The Model of Hierarchical Complexity as a Measurement System

Michael Lamport Commons
Harvard Medical School

Robin Gane-McCalla
Dare Institute

Cory David Barker
Antioch University

Eva Yujia Li
Harvard Graduate School of Education

Author Note

Correspondence regarding this article should be addressed to Dr. Michael Lamport Commons, Harvard Medical School, 234 Huron Avenue, Cambridge MA 02138. E-mail: commons@tiac.net

Abstract

The Model of Hierarchical Complexity (MHC) is a mathematical model based on the "Theory of Measurement" that has gone through a number of iterations as a measurement system (Commons, Goodheart, Pekker, et al., 2005; Commons & Pekker, 2008; Commons & Richards, 1984a, 1984b; Commons, Trudeau, Stein, et all, 1998). It sets forth the measurement system by which actions are put into a hierarchical order and each order is assigned an ordinal number. In this paper, the components of the Model will be described: Actions and tasks, measurement and operations, and the axioms, followed by an articulation of emerging properties from axioms, and then a description of orders of hierarchical complexity of tasks. These are a reworked smaller set of axioms, which are more measurement-theoretical in nature. They also parallel the informal conditions underlying the kind of complexity that the MHC entails.

Keywords: Model of Hierarchical Complexity, Theory of Measurement, stages, actions, tasks, subtasks, subsubtasks, nominal scale, operations, measurements

The Model of Hierarchical Complexity as a Measurement System

The Model of Hierarchical Complexity (MHC) is a mathematical model that sets forth the measurement system by which actions are put into a hierarchical order. The model assesses a general, unidimensional developmental measure of difficulty across domains. Dawson-Tunik's (2006) studies have found that the stage of development scored according to the Model of Hierarchical Complexity was consistent with multiple other instruments that were designated to score development in specific domains.

Model of Hierarchical Complexity is not the only theory of development based on task complexity. Other metrics of task complexity have been proposed as well. Horizontal or classical information complexity is one of them. It describes the number of "yes-no" questions. In classical information complexity, if a task requires one such question, the answer would transmit 1 bit of "horizontal" information. Similarly, if a task requires two such questions, the answers would transmit 2 bits. Each additional 1-bit question would add another bit. Horizontal complexity, then, is the sum of bits required by tasks that require "yes-no" questions. The number of actions is 2^n.

Older metrics of task complexity such as the horizontal complexity and others have a few limitations. What is promising about the Model of Hierarchical Complexity is that it is a newer model which overcomes those limitations as it is not content bound, does not miss stages and does not have any assumptions. It is based on vertical complexity and involves hierarchical information. Hierarchical complexity refers to tasks that require the performance of lower-level subtasks in order to perform more complex, higher level tasks.

An advantage of the Model of Hierarchical Complexity is that it is applicable to any domain of development in both humans and animals, such as social, cognitive, personal and such. This is not the only advantage of MHC. MHC also seems to have advantage over previous proposals about developmental stages of humans. While previous models attribute behavioral changes across a person's age to the development of mental structures, MHC posits that task sequences of task behaviors form hierarchies that become increasingly complex. According to this model, less complex tasks must be completed and practiced before more complex tasks can be acquired. Thus, it accounts for developmental changes. Furthermore, previous theories of stage have confounded the stimulus and response in assessing stage by simply scoring responses and ignoring the task or stimulus. The model of hierarchical complexity separates the task or stimulus from the performance. The participant's performance on a task of a given complexity represents the stage of developmental complexity. Another factor which sets this model apart from previous models is that it not only extends developmental stages up to 15 stages, but also includes subtasks and subsubtasks which explain what happens between those stages.

As explained above, the model has many advantages. It has gone through a number of iterations as a measurement system (Commons, Goodheart, Pekker, et al., 2005; Commons & Pekker, 2008; Commons & Richards, 1984a, 1984b; Commons, Trudeau, Stein, et al, 1998). The model's empirical usefulness has also been set forth in earlier papers (Commons, Goodheart, Pekker, et al., 2005; Commons, Goodheart, Rodriguez & Gutheil, 2006; Commons & Pekker, 2008). However, this model has never been formally described. In this paper, the components of the model will be formally described: actions and tasks, measurement and operations, and the axioms. It will be followed by an articulation of emerging properties from axioms and a description of stages.

Distributivity as an Example

Hierarchical Complexity can be illustrated with the example of the distributive property. The distributive property refers to when two sides of an equivalence "=" are represented differently, yet are equal. The distributive property describes a characteristic feature of some binary operators, namely that one argument must be "distributed" to the various elements of the other argument. Take for example $a \times (b + c) = (a \times b) + (a \times c)$. That says that one distributes the × across each term connected by the + action. The distributive property plays a fundamental role in more general contexts, such as the complex numbers and the definition of rings in modern algebra.

The distributive law serves as a motivation for a newer form of complexity, called hierarchical complexity, formally presented here. In the case of evaluating $a \times (b + c)$, the task of distributing is more hierarchically complex than the two-part task of first evaluating $b + c = d$ and then evaluating c × d. In the case of $(a + b) + c$, the organization of two actions of addition is arbitrary and no more hierarchically complex than addition in the evaluation of $(a + b) + c$ or $a + (b + c)$, because addition is associative. In the case of evaluating $a \times (b + c)$ it is more hierarchically complex than the task of evaluating $(a + b) + c$, because evaluating $a \times (b + c)$ requires the two actions of addition and multiplication to be performed in a certain order.

Actions and Tasks

In the context of the Model of Hierarchical Complexity, *actions* are defined as behavioral events that produce outcomes. Actions may be attributed to organisms, social groups, and computers. Actions may be combined to produce new, more complex actions (Binder, 2000). *Events* are perturbations that can be detected by at least two independent paths (Commons, 2001). A *task* can be defined as a set of required actions that obtain an objective, though the performed actions may or may not complete a given task. The study of tasks appears in psychophysics, a branch of stimulus control theory in psychology (Green & Swets, 1966; Luce, 1959) and in artificial intelligence (Goel & Chandrasekaran, 1992).

Hierarchical Structure of Tasks

The hierarchical order is constructed by seeing how one action is more complex than another, as illustrated above with distributivity as an example. A higher order action is defined in terms of two or more order actions of one order below, and the higher order action non-arbitrarily organizes those next lower order actions, as illustrated by Figure.1 below. Mathematically speaking, we refer to distribution used in long multiplication, such as $a \times (b + c)$, as organizing the lower order actions of addition and multiplication, in non-arbitrarily ways.

By definition, only the coordination of two or more next lower order actions produces an action at the next higher order. Coordinating actions of different orders result in other types of actions, and observations of these differences allow descriptions to be given about the orders. Orders have subtasks and subsubtasks between them. *Subtask actions* organize only one action from the same order and one or more from previous orders. They are prerequisites to other same order tasks. For example, the seventh order of Hierarchical Complexity is called primary. At this stage, the ability to do simple logical deduction, and simple arithmetic is attained. Examples of tasks it accomplishes are counting, addition and multiplication. Counting is one subtask action that is a prerequisite for addition. Addition, is another subtask action, and is a prerequisite for multiplication. They do not coordinate two or more actions, but coordinate one action from the same order and one or more from lower orders. Such coordination does not result in an increment of order. *Subsubtask actions* coordinate actions from different orders that are precursors but not prerequisites for subactions (see Figure 1).

Orders of Hierarchical Complexity form an ordinal scale with the first four axioms and definitions that follow. A fifth axiom makes all of the orders of hierarchical complexity equally spaced – that is of equal difficulty.

In the next section of the paper, a description of the mathematical basis that defines the Model of Hierarchical Complexity will be given, followed by the presentation of the formal, axiomatic version of the theory.

Measurement and Operation

Measurement is the process of associating numbers with entities or objects. In this section of the paper, a description of the components of the model will be given - the system of entities, concatenation and comparison mathematical operators and the assignment function.

System of Entities

To develop a system of measurement, one must represent the entities to be measured. In this case, the entities are task actions of organisms, social groups, and computers (Krantz, Luce, Suppes, & Tversky, 1971). In the modern algebraic representation of the Model of Hierarchical Complexity, actions are represented by letters or numbers. A system of entities, as a set of actions, is represented by letters such as *A*. Unless these actions are the most simple and irreducible of actions, they are composed of other actions concatenated together – the simplest actions do not act upon other actions.

Concatenation Operators

In mathematics and logic, an operation is an agent which executes on one or more input values and produces a new value. Concatenation operators are methods in which actions are connected. They are represented by "○". They specify the order in which actions are executed, so that the order is fixed and not commutative: $a \circ b \neq b \circ a$. In the simplest terms, the concatenation operator is an ordering relation on the entities and also may be described by order relations: $A = (a, b) = \{a, \{b\}\}$, an ordered pair.

It was stated that mathematically speaking, we refer to distribution used in long multiplication as organizing the lower order actions of addition and multiplication in a non-arbitrary way. The non-arbitrary organization of addition and multiplication is accomplished by the concatenation.

Comparison Operators

It was stated that actions are put into a hierarchy. Higher order actions are defined in terms of next lower order actions and non-arbitrarily organize the next lower order actions. The comparison operator, represented by $\succ$, is used to arrange actions in a hierarchy.

In the case of the real numbers, the system of entities is *R,* the real numbers, the comparison operator is > and the concatenation operator is +. In the case here, the entities are the actions in a system and the numerical relational structure is the ordinal numbers (positive numbers and zero). The comparison and concatenation operators are the same as they are for the real numbers.

Assignment Function

The assignment function is used to assign a numerical relational structure to the complexity of actions, which allows the complexity of actions to be stratified hierarchically into orders. In other words, the assignment function numbers the incrementally increasing complexity of actions *as orders*. It assigns numbers to those actions based on the complexity of those actions. Mathematically, the assignment function is represented by the Greek letter φ(phi).

An assignment function is a *homomorphic mapping* that transforms the entities to be measured into a numerical relation structure. Abstract algebra studies sets that are endowed with operations that generate interesting structure or properties on the set. Therefore, functions that preserve the operations are especially important. These functions are known as *homomorphisms*. In our case, the function maps a set of actions and their concatenation to a number, n (positive whole number or zero). The numerical relational structure preserves the relationship between actions – the more hierarchically complex the actions, the higher the number assigned.

For example, if a is an n order action the assignment function φ assigns the number n to a, which is denoted by $\varphi(a) = n$. The assignment function, $\varphi(a)$, denotes the *order of hierarchical complexity* (OHC) of a task action. The *measure of hierarchical complexity at order n* is defined as the minimum number of simple actions required to complete an action of that order.

The most irreducible action is at order 0. Order 0 is not really an action in the usual sense because it is not planned or controlled by the machine but by the programmer. It has no order of hierarchical complexity and therefore cannot be reduced. The first order that has an order of hierarchical complexity is of order 1 actions. Order 1 actions do not organize any actions so they are simple actions, because those actions have no order; that is why it is called order 0. The actions of order 2 are made out of actions of order 1. The actions of order 3 are made out of actions of the order 2, and so on. The repeating process of an order of actions defined in terms of next lower order actions produces the numerical relation structure, and stratifies orders of Hierarchical Complexity.

Axioms

The measurement system of the Model of Hierarchical Complexity is composed of axioms. Axioms are rules that are followed to determine how the Model of Hierarchical Complexity orders actions to form a hierarchy. There are five axioms: *well ordered, transitive, chain rule, coordination rule* and *equal spacing* (optional axiom). The axioms presented in the sections that follow are simplifications, reductions, refinements, and improvements that build on Piaget (e.g., Inhelder & Piaget, 1958) and his intellectual descendants (e.g., Campbell, 1991; Campbell & Bickhard, 1986; Commons & Richards, 1984a, 1984b; Commons, Richards & Kuhn, 1982; Tomasello & Farrar, 1986).

The Well Ordered Axiom and the Transitive Axiom are rules that describe how the orders of hierarchical complexity are arranged. These axioms are found in most systems of measurement (Krantz, Luce, Suppes, & Teversky, 1971) including an ordinal a system of measurement.

Axiom 1, *Well Ordered*: If $a \succ$ b, then $\varphi(a) > \varphi(b)$

Axiom 1 states that if one action is less complex than another action, then the assignment function, which gives a numerical order of hierarchical complexity to an action, must preserve the action's order in the hierarchy. In other words, under the conversion of actions into numbers by applying the mathematical assignment function φ, action a remains more hierarchically complex than action b. Breaking this rule would mean that the order of hierarchical complexity would not be uniform for all actions in which simpler actions are non-arbitrarily put into order by the Model of Hierarchical Complexity.

Axiom 2, *Transitivity*: If $a \succ b$ and $b \succ c$ then $a \succ c$

Axiom 2 means that if action a is more complex than action b, and action b is more complex than action c, then action a is more complex than action c. In other words, the transitive property places the actions in a sequential hierarchical order. Breaking this rule would be similar to breaking Axiom 1, in that the numerical relational structure could not preserve the non-arbitrary sequence of Orders of Hierarchical Complexity.

The following axioms regard how the concatenation operator affects the assignment function: the Chain Axiom and the Coordination Axiom describe the arbitrary and non-arbitrary character of the order of actions.

Axiom 3, *Chain rule:* $\varphi(a \circ b) = \max(\varphi(a), \varphi(b))$ if $\varphi(a \circ b) = \varphi(b \circ a)$

Axiom 3 states that when actions a and b are chained together in some order, and the order in which they are executed is not influential to accomplishing a task, the order of hierarchical complexity of $(a \circ b)$ equals to that of the highest subaction. Chaining together the two actions does not produce an action that is hierarchically more complex than either of the subactions.

Consider a scenario when a person's goal is to put on a pair of socks. It does not matter which sock a person puts on first, because the end result of the task is the same – both feet have socks on them. Another example would be a child who wants to try all the equipment at a playground. If the child sets out to play on swings and use the slide, it does not matter in which order these actions take place, so long as both are accomplished. The sequence in which the actions are combined does not bring about a higher order of complexity. In this case, $\circ$ is chaining together of actions in some order. But with chaining, the hierarchical complexity of the new task does not increase. What this rule means, is that the way that two actions are combined is arbitrary. This is shown by the commutivity of a and b.

By axiom 3, an action of order n organizes at least two actions of order $n - 1$, each of which in turn organizes at least two actions of order $n - 2$, and so forth, until we reach the lowest-order, simple actions.

Axiom 4, *Coordination rule*: $\varphi(a \circ b) = \max(\varphi(a), \varphi(b)) + 1$ if $\varphi(b) = \varphi(a)$ and $\varphi(a \circ b) \neq \varphi(b \circ a)$.

In this case, ○ coordinates the organization of the ordering of action rules in a non-arbitrary way. In addition, action *a* and action *b* has to be on the same stage. When these two conditions are satisfied, the coordination of action *a* and action *b*, which is represented by ($a \circ b$), is one order more complex than either of the subactions. $\varphi(b) = \varphi(a)$ is necessary because, in order for the coordinated action to move up a stage, the actions have to be on the same stage. The coordination of two actions on different stages does not produce an action that is one stage higher.

A permutation, *P*, can be defined as the same elements happening in different orders, for example (x, y) or (y, x). Such permutations are not commutative of this axiom, i.e., (x, y) ≠ (y, x). With the Coordination Axiom, not all permutations, *P*, are acceptable, that is, not *P* (*a, b*). This rule indicates that there is a possible match between the model-designated orders and the real world functioning of the order of those actions which the model-designed orders represent. To give examples, consider the above two. Returning to the sock example, one does not put shoes on first, then put socks over the shoes. Similarly with the child at the playground, the child must climb the stairs to the slide before going down the slide. These are examples of coordinating actions.

Axiom 5, *Equal spacing* (optional): $\mathrm{OHC}(n + 1) - \mathrm{OHC}(n) = 1$
where, $\mathrm{OHC}(n) = \varphi(a)$, then for every order *n*, $(n)(\mathrm{OHC}(n+1) - \mathrm{OHC}(n)) = 1$

Axiom 5 states that the a priori difficulty of a task action changes by 1 for each change in the Order of Hierarchical Complexity, irrespective of what adjacent Orders of Hierarchical Complexities one is comparing. In other words, there is equal spacing between each order. This implies that the difficulty of going to the next order is the same regardless of what order is being required. This allows one to treat orders as actual numbers, and not just indication of relative position.

It might mean the order of Hierarchical Complexity, *n*, is a measure of the quantity of hierarchical information. Given that tasks at order $n + 1$ are defined by and coordinate two or more tasks at order *n*, the minimum number of order 1 tasks that an order *n* task is 2^n. Equal spacing might indicate that 2^n is well defined and therefore, $\log 2^n = n$, a parallel notion to bits. That might mean that *n* is a measure of the quantity of hierarchical information and could be called Hbits.

A Formal Definition of the Model of Hierarchical Complexity

With the above mentioned, we can now give more specific definitions about the Model of Hierarchical Complexity. There are certain properties that emerge when certain rules are in play. In this section, these properties are articulated as definitions.

Definition 1: There exists *simple actions*, x with $\varphi(x) = 1$. This is the lowest order action.

Definition 2: If there is no action, such as a computer calculating what has been programmed by a person, then the null action, such as the computer action, is at order 0. The computer program may act at a higher order, but it is just a reflection of a programmer solving a very much higher order task. There is no flexible action. That does not mean that variables may not be part of the program, or that randomness could not also be generated, but that the program only does what it is programmed to do. When a person enters information to a computer and a program, the action is by the person. All the action is done under control of the programmer. The exception would neural networks and stacked neural networks and the like.

Definition 3: A higher order hierarchically complex action is defined in terms of two or more next lower order actions, $A = \varphi\ (B \circ C)$, where $\varphi\ (B)$ and $\varphi\ (C)$ are both less hierarchically complex than $\varphi\ (A)$ if $\varphi\ (B \circ C)$ is a coordination. This creates the hierarchy:

$A = \{a, b\}$ a, b are lower order of hierarchically complexity than A and together composes set A

$A \neq \{A,...\}$ A set cannot contain itself . (See Russell's paradox (1902; 1980)

This definition follows definition 1 and Axiom 4.

Next, the differences between chain rules and coordination rules are explained in more depth.

Definition 4: Given a permutation of concatenated actions = $(i_1,i_2, \ldots ,i_n)$ of the natural numbers 1, 2, . . . , n, the execution of action A is simply $A_{i1} \circ A_{i2} \ldots \circ A_{\mathrm{in}}$.

The rule, *R,* is a chain rule if the outcome of the action is the same for all $n!$ permutations of the numbers 1, 2, . . . , n. The outcome of the order of actions, $A_{\mathrm{i}1} \circ A_{\mathrm{i}2} \circ \ldots \circ A_{in}$ is the same for all permutations $(i_1,i_2, \ldots ,i_n)$ of 1, 2, . . . , n.

Rule, *R,* is a coordination rule if there exists at least one permutation of actions $R = (j_1, j_2, \ldots, j_n)$ of the numbers 1, 2, . . . , n so that the execution of the actions A_i i.e., $A_{j1} \circ A_{j2} \ldots A_{jn}$, is *not* the same as the outcome of the action A. Hence, the outcome of A_i is given by at least one, but *not all*, permutations of the Ai. This extends similarly to the cases where A consists of infinitely many actions.

Note that by Axiom 4, a coordination action $A = (\{A_1, \ldots\}, R)$ necessarily coordinates subactions of subtasks of equal orders of hierarchical complexity (i.e., $\varphi(A_1) = \varphi(A_2) = \ldots$). Thus the order of hierarchical complexity of A is one higher than the order of hierarchical complexity of all its subactions. Therefore, A_1 may be replaced by any subaction of A and still obtain the same result. As a consequence of these axioms, we see that if we let $\boldsymbol{A}$ denote the collection of all actions in a given system, then the order of hierarchical complexity is a function h: $\boldsymbol{A} \rightarrow \boldsymbol{N}$, where $\boldsymbol{N} = \{0, 1, \ldots\}$ is the set of natural numbers (and zero) under the usual ordering.

The following properties emerge from the axioms and the definitions:

1. *Discreteness*: The order of hierarchical complexity of any action is a nonnegative integer. In particular, there are gaps between orders.

2. *Existence*: If there exists an action of order n and an action of order $n + 2$, then there necessarily exists an action of order $n + 1$.

3. *Comparison*: For any two actions A and B, exactly one of the following holds: $\varphi(A) > \varphi(B)$, $\varphi(A) = \varphi(B)$, $\varphi(A) < \varphi(B)$. That is, the orders of hierarchical complexity of any two actions can be compared.

4. *Non-reducibility*: A higher order action cannot be equal to any lower order actions. This property arises from the coordination rule, which claims that the coordination of two or more actions at the same order produces an action that is one order above.

Concepts from set theory are applied here to clarify why two order tasks can be non-arbitrarily ordered only at the next order. The higher order corresponds to a set A. Assume $A = \{a, b\}$. The lower order relations in the system correspond to the elements of lower order elements of the set, actions a and b. This creates the hierarchy because the set A is not the same as its elements a and b. The elements are at a lower order than the set. Therefore, the order of the set is not equal to the order of its elements, and $n + 1 \neq n$. Hence, the orders cannot be collapsed.

For example, consider an empty set Ø. Russell argued that an empty set cannot be a member of itself (Godehard, 2004). An empty set Ø = { } has no member. Having no members mean that there is nothing in it, or the member is "nothing". Because Ø is a set, it is "something". Something cannot equal to nothing. Therefore, an empty set Ø cannot equal to its member. Likewise, a higher order action cannot equal to any lower order action from which it is made.

This is consistent with Inhelder and Piaget and the Model of Hierarchical Complexity. These theories state that each next order actions coordinates the actions performed at the preceding order of complexity. To apply the premise successfully, the actions of each stage must be unambiguously specified. The stage generator concept successfully eliminates ambiguity about what makes a stage a stage by precise specification.

Stages Defined

The notion of stages is fundamental in the description of human, organismic, and machine evolution. Previously it has been defined in some ad hoc ways; here it is described formally in terms of the Model of Hierarchical Complexity. Given a collection of actions A and a participant S performing A, the *stage of performance* of S on A is the highest order of the actions in A completed successfully.

$\text{Stage}(S, A) = \max \{h(A) \mid A \in A \text{ and } A \text{ completed successfully by } S\}$.

This is in agreement with previous definitions (Commons, Trudeau et al. 1998; Commons and Miller 2001).

Conclusion

This paper accomplishes two goals. First, it formalizes the Model of Hierarchical Complexities as a measurement system. The components of the measurement system are conceptualized for the first time - the system of entities, comparison operator, concatenation operator and the assignment function. The comparison and concatenation operators describe how the actions are structured. The assignment function is the procedure which assigns numbers to the actions.

Secondly, this paper clarifies key concepts of the model. Axiom 4 articulates the coordination rule. The 4th property, non-reducibility, is demonstrated by proving that a higher order action cannot equal to any lower order actions. In addition, the chain rule and coordination rule are put into mathematical expressions, making them succinct and absolute. This paper substantializes axioms and definitions of the model, which provides theoretical foundation for utilizing the model to measure the task order of actions.

References

Binder, C. (2000). *Component/composite analysis and programming from the real world*. Paper presented at the annual meeting of the Association for Behavior Analysis, May, in Washington, DC.

Campbell, R. L. (1991). Does class inclusion have mathematical prerequisites? *Cognitive Development,* 6:169-94.

Campbell, R. L. & Bickhard, M.H. (1986). *Knowing levels and developmental stages*. Basel, Switzerland: Karger.

Commons M.L. (2001). Problems with mentalistic explanations, freewill, self, soul, and intrinsic motivation. *Metanexus: The Online Forum on Religion and Science*, 08.21. http://www.metanexus.net/magazine/ArticleDetail/tabid/68/id/3966/Default.aspx

Commons, M. L., Goodheart, E.A., Pekker, A., Dawson, K., Draney, T. L. & Adams, K. M.(2007). Using Rasch scaled stage scores to validate orders of hierarchical complexity of balance beam task sequences. In *Rasch measurement: Advanced and specialized applications*, ed. E. V. Smith, Jr. and R. M. Smith, 121-47. MapleGrove, MN: JAM Press.

Commons M.L., Goodheart, E.A., Pekker, A., Dawson-Tunik, T.L., Cyr, E., Rodriguez, J.A., et al. (2005). The relationship between orders of hierarchical complexity and Rasch scaled stage scores: Balance beam, laundry, and counselor-patient task sequences. *Journal of Applied Measurement 2005;*10(2): 1-11.

Commons, M. L., Goodheart, E. A., Rodriguez, J. A. & Gutheil, T. G. (2006). Informed Consent: Do you know it when you see it? *Psychiatric Annals*, June 430-435.

Commons, M. L., Trudeau, E. J., Stein, S. A., Richards, F. A. & Krause, S. R. (1998). The existence of developmental stages as shown by the hierarchical complexity of tasks. *Developmental Review 8 (*3), 237-278.

Commons, M. L., Richards, F. A. & Kuhn, D. (1982). Systematic and metasystematic reasoning: A case for a level of reasoning beyond Piaget's formal operations. *Child Development, 53*, 1058-1069.

Commons, M. L. & Miller, P. M. (2001). A quantitative behavioral model of developmental stage based upon hierarchical complexity theory. *Behavior Analyst Today,* 2(3): 222-240.

Commons, M. L., & Pekker, A. (2008). Presenting the formal theory of hierarchical complexity. *World Futures: Journal of General Evolution 65*(1-3), 375-382.

Commons, M. L., & Richards, F. A. (1984a). A general model of stage theory. In M. L. Commons, F. A. Richards, & C. Armon (Eds.), *Beyond formal operations: Vol. 1. Late adolescent and adult cognitive development* (pp. 120-140). NY: Praeger.

Commons, M. L., & Richards, F. A. (1984b). Applying the general stage model. In M. L. Commons F. A. Richards, & C. Armon (Eds.)*, Beyond formal operations: Vol. 1. Late adolescent and adult cognitive development* (pp. 141-157). NY: Praeger.

Godehard, L. (2004). *One hundred years of Russell's paradox*. Germany: Walter De Gruyter Inc

Goel. A. & Chandrasekaran, B. (1992). Case-based design: A task analysis. In *Artificial intelligence approaches to engineering design, Vol. II: Innovative design*, ed. C. Tong and D. Sriram, 165-184. San Diego: Academic Press.

Green, D. M. & J. A. Swets. (1966). *Signal detection theory and psychophysics*. Huntington, NY: Krieger.

Inhelder, B. & Piaget, J. (1958). *The growth of logical thinking from childhood to adolescence.* New York: Basic Books.

Krantz, D. H., Luce, R.D, Suppes, P. & Tversky, A. (1971). *Foundations of Measurement, Vol. I: Additive and polynomial representations.* New York: Academic Press.

Luce, R. D. (1959). *Individual choice behavior: A theoretical analysis*. New York: Wiley. ISBN 0-486-

44136-9.

Luce, R. D. (1963). Detection and recognition. In *Handbook of mathematical psychology*, ed. R. Luce and R. Bush, 103-90. New York: Wiley.

Rasch, G. (1980). *Probabilistic model for some intelligence and attainment tests.* Chicago: University of Chicago Press.

Russell, B. (1902). Letter to Frege. In van Heijenoort, J (Eds.), *From Frege to Gödel*, (pp. 124-125), Cambridge, MA : Harvard University Press.

Russell, B. (1980)*. Correspondence with Frege. In Philosophical and Mathematical Correspondence*, by Gottlob Frege. Translated by Hans Kaal. Chicago: University of Chicago Press

Tomasello, M. & Farrar, M. J. (1986). Joint attention and early language. *Child Development 57*(14), 54-63.

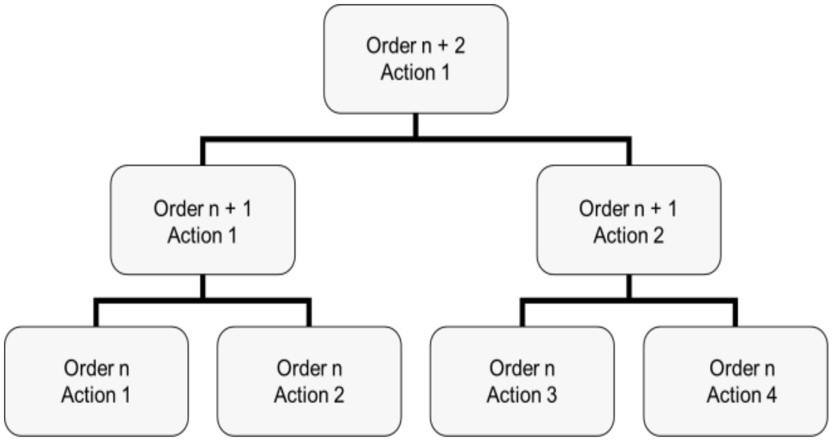

Figure 1. Hierarchical structures of tasks