\mathbf{B}_a$C1. To look for alternative concepts, we need to identify concepts that involve A2-A4 – all cases of unplanned mobility on Mars. We can write this as C1\C0, i.e. C1 as defined, but without C0. Now the set of all design alternatives is just:

$$\mathbf{D}_a(C1 \backslash C0) \Leftrightarrow_{def} \{x: \mathbf{K}_a\Delta x \Rightarrow \mathbf{B}_a(A2 \wedge A3 \wedge A4)\}. \tag{13}$$

$\mathbf{D}_a(C1 \backslash C0)$ includes all the design concept alternatives to C1. To gather these alternatives, the designers began with a belief C1, and did the appropriate research (a C→K operator in CK) to find the design alternatives $\mathbf{D}_a(C1 \backslash C0)$. We can represent this as a causal relation in ALX3d: $\mathbf{B}_a C1 \therefore \mathbf{D}_a(C1 \backslash C0) \vee \neg \mathbf{D}_a(C1 \backslash C0)$ – that is, every subsequent state following the search for design alternatives is one that either definitely does or does not have such alternatives. Obviously, to continue the case study, we must assume that $\mathbf{D}_a(C1 \backslash C0)$ is in fact the case.

One might ask: is there some feature of a state where $\mathbf{B}_a$C1 that draws the agent to look for alternatives? The original case study [10] only states "…the prototype should overcome the rover solution for the next known missions…." At this time, the current author can only propose that setting a goal of comparing concepts to alternatives is an *extralogical* design principle. This activity may be a part of a validation action; that is, it is one way to determine if a design concept has merit. This might suggest an *axiom* (a statement accepted as true but not provable within a logic) for ALX3d, but setting out exactly what this axiom might be remains an item for future study.

We are now ready to describe this phase of the case study in ALX3d. Let the design parameters **P** for the Mg-CO2 motor be m_m and m_p. Let the (possibly vague) values of the design parameters be written as functions mapping a design to a parameter value: $m_m(d)$, $m_p(d)$. Let the performance metrics of any designs: $\mathbf{M} \Leftrightarrow_{def} \{p, t, r\}$ (*p*ower, *l*ifetime, *r*ange). The (possibly vague) values of metrics can be written as functions p(d), t(d), and r(d) for a design d.

The metric values are *caused* by the parameter values. That is, the case study model described above indicated that p, t, and r were dependent values, and m_m and m_p were the independent values. In ALX3d, this is written:

$$\forall d[[m_m(d) \wedge m_p(d)] \therefore [p(d) \wedge t(d) \wedge r(d)]]. \quad (14)$$

Furthermore, the values can be partially ordered, e.g. $p(x)\mathbf{O}_p p(y)$ for different designs, where $\mathbf{O}_p$ is a generalised ordering operator (like >) on characteristic p. The ordering may involve imprecision e.g. fuzzy or interval math.

Constraints were defined in the case study based on knowledge of rocketry and physics. Let the constraints be written as: **p** (200 W), **t** (180 d), **r** (10 km). Therefore, a condition for a *satisficing* design [5], $p(d)<\mathbf{p} \wedge t(d)<\mathbf{t} \wedge r(d)>\mathbf{r}$. We can now write a satisficing goal for the Mg-CO2 concept C1 as a belief in a causal relation. Since the design is only a concept, we cannot *know* this satisficing relation, but only believe it. In ALX3d, we can write a satisficing goal for this case as:

$$\mathbf{G}^s[C1] \Leftrightarrow_{def} \mathbf{B}_a[[m_m(C1) \wedge m_p(C1)] \therefore [p(C1)<\mathbf{p} \wedge t(C1)<\mathbf{t} \wedge r(C1)>\mathbf{r}]]. \quad (15)$$

This statement essentially captures the domain of possible values for the identified design parameters such that any design that satisfies this statement is a possible solution.

We can also go beyond the case study somewhat by considering a way to find the best design within the domain of satisficing solutions given by $\mathbf{G}^s[C1]$. Given two designs $C2 \Leftrightarrow_{def} C1 \wedge x$ and $C3 \Leftrightarrow_{def} C1 \wedge y$ that both satisfy $\mathbf{G}^s[C1]$, we can use the formalism of trade-off goals in Section 3 to capture the agent's preference for one satisficing design over another. Given C2 and C3 as defined above, and letting *u*, *v*, and *z* be other satisficing designs in $\mathbf{G}^s[C1]$, and letting ϕ and ψ stand for any two of the metrics, we can write the following.

$$\mathbf{G}^t[\phi(C2)] \Leftrightarrow_{def} [\phi(C2)\mathbf{P}_a\phi(C3)] \wedge [\phi(C2) \therefore \psi(u)] \wedge$$
$$\mathbf{B}_a[\neg\exists z\,(\phi(z)\mathbf{P}_a\phi(C2) \wedge \phi(z) \therefore \psi(v) \wedge \psi(u)\mathbf{P}_a\psi(v)\,], \quad (16a)$$

$$C2\mathbf{P}_aC3 \Rightarrow \mathbf{B}_a[\phi(C2)\ \mathbf{O}_\phi\ \phi(C3)]. \quad (16b)$$

This says that C2 is preferred to C3 if C2 attains a "better" value of one of the metrics (ϕ) than does C3, and doing so will not limit finding a more preferred value for one of the other metrics (ψ).

We have now developed a new model of the case study in [10] that is grounded far better in a formalism of design activities than CK can provide.

5 Potential benefits of ALX3d

In Section 3, the author introduced ALX3d, a formal theory of design activities built upon the action logic ALX3, and designed to account for the key features and intent of CK theory. In Section 4, a case study from the existent CK literature was reworked in ALX3d to demonstrate its representational richness.

This work in no way invalidates CK. Rather, it demonstrates that the fundamental premises of CK are reasonable premises regarding the act of designing; namely, that there is an important difference between knowledge and concepts, and that a rational (logical) process can describe (but not necessarily *explain*) at least some parts of the act of designing. ALX3d also demonstrates the power of logical systems to capture essential aspects of design processes, especially the decisions that designers must make based not only on knowledge but also on their beliefs and preferences.

ALX3d is a research tool, not something to be used by practicing designers. However, continued development of theories in mathematics and the sciences have often let eventually to practical benefits for designers. It is reasonable to assume the same could happen with logical theories like ALX3d and CK. As ALX3d matures, it will be possible to use it for several purposes in this regard, some of which include the following.

Appeal to formal systems researchers. CK theory, which has distinct benefits as a design research tool, is somewhat hindered because it does not conform to conventions of formal systems. ALX3d maintains the intent and basic principles of CK while casting it in a form more readily understood by those with grounding in formal systems. As such, ALX3d makes CK theory more appealing to the community of design researchers who understand and use formal systems, including researchers in artificial intelligence, computer science, and cognitive science.

Reasoning about documented design processes. Assuming a complete description of a design process as documented either in industry or the literature can be constructed (and the author currently believes this is entirely possible), then the description can be reasoned about using the inference rules that are built into ALX3 to study the process, and find and address its problematic aspects. This would significantly advance our understanding of the nature of engineering design.

Construction of new design processes. It may well be that in the natural course of analysing design processes, new process descriptions may arise that could significantly improve the design capability of a group of designers.

Construction of new computer-based design aids. Logical systems are well-suited to implementation in computer tools. It should be possible to use ALX3d to develop new design applications of artificial intelligence and knowledge-based systems. Such systems may also yield significant advantages for practicing designers.

6 Conclusion

ALX3d, a formal version of CK theory based on the action logic ALX3, is introduced. Beyond what is currently possible with CK, ALX3d leverages ALX3 to provide a richer framework for describing design activities in formal terms. While adding support to the CK approach, ALX3d also demonstrates the potential benefits of using formal systems in design research. Although ALX3d is still being developed, there are strong indications, as demonstrated in this paper that it may be a useful tool for design research.

References

[1] A. Hatchuel and B. Weil. 2003. A new approach of innovative design: an introduction to C-K theory. Proc ICED, paper #1794.

[2] A.O. Kazakci and A. Tsoukias. 2004. Extending the CK design theory to provide theoretical background for personal design assistants. Proc Design 2004, pages 45-52.

[3] F.A. Salustri. 2003. Towards an action logic for design processes. Proc ICED, paper #1051.

[4] Z. Huang. 1994. Logics for agents with bounded rationality. PhD Thesis, University of Amsterdam.

[5] H.A. Simon. 1981. The Sciences of the Artificial. The MIT Press, Cambridge, Mass.

[6] V. Akman and M. Surav. 1996. Steps toward Formalizing Context. AI Magazine, 17(3):55-72.

[7] R.J. Brachman, D.L. McGuinness, P.F. Patel-Schneider and L.A. Resnick. 1991. LIVING WITH CLASSIC: When and How to Use a KL-ONE-Like Language. In Principles of Semantic Networks: Explorations in the Representation of Knowledge (ed. John F. Sowa); Morgan Kaufmann Publishers, Inc., San Mateo; pages 401-456.

[8] J.S. Gero. 2004. Situated design computing: Introduction and implications, in D. Marjanovic (ed), DESIGN 2004, The Design Society, Glasgow, pages 27-36.

[9] M.J. French. 1992. The Opportunistic Route and the Role of Design Principles. Research in Engineering Design, 4(3):185-190.

[10] A. Hatchuel, P. Le Masson, and B. Weil. 2004. CK theory in practice: lessons from industrial applications. Intl Design Conference, pages 245-257.

Filippo A. Salustri
Department of Mechanical and Industrial Engineering
Ryerson University
350 Victoria Street
Toronto, ON, M5B 2K3, Canada
tel: +001-416-979-5000 x7749
fax +001-416-979-5265
email: salustri@ryerson.ca
http://deed.ryerson.ca/~fil