ARTICLE IN PRESS

Information Sciences xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment

S.J. Overbeek [a,*], P. van Bommel [b], H.A. (Erik) Proper [c]

[a] Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, 2600 GA Delft, The Netherlands
[b] Institute for Computing and Information Sciences, Radboud University Nijmegen, Heijendaalseweg 135, 6525 AJ Nijmegen, The Netherlands
[c] Public Research Centre Henri Tudor, 29, Avenue John F. Kennedy, L-1855 Luxembourg-Kirchberg, Luxembourg

ARTICLE INFO

Article history:
Received 29 May 2008
Received in revised form 20 August 2010
Accepted 1 September 2010
Available online xxxx

Keywords:
Cognitive characteristics
Knowledge intensive tasks
Matchmaking
Object-Role Modeling
System dynamics
Quality factors

ABSTRACT

Two main ingredients related to successful task performance are cognition and quality. Supply and demand of these concepts for knowledge intensive tasks are studied in this paper to fuel successful task fulfillment. Cognitive characteristics are supplied by actors performing tasks. Organizational developments such as growing complexity and increasing customer orientation may increase cognitive load. Stakeholders of tasks have quality requirements. These requirements may be affected if actors experience an increase in cognitive load. It is observed that knowledge intensive tasks demand cognitive characteristics and supply quality factors. Actors supply cognition and stakeholders demand quality. The gap between supply and demand can be bridged by introducing several models. These models consist of a matchmaking framework, conceptual models, and dynamic models. The matchmaking framework shows how supply and demand of cognitive characteristics or quality factors can be matched. Relations and roles of the concepts involved in task fulfillment are mapped out by the conceptual models. The dynamic models show causes that have effects on the supply of cognitive characteristics and the level of quality. These insights in the relations and dependencies between cognition and quality increase our understanding of the key concepts for successful task fulfillment.

1. Introduction

The importance of an actor's abilities to acquire, apply and test knowledge increases due to, e.g. growing process and product complexity, the move toward globalization, the emergence of virtual communities and organizations, and the increase in focus on customer orientation [28]. When the pressure to acquire, apply and test more knowledge increases, actors struggle to manage basic cognitive characteristics like, e.g. the willpower to fulfill a knowledge intensive task or maintaining awareness of the requirements to fulfill such a task. These cognitive characteristics are also referred to as *volition* and *sentience*, respectively in cognitive literature [8,34]. Cognitive characteristics are specific cognitive parts that are possessed by an actor which enable an actor to think, learn and make decisions [2,3,26]. A knowledge intensive task is a task for which acquisition, application or testing of knowledge is necessary in order to successfully fulfill the task [21,25].

The level on which cognitive characteristics are supplied by actors may influence the quality of fulfilled knowledge intensive tasks [12]. If an actor fulfills a task, it is possible to conceive the quality of the *process* that has led to the fulfillment as well as the quality of the task *result* [5]. Quality of the task result refers to product factors and the extent to which they meet stakeholder demands [32]. A stakeholder may be highly concerned about, or have interests in, the quality factors of a fulfilled

* Corresponding author. Tel: +31 15 2785526; fax: +31 15 2783741.
E-mail addresses: S.J.Overbeek@tudelft.nl (S.J. Overbeek), P.vanBommel@cs.ru.nl (P. van Bommel), Erik.Proper@tudor.lu (H.A. (Erik) Proper).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.09.002

knowledge intensive task. Typical examples of stakeholder types found in the literature are, amongst many others, the supplier, the employee, the customer, and the shareholder [24]. It is assumed that a stakeholder has a *goal* related to the fulfillment of a knowledge intensive task that leads to quality expectations. When viewing process quality from a cognitive standpoint, the quality factors of the application of cognitive characteristics during task execution can be considered. This can be dubbed as *cognitive process quality*. Simply put, an actor applies several cognitive characteristics during task execution. This application may vary in quality dependent of how well an actor is able to supply them.

It is observed that cognitive characteristics are supplied by actors and demanded by knowledge intensive tasks. Quality factors are 'supplied' by means of the task result and the cognitive process. These factors are demanded by stakeholders of fulfilled knowledge intensive tasks. Supply and demand of cognitive characteristics and quality factors can be *matched* to improve task performance, decrease cognitive load, and increase product and process quality [7,10,12,33]. In previous work, a formal framework for cognitive matchmaking [19,22] and a prototype of the cognitive matchmaker system [18,19] has been elaborated. The prototype is in fact a Web-based implementation of the framework for cognitive matchmaking. The framework and the prototype have been evaluated in a case study in information systems engineering [18–20]. An overview of the formal framework for cognitive matchmaking, the prototype, and the case study can be found in [16,17].

The research reported in this paper is specifically concerned with bridging the gap of supply and demand for knowledge intensive tasks based on cognition and quality. It is desired to reach three main goals by bridging this gap, which are also pictured in the reasoning framework of Section 2. First, Section 3 shows how the framework for cognitive matchmaking from earlier work can also be used as a framework for qualitative matchmaking. Second, it is intended to understand the relations between the five main concepts that we have related with knowledge intensive task fulfillment and the roles that these concepts play in knowledge intensive work. This has been done by conceiving the conceptual models introduced in Sections 4 and 7. The five main concepts are the actor, the cognitive characteristic, the task type, the quality factor, and the stakeholder. In order to develop a detailed conceptual model of quality factors for knowledge intensive work, several quality factors are elaborated in Section 6. The descriptions of cognitive characteristics as described in [19] are used as input for the conceptual model of cognitive characteristics. Third, it is intended to understand the effects of changes to cognitive characteristics and quality factors during knowledge intensive work and the dependencies between the five main concepts. This has be done by introducing the system dynamics models of Sections 5 and 8. Finally, Section 9 briefly compares our work with other approaches in the field and outlines the benefits of our approach compared to others. Section 10 concludes this paper and gives an overview of future research.

2. Reasoning framework for bridging supply and demand

Fig. 1 shows the reasoning framework for bridging supply and demand for knowledge intensive tasks. The concepts displayed in the figure will be explained in this section. The reasoning framework is split in two layers, namely the application layer and the domain layer. The concepts that are part of the application layer can be positioned in a specific case that involves knowledge intensive work. Examples of such cases are: a request for a mortgage loan, an insurance claim, or a

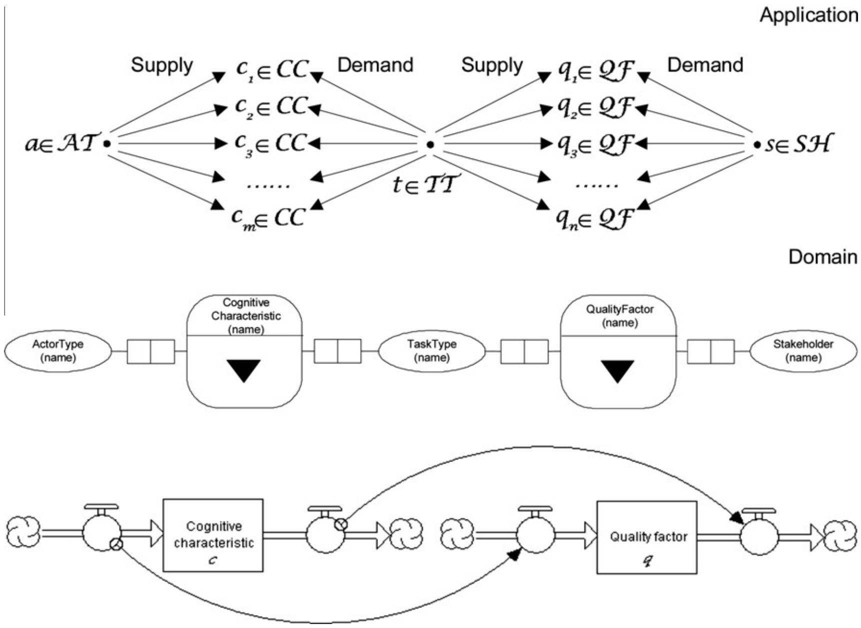

Fig. 1. Reasoning framework for bridging supply and demand.

Please cite this article in press as: S.J. Overbeek et al., Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.002

tax declaration [31]. A domain, also referred to as a *Universe of Discourse*, can be viewed as an umbrella under which applications of a specific type are situated [29]. Typically, tax declarations are part of a tax domain and insurance claims are processed in the insurance domain.

2.1. Cognitive characteristics

Cognitive characteristics function as the bridge between actor types and types of knowledge intensive tasks on the application layer. The set $\mathcal{CC}$ contains cognitive characteristics and the set of task types is denoted as $\mathcal{TT}$. It can be seen that an actor type $a \in \mathcal{AT}$ supplies a certain number of cognitive characteristics that are necessary to perform knowledge intensive tasks. The set $\mathcal{AT}$ contains actor types, which an actor can instantiate. Several actor types that can supply cognitive characteristics at a certain level have been introduced in [19]. For example, one of the types discussed in this earlier work is the *experiencer*. An experiencer is aware of knowledge requirements to fulfill some instance of a knowledge intensive task type. Consider for example the following sentence: *John thoroughly reads an article about balanced scorecards before joining a meeting about balanced scorecards*. This indicates that John, as an experiencer, probably understands that reading an article about balanced scorecards is enough to successfully prepare himself for a meeting about that topic. The cognitive characteristics that are supplied at a certain level by an actor of a certain type are also demanded at a certain level by a task of a certain type. Obviously, these levels of supply and demand differ for each actor type and task type. Several knowledge intensive task types have been defined in [21]. For instance, an actor of the experiencer type can perform a task of the knowledge *acquisition* type. If this is the case, then the *sentience*, *satisfaction*, and *relevance* characteristics are supplied respectively demanded. A knowledge acquisition task is related with the *acquisition* of knowledge. This can be illustrated by a student reading a book in order to prepare himself for an exam. *Sentience* expresses that an actor has much awareness of required knowledge to fulfill some task instance. When a project manager creates a project plan he may have all the necessary knowledge to create such a plan. This may be due to earlier planning experiences of the project manager or by education. *Satisfaction* is related with an actor's need for knowledge during a task's fulfillment and the eventual disappearance of that need. *Relevance* is concerned with whether knowledge acquired is deemed appropriate during the fulfillment of a task or not.

2.2. Quality factors

Besides that cognitive characteristics can be supplied and demanded, quality factors can also be supplied and demanded as is shown in Fig. 1. The set $\mathcal{QF}$ contains quality factors. Recall from Section 1 that it is possible to conceive the quality of the *process* that has led to task fulfillment as well as the quality of the task *result*. This implies that the levels on which a fulfilled task supplies certain quality factors is dependent of the process that led to task fulfillment as well as the resulting product. Those that actually demand on which level the quality factors are supplied by a task of a certain type are the *stakeholders*. The set of stakeholders is denoted as $\mathcal{SH}$. In this case, a stakeholder $s \in \mathcal{SH}$ may be highly concerned about, or have interests in, the process quality factors and the product quality factors of a fulfilled knowledge intensive task. Typical examples of stakeholder types found in the literature are, amongst many others, the supplier, the employee, the customer, and the shareholder [24]. It is assumed that a stakeholder has a *goal* related to the fulfillment of a knowledge intensive task that leads to quality expectations. These quality expectations can be translated to levels on which quality factors supplied by tasks are demanded by a stakeholder. To achieve the desired quality, the required cognitive characteristics should also be supplied at levels that contribute to the achievement of the intended process quality and product quality.

2.3. Coupling between cognition and quality

The actor type, cognitive characteristic, task type, quality factor, and stakeholder concepts can also be discovered in the Object-Role Modeling (ORM) model as part of the domain layer. In such an ORM model, ovals represent object types (which are counterparts of classes), whereas boxes represent relations between object types. These relations are dubbed as fact types. For more details on Object-Role Modeling, see e.g. [6]. By means of ORM, the object types that comprise the five mentioned concepts and the relations between those object types can be visualized. In Fig. 1, it can be seen how the concepts are related with each other. For example, the task type concept is related with the cognitive characteristic concept and the quality factor concept. The task type concept functions as the pivot in these relations. This is caused by the demand of a task type for cognitive characteristics and the supply of quality factors by a fulfilled instance of a task type. Two detailed ORM models related with cognitive characteristics and quality factors are elaborated in Section 4, respectively, Section 7. These models show how actor types, task types, and stakeholders can be related with the concepts of cognitive characteristics and quality factors. This provides insight in the way how object types that conceptualize cognitive characteristics and quality factors are related with each other. These relations provide an understanding of which cognitive characteristics influence which quality factors. To understand how the supply of a certain characteristic influences the supply of a certain quality factor, it is necessary to introduce a more dynamic modeling approach.

Particularly, the effects of changes to the supply and demand levels and the dependencies between cognitive characteristics and quality factors can be modeled by means of system dynamics. System dynamics is a method that is capable of dealing with assumptions about system structures in a stringent fashion [4]. A system dynamics model is constructed by the building blocks (variables) categorized as stocks, flows, connectors, and converters. Stock variables (symbolized by

Please cite this article in press as: S.J. Overbeek et al., Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.002

rectangles) are the state variables and they represent the major accumulations in the system. Flow variables (symbolized by valves) are the rate of change in stock variables and they represent those activities that fill in or drain the stocks. Converters (represented by circles) are intermediate variables used for miscellaneous calculations. Finally, the connectors (represented by simple arrows) are information links representing the cause and effects within the model structure. Fig. 1 visualizes these building blocks of system dynamics. The two valves indicate that there are variables in the domain that may influence the levels on which cognitive characteristics or quality factors are supplied and demanded. Two simple arrows are also pictured. One arrow indicates that the variables that influence the levels on which cognitive characteristics are supplied also influence the levels on which quality factors are supplied. The other arrow expresses such a dependency between the demand levels. Figs. 4 and 8 show detailed system dynamics models of cognitive characteristics and quality factors. These models provide a complete picture of the variables that influence the supply and demand of cognitive characteristics and quality factors.

Recall that a framework for cognitive matchmaking and a prototype of the cognitive matchmaker system have been elaborated in [18,19,22]. The framework has been constructed in such a way that it is also possible to use it for *qualitative matchmaking*, i.e. to match supply and demand of quality factors. Therefore, the framework for cognitive matchmaking can also be adopted to match quality factors without modifying the structure and the formal foundations of the current framework. This extension is briefly discussed in the following section. For an elaborate version of the framework for cognitive matchmaking, see [19].

3. Framework for cognitive and qualitative matchmaking

At first, the framework for qualitative matchmaking is illustrated in Fig. 2. This framework is almost identical to the framework for cognitive matchmaking, but in this case it can match quality factors instead of cognitive characteristics. The different concepts shown in Fig. 2 are functions that are necessary to calculate the eventual total quality match of a fulfilled task. Even though the formal signature of these functions are not exhaustively repeated in this section, we will show some examples for clarification. First, the *supply* function shows the level on which a task type supplies a quality factor. The levels on which a task type supplies a quality factor may vary over the natural numbers from 0 up to and including 10. These levels are part of the *factor rank domain* indicated by the set $\mathcal{FRN}$. This ranking domain includes the rank values that can be used to indicate the level on which a factor can be supplied by a task type or demanded by a stakeholder. The *demand* function depicted in Fig. 2 shows the level on which a stakeholder requires a certain quality factor of a fulfilled task instance.

The factor match or *FacMatch* function shown in Fig. 2 matches supply and demand of a specific factor. There is an optimal factor match if a task offers a quality factor at the same level as a certain stakeholder requires the factor. A factor match is calculated for every quality factor that is supplied by a task type and demanded by a stakeholder. The result is part of the *match rank domain*, which may vary over the real values from 0 up to and including 1. An optimal factor match is indicated by the match rank value 0.5. On the one hand, this is because 0 indicates a task is not able to supply a certain factor at all. On the other hand, a value of 1 indicates that the supply of a factor is not necessary at all for a stakeholder whilst a task supplies that certain factor at the highest level.

The weighed factor match function or *Weigh* function weighes the result of the factor match function. The result is part of the *total quality rank domain*, which may vary over the real values from 0 up to and including 10. The results of the weigh function are then summated by the *Match* function which shows the *total quality match*. This total quality match is also expressed by a value from the total quality rank domain. To show how we have formalized the functions of the framework, the formal signature of, e.g. the match function is modeled as follows [19]:

$$\mathsf{Match} : \mathcal{TT} \times \mathcal{SH} \rightarrow \mathcal{TRN} \tag{1}$$

Note that the set $\mathcal{TT}$ contains task types, the set $\mathcal{SH}$ contains stakeholders and the set $\mathcal{TRN}$ contains total quality rank values. This function can be defined using the aforementioned functions:

$$\mathsf{Match}(\mathtt{synthesis}, \mathtt{customer}) \triangleq \bigoplus_{q \in \mathcal{QF}} \mathsf{Weigh}(q, \mathsf{FacMatch}(\mathtt{synthesis}, \mathtt{customer}))$$

For this example the total quality match of the *synthesis* task type and the *customer* stakeholder type has been calculated. A synthesis task is related with the actual utilization of acquired knowledge. An example is a student who utilizes knowledge (acquired by reading a book) while performing an exam. The definition of the match function shows that for every factor the

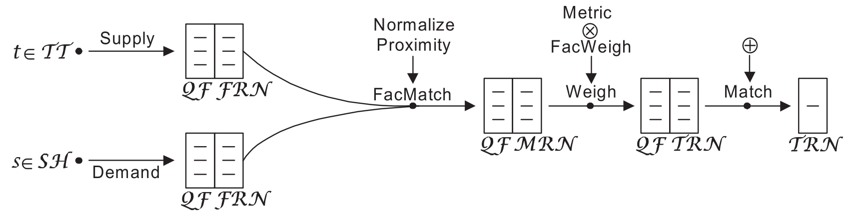

Fig. 2. Framework for qualitative matchmaking.

weighed characteristic match function is executed and the results are then summated. The latter is shown by the $\oplus$ operator. This operator is used instead of the large Sigma because soft (linguistic) suitability rank values can also be used instead of hard (numerical) values. The match function can be expressed as follows: $\text{Match}(\texttt{synthesis}, \texttt{customer}) = \mathbf{4.25}$, which shows that the numerical total quality match of the synthesis task and the quality demanded by the customer is 4.25. This is a fairly good result, knowing that 5 is the best total quality match that can be achieved.

Finally, a function has been introduced to determine the degree of certainty that the total quality expected by the stakeholder is reached [19]:

$$\mu : \mathbb{R} \rightarrow [0, 1] \tag{2}$$

A linear certainty degree function can be defined as follows:

$$\mu(u) \triangleq \begin{cases} \frac{2}{\texttt{min}+\texttt{max}} \cdot u & \texttt{min} \leqslant \texttt{u} \leqslant \frac{\texttt{min}+\texttt{max}}{2} \\ \frac{-2}{\texttt{min}+\texttt{max}} \cdot u + 2 & \frac{\texttt{min}+\texttt{max}}{2} \leqslant u \leqslant \texttt{max} \end{cases}$$

Recall that the result of the total quality may vary over the natural numbers from $\texttt{min} = 0$ up to and including $\texttt{max} = 10$. The degree of certainty that the total quality expected by the customer stakeholder is reached is: $\mu(4.25) = \frac{2}{0+10} \cdot 4.25 = 0.85$ or 85%.

The ORM model and the system dynamics model of cognitive characteristics can be introduced now that the concepts of the application layer have been explained in detail, together with a framework for cognitive and qualitative matchmaking.

4. ORM model of cognitive characteristics

In [19,21] we have created two Object-Role Modeling (ORM) models. These models graphically represent the formal foundations of the cognitive characteristics that characterize the distinguished actor types and task types. The cognitive characteristics have also been formally defined in [19,21]. Based on the characteristics, the different actor types and task types have been characterized. This characterization shows which cognitive characteristics are supplied by an actor of a certain type and demanded by a task of a certain type. To obtain an overall conceptual model of the cognitive characteristics in a domain that contains the distinguished actor types and task types, the two ORM models need to be combined. The merger of both ORM models is visualized in Fig. 3. The formal foundations of, for example, the *volition* characteristic can be introduced in order to explain the ORM model. An actor has the volition characteristic, if an actor has a certain willpower to fulfill some knowledge intensive task instance. It can be said that an actor has a motivation to fulfill a task instance. The level on which an actor has willpower to fulfill a task is incorporated in the volition characteristic. The introduction of a motivation function is necessary to determine an actor's motivation while fulfilling a task instance:

$$\text{Motivation} : \mathcal{AS} \rightarrow (\mathcal{AC} \times \mathcal{TI} \rightarrow \mathcal{MO}) \tag{3}$$

The set $\mathcal{TI}$ contains task instances of a task type. The set $\mathcal{AS}$ contains *actor states*. An actor state is necessary because an actor's motivation might change over time. For example, an actor might be strongly motivated in one state, while an actor might be weakly motivated in another state.

When an actor experiences knowledge, then this will lead to a change in both the actor's knowledge and mood, or, in a state change. State changes caused by anything else than experiencing knowledge are not considered. For example, forgetting knowledge may be seen as a special change of state. Assume $\{\texttt{weak}, \texttt{moderate}, \texttt{neutral}, \texttt{strong}\} \subseteq \mathcal{MO}$. The set $\mathcal{MO}$ includes possible motivation types of an actor. An actor a in a state $t \in \mathcal{AS}$ with a volition characteristic may be weakly, moderately, neutrally or strongly motivated. If an actor a in state t is strongly motivated to fulfill task instance i it can be denoted as: $\text{Motivation}_t(a, i) = \texttt{strong}$. Note that the motivation function cannot be expressed as $\text{Motivation}(t, a, i)$ because of the placed parentheses in the signature of the function. This has been done to indicate that an actor's motivation is coupled with its state. Because of these parentheses, it can be noticed that the *domain* of the motivation function consists of the set of actor states. The *range* of the motivation function consists of a nested total function including the set of actor instances, the set of task instances and the set of motivation types. The volition characteristic is modeled as follows. An actor $a \in \mathcal{AC}$ has the volition characteristic, denoted as $\text{Volition}(a)$, if that actor has a state $t \in \mathcal{AS}$ in which that actor has one of the four motivation types for some task instance to be fulfilled:

$$\exists_{t \in \mathcal{AS}} \exists_{i \in \text{Fulfillment}(a)} [\text{Motivation}_t(a, i) \in \{\texttt{weak}, \texttt{moderate}, \texttt{neutral}, \texttt{strong}\}] \tag{4}$$

The sets and functions as part of the formal definition of the volition characteristic are visualized in the ORM model of Fig. 3. The visual counterparts of sets in the ORM model are object types and the visual counterparts of functions are fact types, sometimes combined with object types. Recall that fact types constitute the relations between object types. The set $\mathcal{AS}$ of actor states is visualized as the object type 'ActorState'. Furthermore, the sets $\mathcal{AC}, \mathcal{TI}, \wp(\mathcal{TI})$, and $\mathcal{MOO}$ are shown as the object types 'ActorInstance', 'TaskInstance', 'TaskInstances' respectively 'MotivationType'. The fulfillment function is visualized by the corresponding fact type. The motivation function is visualized by the object types A, B, C, and the fact type 'Motivation'.

The ORM model of cognitive characteristics can now be used as a basis for the system dynamics model of cognitive characteristics.

Please cite this article in press as: S.J. Overbeek et al., Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.002

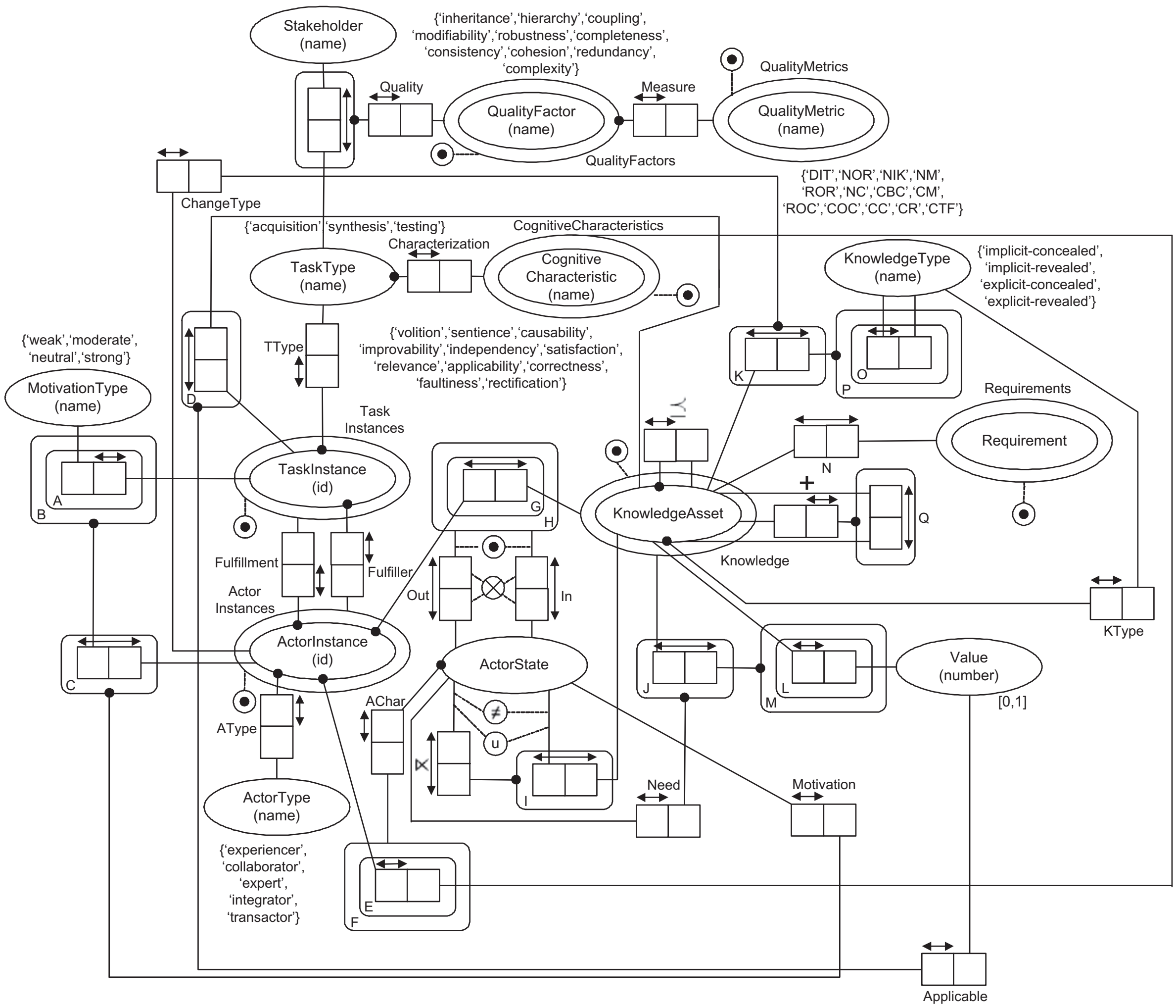

Fig. 3. Object-Role Modeling (ORM) model of cognitive characteristics.

5. System dynamics model of cognitive characteristics

The system dynamics model of cognitive characteristics shown in Fig. 4 can be created using the building blocks mentioned in Section 2 and shown in Fig. 1. The model has been created with the high-level simulation program Stella®. An identical approach to create a system dynamics model for hospital waste management has been elaborated in [4]. The meaning of the system dynamics model can be explained by the system dynamics of, for example, the *volition* characteristic as follows. The motivation level (flow) and the demotivation level (flow) both depend on the motivation type (converter) and the level on which the volition characteristic is applied (stock). The success of the fulfillment of a task also depends on the actor’s supplied volition level. This is indicated by the connector between the volition stock and the ‘Fulfillments’ converter. First, it is possible to model dependencies between the cognitive characteristics by means of the model. Second, the causal links that influence the level on which a cognitive characteristic is applied are modeled. Third, the effects of changes in the model and the relationships between the characteristics can be determined as well.

Now that the cognitive part of the reasoning framework shown in Fig. 1 has been explained, the notion of quality and its role in the framework needs to be elaborated.

6. Quality of knowledge intensive task performance

Recall from Section 1 that a knowledge intensive task is a task for which acquisition, application or testing of knowledge is necessary in order to successfully fulfill the task. The following types were distinguished: The *acquisition* task type, the *synthesis* task type and the *testing* task type. These types are explained below:

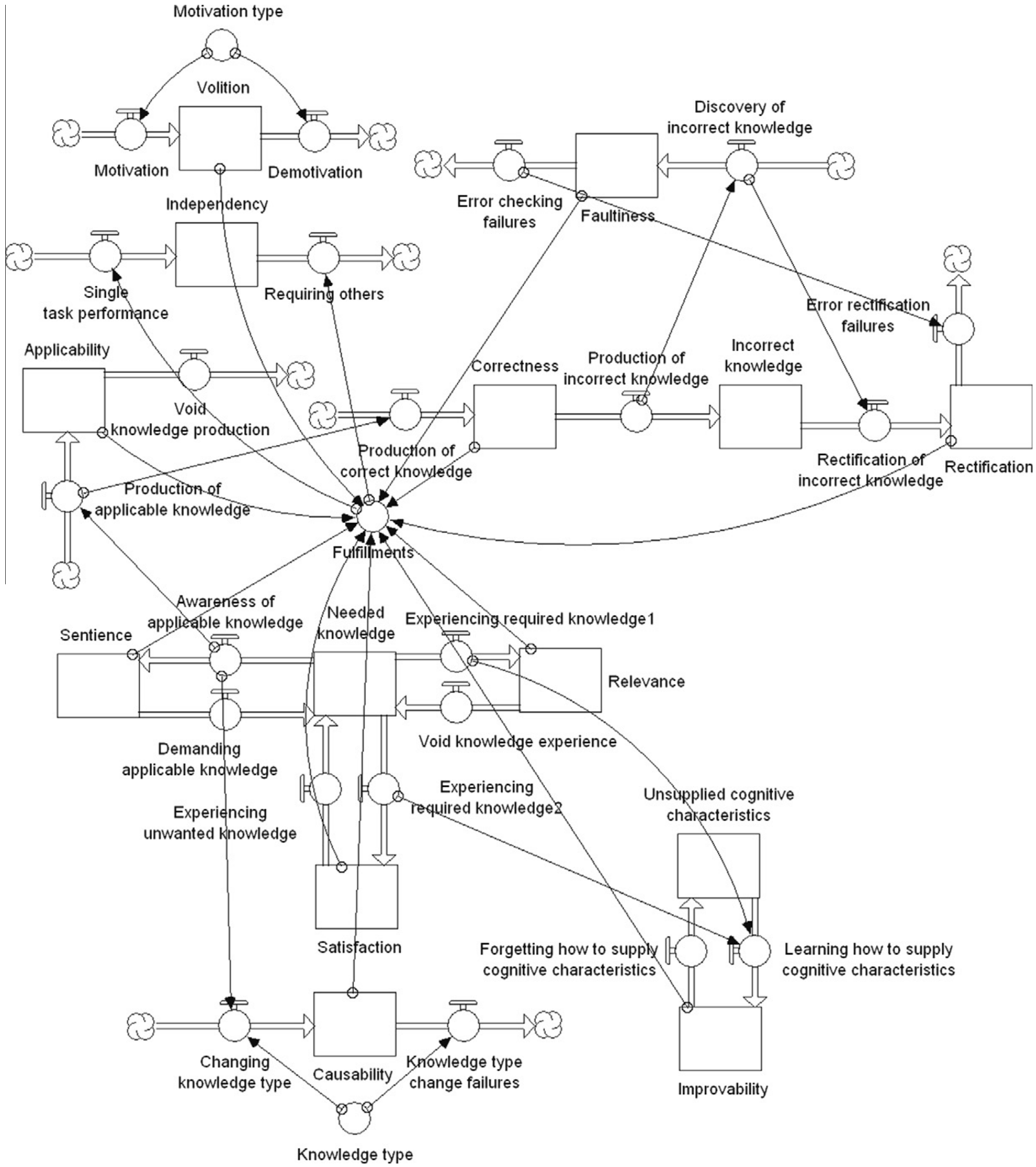

Fig. 4. Structure of the system dynamics model for cognitive characteristics.

(1) Acquisition tasks, which are related with the *acquisition* of knowledge. This can be illustrated by a student reading a book in order to prepare himself for an exam.
(2) Synthesis tasks, which are related with the actual utilization of the acquired knowledge. An example is a student who utilizes knowledge (acquired by reading a book) while performing an exam.
(3) Testing tasks, which are related with the identification and application of knowledge in practice inducing an improvement of the specific knowledge applied. For example a student who failed an exam studies a teacher's feedback on his exam. Then a re-examination attempt follows to improve his previously acquired and utilized knowledge.

An actor may instantiate the task types mentioned above. The results of the task instantiations consist of acquired, utilized or tested knowledge assets. These *assets* are tradeable forms of knowledge, i.e. knowledge that is exchangeable between actors. This may include knowledge obtained by viewing a Web site or a document or by conversing with a colleague. When an instructor explains a learner how to drive a car for instance, the explanation may contain valuable knowledge assets for the learner. The quality of the *representation* of task results and the quality of the *application* of cognitive characteristics during task execution are now discussed. The latter is dubbed as the quality of the *cognitive process* during task execution. The quality of the representation and the cognitive process of this acquired, utilized respectively tested knowledge can be measured by means of the quality factors introduced in [13,14]. Concrete quality measures can be calculated by applying *quality metrics*. It is possible to conceive an ORM model and a system dynamics model of these concrete quality factor metrics. Knowledge representation quality metrics are directly related to the representation of the knowledge assets as a result of a task. Cognitive process quality metrics are related to the cognitive characteristics that are applied to perform a task.

6.1. Knowledge representation quality factor metrics

Firstly, several examples of knowledge representation quality metrics will be discussed. It is possible to represent the structure of the knowledge assets resulting from task fulfillment as a navigation graph [23]. This enables to measure the quality of the representation of the knowledge acquired, applied or tested. As an example, a synthesis task mentioned in the case study in information systems engineering discussed in [19] is considered. In this case study, the framework for cognitive and qualitative matchmaking mentioned in Section 3 and the prototype of the cognitive matchmaker system discussed in [19] are evaluated. This synthesis task, in which knowledge is applied, consists of the design of a *use case* to create a risk report for an 'Action Reporting Tool' (ART for short). The action reporting tool is a Web application that can generate risk reports for the user. This tool should assist risk management to better monitor and control insurance risks. The knowledge assets related to the use case that can be found in the documentation of the ART project can be modeled as a list of strings. Each string represents a knowledge asset:

k_0 Entry.
k_1 The user opens ART and the home page is shown.
k_2 At the home page, the user clicks on the 'create risk report' button.
k_3 ART presents a screen to ask if the user wishes to create a risk report based on an existing report.
  $k_{3.1}$ The user clicks on the 'No' button.
    $k_{3.1.1}$ ART presents a screen to let the user enter risk details.
    $k_{3.1.2}$ The user enters risk details.
    $k_{3.1.3}$ The user clicks on the 'Save' button and returns to the home page.
  $k_{3.2}$ The user clicks on the 'Yes' button.
    $k_{3.2.1}$ ART presents a list of available reports.
    $k_{3.2.2}$ The user clicks on one of the reports.
    $k_{3.2.3}$ ART presents a screen to ask if the user is sure to copy the report.
    $k_{3.2.4}$ The user clicks on the 'Yes' button and continues from $k_{3.1.1}$.
  $k_{3.3}$ The user clicks on the 'Cancel' button and returns to the home page.

Subsequently, the representation of these knowledge assets can be visualized by means of the navigation graph shown in Fig. 5. The arrows in the graph represent hypertext links that enable to navigate through the representation of the nodes. For instance, it is only possible to navigate from k_2 to k_3. After all, the user must click on the 'create risk report' button in order to choose whether or not a new report should be based on an existing report. Several quality metrics adopted from [14] can be used to measure the quality of the knowledge representation. Table 1 contains example metrics comparable to those found

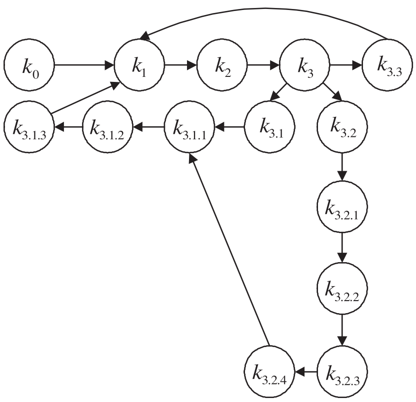

Fig. 5. Navigation graph of nodes containing knowledge.

Please cite this article in press as: S.J. Overbeek et al., Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.002

Table 1
Examples of quality factor metrics for the representation of knowledge.

Quality factor	Factor metric	Name	Description
Inheritance	DIT	Depth of inheritance tree	This metric is a count of the total number of nodes which have edges that exceed or equal three levels relatively to the root of the navigation graph, divided by the total number of nodes
Hierarchy	NOR	Number of root nodes	This metric is a count of the total number of root nodes that have children, divided by the total number of nodes. These children may have more sub children
Coupling	NIK	Number of inherited knowledge	NIK is a measure of the knowledge that a node may inherit from its parent, divided by the total number of nodes
Modifiability	NM	Node modification	NM is a measure of the number of nodes that can potentially be affected in response to modification operations of a node, divided by the total number of nodes
Robustness	ROR	Robustness of representation	The extent to which knowledge contained in the root node can be inherited in sub-nodes without adjustments to the representation, divided by the total number of nodes
Completeness	NC	Node completeness	NC is a measure of the knowledge represented by nodes that have been acquired, applied or tested related to the total number required by the task, divided by the total number of nodes

Table 2
Quality results for the example knowledge representation.

Quality factor	Factor metric	Result for the example	Description
Inheritance	DIT	$\frac{10}{14}$	Node k_0 is the root node of the navigation graph. Nodes k_3 up to and including $k_{3.3}$ are positioned three or more levels deep in the graph. There are 14 nodes in total, of which 10 nodes have edges that are three or more levels deep
Hierarchy	NOR	$\frac{1}{14}$	The example graph has one root node and 14 nodes in total
Coupling	NIK	$\frac{9}{14}$	Coupling can be measured for every node in the graph. Node k_1 can be split in two knowledge-containing parts: The user opens ART and the home page is shown to the user. Only the latter is inherited by node k_2. This implies that 50% of the knowledge of node k_1 is inherited by node k_2. Repeating this exercise for the remaining nodes results in a value of $\frac{9}{14}$ for this metric
Modifiability	NM	$\frac{62}{169}$	The result of this metric is dependent of which node is modified. If node k_1 is modified, this obviously influences the other 13 nodes too. To compute the result of this metric this process is repeated for every other node. Thus, modifying a single node may influence at most 13 other nodes in this case
Robustness	ROR	1	The root node does not contain knowledge that may affect other nodes if it is inherited by sub-nodes. It symbolizes an initial state before the use case is executed. Therefore, this metric results in $\frac{14}{14} = 1$
Completeness	NC	1	In this case, 14 nodes have been created, which comprise the 'create risk report' use case. Assuming that these nodes were all required, this metric results in $\frac{14}{14} = 1$

in [14]. The example metrics of Table 1 are used to measure the knowledge representation quality of the mentioned synthesis task. The graph of Fig. 5 can be used as input to determine the representation quality. Table 2 contains the resulting values for the knowledge representation quality metrics.

So far, only quality factors and metrics have been introduced to measure the quality of the knowledge representation as a result of task fulfillment. What remains are the factors and metrics to measure the quality of the process during which cognitive characteristics have been applied while executing a task.

6.2. Cognitive process quality factor metrics

Besides distinguishing several knowledge representation quality metrics, it is possible to introduce example quality metrics for the process leading to a task result. When viewing this topic from a cognitive standpoint, the quality of the application of cognitive characteristics during task execution can be considered. Simply put, an actor applies several cognitive characteristics during task execution. This application may vary in quality dependent of how well an actor is able to supply them. Table 3 contains the list of *cognitive process* metrics. When applying the metrics shown in Table 3 to the *use case* example it is necessary to know which cognitive characteristics have been applied during the fulfillment of the task to create the use case. Remember that the task instance to create a use case can be abstracted as a *synthesis* task type. The case study discussed in [19] reveals that this task is performed by an actor of the *expert* type. A total of seven characteristics are applied when an expert performs a synthesis task. These are the volition, sentience, causability, improvability, independency, applicability, and correctness characteristics. These characteristics are discussed in detail in [19,21], but they can be briefly described as follows:

- The *volition* characteristic is concerned with an actor's willpower to fulfill some knowledge intensive task instance. For instance, a skilled software developer may have more willpower to implement an intelligent search algorithm than implementing source code to access a database.

Table 3
Examples of quality factor metrics for the application of cognitive characteristics.

Quality factor	Factor metric	Name	Description
Coupling	CBC	Coupling between characteristics	This is a measure of the number of characteristics whose application depends on the application of other characteristics, divided by the total number of characteristics
Modifiability	CM	Characteristic modification	This is a measure of the number of applied characteristics that are potentially influenced in response to the modification of the level on which another characteristic is applied
Consistency	COC	Consistency of characteristics	This is a count of the characteristics that have been insufficiently applied during task execution, divided by the total number of characteristics
Robustness	ROC	Robustness of characteristics	In this case, robustness is the number of applied cognitive characteristics that mismatch the level on which they are demanded. This number is then divided by the total number of characteristics
Cohesion	CC	Characteristic cohesion	This is the degree of relatedness between applied characteristics. It can be measured through counting the number of dependencies in the system dynamics model of cognitive characteristics, divided by the total number of dependencies
Redundancy	CR	Characteristic redundancy	A redundant characteristic is a count of the applied characteristics that can have the same effect on the outcome of the task result, divided by the total number of characteristics
Complexity	CTF	Characteristic complexity	CTF is a count of those applied characteristics that are defined by using three or more functions, divided by the total number of characteristics

- *Sentience* expresses that an actor has much awareness of required knowledge to fulfill some task instance. When a project manager creates a project plan he may have all the necessary knowledge to create such a plan. This may be due to earlier planning experiences of the project manager or by education.
- The *causability* characteristic expresses that an actor has the ability to exert an influence on state changes of knowledge involved during fulfillment of a task instance. Suppose that a business consultant facilitates a brainstorm session in which he or she writes models on a whiteboard. In this case, the consultant causes knowledge that is implicitly present in his or her head to be made explicit on the white board [15]. This means that the consultant *causes* knowledge to change from one type to another.
- During fulfillment of certain knowledge intensive task instances an actor should be able to improve its own cognitive abilities. This is indicated by the *improvability* characteristic. For instance, a manager may have recently completed a course about cybernetics. During his work he or she successfully applies several principles that the manager has learned in the course. Participating in the course may thus have improved his or her cognitive abilities.
- The *independency* characteristic is necessary to be able to determine if an actor is able to fulfill a task instance on his own or not. An example is a journalist who may successfully write a news article without having to collaborate with others.
- The *applicability* characteristic expresses to what extent knowledge is applicable in a task.
- When knowledge is applied it should meet its requirements. This is indicated by the *correctness* characteristic.

6.3. Results for the application of cognitive characteristics

Now that the applied cognitive characteristics are known it is possible to compute the results of the cognitive process quality metrics. However, there are three aspects that are required to compute the results:

(1) The formal definitions of the cognitive characteristics.
(2) Supply and demand levels of the cognitive characteristics.
(3) The system dynamics model of the cognitive characteristics.

The cognitive characteristics have been formally defined in [19,21]. These formalisms are necessary to compute a result for the redundancy and complexity metrics. To compute a result for the redundancy metric the formalisms are compared with each other to determine if application of the characteristics have equal effects on task fulfillment. Counting the number of functions that have been used to formally define a characteristic is necessary to compute a result for the complexity metric.

The second aspect that is mandatory to determine results for the metrics is related to the levels on which the cognitive characteristics are demanded by the task respectively supplied by the actor. Based on the framework for cognitive and qualitative matchmaking, the cognitive characteristics supplied by actors respectively demanded by tasks can be matched by using the prototype of the cognitive matchmaker system [19]. The prototype also contains the levels on which characteristics are supplied and demanded. In the case of the use case example, we would like to know the levels on which the seven aforementioned characteristics are supplied and demanded. This is shown in Fig. 6, which is a screen shot of the prototype. The screen shot shows the levels on which the expert supplies the seven characteristics and the synthesis task demands these characteristics. In the prototype, the levels can be expressed by values from 0 up to and including 10. A level of 0 shows that a characteristic is supplied or demanded at the minimum level. A level of 10 shows that a characteristic is supplied or demanded at the maximum level. Results for the modifiability, consistency, and robustness metrics can be determined by taking these levels into account.

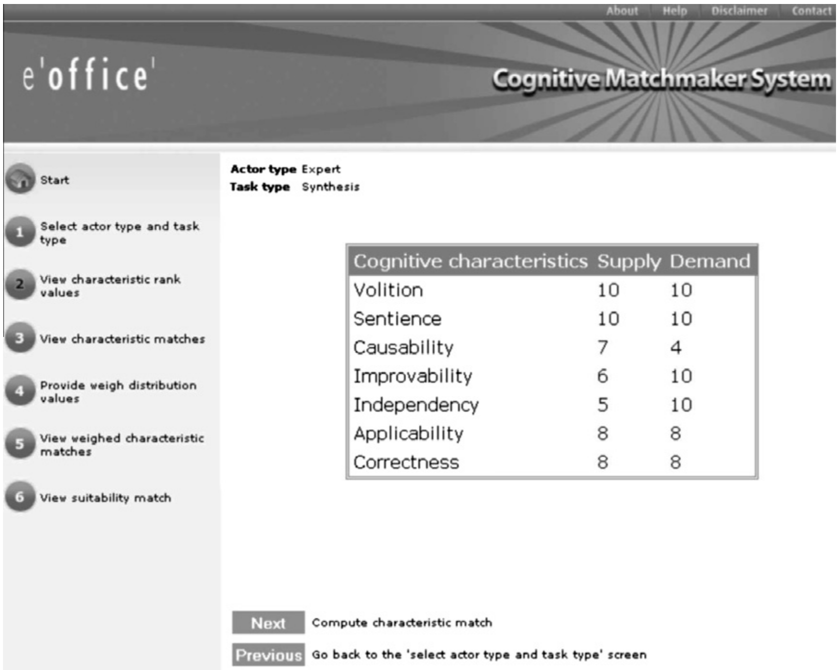

Fig. 6. Cognitive characteristic levels.

The third aspect is related to the system dynamics model of cognitive characteristics shown in Fig. 4. The system dynamics model for cognitive characteristics is required to determine the coupling, modifiability, and cohesion metrics. Fig. 4 shows the system dynamics model for the cognitive characteristics. Note that more cognitive characteristics are shown than the seven characteristics mentioned up till now. This is because other actor types or task types may supply respectively demand additional characteristics. Regarding the use case example, the results for the cognitive process quality metrics are shown in Table 4, together with a description of the results. The complete suite of the discussed quality factors and metrics are summarized in Table 5. The quality factor metrics can be used to create an ORM model and a system dynamics model of quality factors.

7. ORM model of quality factor metrics

The quality factor metrics of Tables 1 and 3 can be visualized by means of an ORM model, which is shown in Fig. 7. After all, the quality factor metrics provide a concrete and measurable implementation of the quality factors. A stakeholder that is

Table 4
Quality results for the example application of cognitive characteristics.

Quality factor	Factor metric	Result for the example	Description
Coupling	CBC	$\frac{5}{7}$	The system dynamics model for cognitive characteristics shows that five out of the seven applied characteristics depend on the application of other characteristics. Volition and independency are not directly dependent of other characteristics
Modifiability	CM	$\frac{34}{42}$	Modifying the level of a characteristic may influence the other six characteristics of the example. It has to be determined if there is any impact on the other characteristics when modifying the level of one of the characteristics
Consistency	COC	$\frac{2}{7}$	The improvability characteristic is supplied at level 6 by the expert whilst it is demanded at level 10 by the synthesis task. The independency characteristic is supplied at level 5 by the expert whilst it is demanded at level 10 by the synthesis task. This implies that only these two characteristics are insufficiently applied
Robustness	ROC	$\frac{3}{7}$	Causability, improvability, and independency are applied at a level that does not equal the demand level
Cohesion	CC	$\frac{8}{24}$	A total of eight dependencies between characteristics can be found in the system dynamics model
Redundancy	CR	0	Regarding the formal definitions of the characteristics, it is not possible that there are equal effects on task fulfillment
Complexity	CTF	$\frac{5}{7}$	The volition and independency characteristics are the only characteristics that have been defined by using less than three functions

Table 5
Summary of example quality factors and metrics.

Quality factor	Knowledge representation metric	Cognitive process metric
Inheritance	DIT	–
Hierarchy	NOR	–
Coupling	NIK	CBC
Modifiability	NM	CM
Robustness	ROR	ROC
Completeness	NC	–
Consistency	–	COC
Cohesion	–	CC
Redundancy	–	CR
Complexity	–	CTF

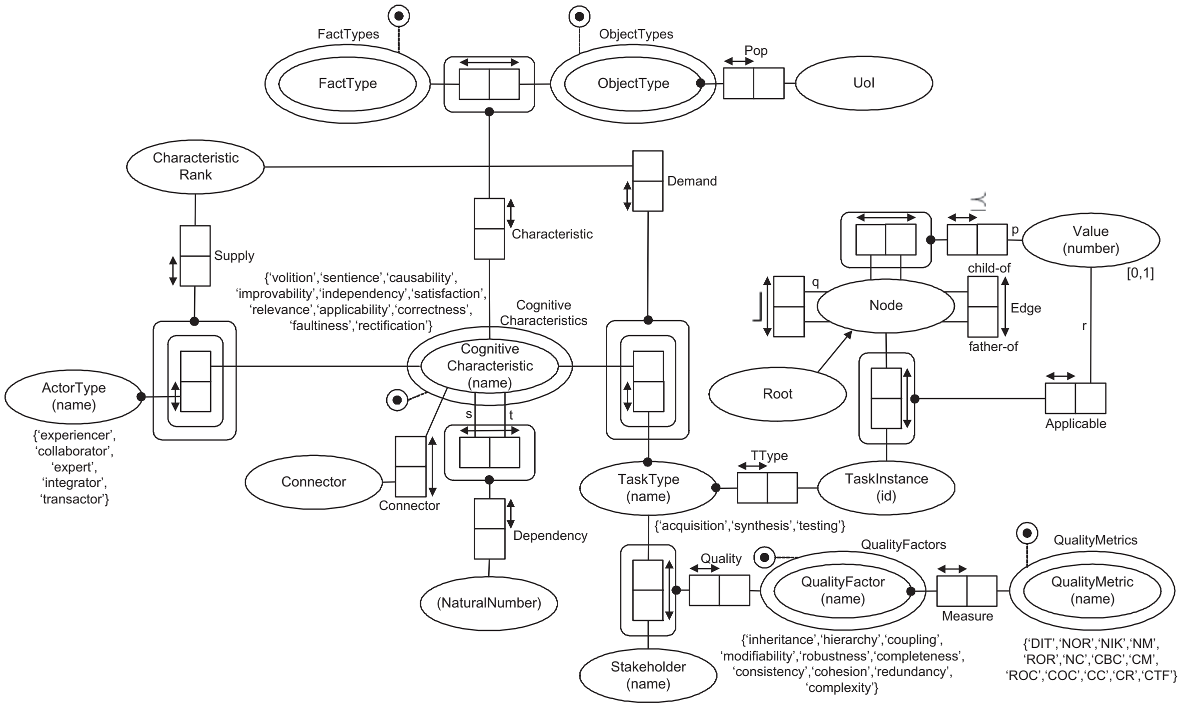

Fig. 7. Object-Role Modeling (ORM) model of quality factor metrics.

concerned about, or has interests in a task of a certain type may express these concerns or interests in terms of quality factors. The quality factors that are deemed important by a stakeholder for a task of a certain type can be expressed by the quality function:

$$\text{Quality} : \mathcal{SH} \times \mathcal{TT} \rightarrow \wp(\mathcal{QF}) \quad (5)$$

Assume that $\texttt{customer} \in \mathcal{SH}$, $\texttt{synthesis} \in \mathcal{TT}$, and also assume that:

$$\{\texttt{robustness}, \texttt{cohesion}\} \subseteq \mathcal{QF}$$

The expression $\text{Quality}(\texttt{customer}, \texttt{synthesis}) = \{\texttt{robustness}, \texttt{cohesion}\}$ indicates that the customer stakeholder is interested in the robustness and cohesion quality factors for a task of the synthesis type. The quality factor metrics that can concretely measure a certain quality factor can be found by the measure function:

$$\text{Measure} : \mathcal{QF} \rightarrow \wp(\mathcal{QM}) \quad (6)$$

The set $\mathcal{QM}$ contains quality factor metrics. For instance, the expression $\text{Measure}(\texttt{robustness}) = \{\texttt{ROR}, \texttt{ROC}\}$ shows that the robustness factor can be measured by the ROR and ROC metrics. The quality and measure functions have also been visualized in the ORM model of quality factors.

The ORM models of Figs. 3 and 7 together comprise an overall conceptual model to bridge supply and demand for knowledge intensive tasks. The main concepts can be found in both detailed models, namely: Actor type, cognitive characteristic, task type, quality factor, and stakeholder. Notice that the ORM model of the cognitive characteristics is more complex than the ORM model of quality. There are several reasons for this observation. At first, the quality factor metrics related to the result of a fulfilled task are for a large part described by the node and root concepts. This curbs the introduction of additional concepts. Next, the cognitive process metrics can be measured by introducing a few additional functions, such as: Connector, dependency, supply, and demand. However, these functions do not require a large number of concepts to be introduced. In contrast, the description of the cognitive characteristics themselves are relatively complex. This requires the visualization of many fact types and object types as is shown in Fig. 3. The complexity of Fig. 3 is caused by the diversity of all possible cognitive characteristics. Each characteristic can be seen a unique part of the cognitive system of an actor. The quality factors are not that diverse, because they can only be related to the determination of the process or product quality. Also note that the task type concept functions as the pivot between the cognitive characteristic and quality concepts in both detailed ORM models. Translated to the reasoning framework of Fig. 1, this is caused by the demand of a task type for cognitive characteristics and the supply of quality factors by a fulfilled instance of a task type. Eventually, the elaboration of the metrics that has enabled the creation of the ORM model for quality can be found in the next sections.

7.1. Depth of inheritance tree metric

First, recall that the DIT metric of Table 1 is a count of the total number of nodes in a graph-based knowledge representation which have edges that exceed three levels relatively to the root of the graph, divided by the total number of nodes. To visually model this metric, it is necessary to introduce object types ‘Root’, ‘Node’, and a fact type ‘Edge’. This fact type should consist of a role ‘child-of’ and a role ‘father-of’. These object types and the fact type are visualized in Fig. 7. Formally, the fact type can be modeled as follows:

$$\text{Edge} \subseteq \mathcal{ND} \times \mathcal{ND} \tag{7}$$

Note that the set $\mathcal{ND}$ contains nodes. Fig. 7 shows that an arrow is drawn from object type ‘Root’ to object type ‘Node’. This denotes that there exists a *specialization relation* between a subtype and a supertype [29]. Such a relation implies that the instances of the subtype are also instances of the supertype (each ‘Root’ is also a ‘Node’). For proper specialization, it is required that subtypes be defined in terms of one or more of their supertypes. Such a decision criterion is referred to as a *Subtype Defining Rule* [30]. The subtype defining rule for ‘Root’ can be expressed as:

$$\text{Root} = \text{Node}(\text{child-of} = \emptyset)$$

Next, a *path expression* is required to determine which parts of the ORM model of Fig. 7 represent the DIT metric. Path expressions are constructs for expressing derived fact types closely following the underlying information structure [29]. Path expressions can be constructed from elements of the information structure, such as roles and object types, and a number of operators. In its elementary form, a path expression corresponds to a path through the information structure, starting and ending in an object type. The following path expression identifies the nodes that are required for the DIT metric (i.e. nodes that are at least three levels deep in the graph):

Node child-of Node child-of Node child-of Node

The path expression starts in object type ‘Node’. All the nodes that are at least three levels deep in a graph with a root node can easily be found by repeating the expression ‘child-of Node’ three times. The path expression will also end in the object type ‘Node’ again. The DIT metric can be expressed as follows:

$$\text{DIT} = \frac{|\text{Node child-of Node child-of Node child-of Node}|}{|\mathcal{ND}|} \tag{8}$$

The number of nodes that are at least three levels deep in the graph are now divided by the total number of nodes. Note that the pipe symbols denote the cardinality. In other words, they are used for counting the number of elements in the numerator of the fraction as well as in the denominator. These elements can be, for example, instances of an object type or instances of a set. There are 10 nodes that are positioned three or more levels deep in the graph in case of the knowledge representation shown in Fig. 5. There are 14 nodes in total in case of that representation. This results in $\text{DIT} = \frac{10}{14}$.

7.2. Number of root nodes metric

For this metric the number of root nodes that have children need to be counted. This count then needs to be divided by the total number of nodes. The following path expression results in those root nodes that have children:

Root father-of Node

Subsequently, the NOR metric can be modeled as follows:

$$\text{NOR} = \frac{|\text{Root father-of Node}|}{|\mathcal{ND}|} \tag{9}$$

The cardinality of root nodes that have children is now divided by the cardinality of the set of nodes. This leads to the number of instances of the 'Root' object type that have children divided by the number of instances as part of the set of nodes. There is 1 root node in Fig. 5 and 14 nodes in total. This results in $\text{NOR} = \frac{1}{14}$.

7.3. Number of inherited knowledge metric

NIK is a measure of the knowledge that a node may inherit from its parent node. A knowledge containment function has been introduced in [21] to indicate that knowledge contained in a node is also contained in another node. This function can be slightly extended to determine the knowledge containment percentage. The description of the example result for the NIK metric in Table 2 shows that the knowledge of node x is contained within node y. The notation $x \preceq y = 0.5$ is verbalized as *the knowledge in node x is contained within node y for 50%* and is modeled by the function:

$$\preceq : \mathcal{ND} \times \mathcal{ND} \to [0, 1] \tag{10}$$

Note that the infix notation is used, which is semantically equivalent to the prefix notation: $x \preceq y \equiv \preceq(x, y)$. The results of the knowledge containment function need to be summated to determine the total amount of knowledge containment in a knowledge representation. This total amount needs to be divided by the total number of nodes. Notice that the ORM model of Fig. 7 includes a role p, which contains all the knowledge containment values. The population of elements in an ORM model, such as instances of object types, fact types, and roles can be found by applying the population function introduced in [30]. This function can be referred to as Pop and can be modeled as follows:

$$\text{Pop} : \mathcal{OT} \to \Omega \tag{11}$$

The set $\mathcal{OT}$ contains object types. Note that roles and fact types are subsets of object types. The set Ω can be referred to as the *Universe of Instances*, abbreviated to UoI. The Universe of Instances contains all possible instances of types found in an ORM model. Regarding the example result for the NIK metric depicted in Table 2, it can be concluded that $9 \in \text{Pop}(p)$. The NIK metric can be formalized as follows:

$$\text{NIK} = \frac{\sum_{x \in \text{Pop}(p)} x}{|\mathcal{ND}|} \tag{12}$$

Every element of the population of role p is summated and divided by the total number of nodes. Recall from Table 2 that $\text{NIK} = \frac{9}{14}$ for the example representation.

7.4. Node modification metric

NM is a measure of the number of nodes that can potentially be affected in response to modification operations of a node. This result is then divided by the total number of nodes that can be affected. Modifying one certain node may affect the remaining nodes. This implies that every other node needs to be examined every time if one certain node is modified. The notation $x \rfloor y$ denotes that a modification operation on node x affects node y. This modification function can be modeled as follows:

$$\rfloor \subseteq \mathcal{ND} \times \mathcal{ND} \tag{13}$$

Notice that the ORM model of Fig. 7 includes a role q, which contains the node instances that are affected by modifications to other nodes. Using the population function, the NM metric can be formalized as follows:

$$\text{NM} = \frac{|\text{Pop}(q)|}{(|\mathcal{ND}| - 1)^2} \tag{14}$$

The number of affected nodes in role q is now divided by the total number of nodes that can be affected. This results in $\text{NM} = \frac{62}{169}$ for the example representation.

7.5. Robustness of representation metric

The extent to which knowledge contained in the root node can be inherited in sub-nodes without adjustments to the representation can be determined by this metric. This result should also be divided by the total number of nodes. By means of the knowledge containment function of Section 7.3, it is possible to determine to what extent knowledge can be inherited by sub-nodes. The modification function can be used to determine to what extent a node is affected due to knowledge inheritance. Those nodes that may inherit knowledge from the root node without causing changes to the representation are part of the set $\mathcal{NR}$. This set can be defined as follows:

$$\mathcal{NR} = \{ y \in \mathcal{ND} | \exists_{x \in \mathcal{RT}} \forall_{i \in \mathcal{ND}} [x \preceq y > 0 \wedge (y, i) \notin \rfloor] \} \tag{15}$$

Note that the set $\mathcal{RT}$ is the set of root nodes. The ROR metric can be formalized by dividing the cardinality of the set $\mathcal{NR}$ with the cardinality of the set $\mathcal{ND}$:

$$\text{ROR} = \frac{|\mathcal{NR}|}{|\mathcal{ND}|} \tag{16}$$

The numerator shows that the knowledge contained in a root node *x* is inherited by a sub node *y*. This modification (i.e. the inheritance of certain knowledge) should not affect a random sub node *i*. Applying the ROR metric on the example representation results in $\text{ROR} = \frac{14}{14} = 1$.

7.6. Node completeness metric

NC is a measure of nodes containing knowledge that has been acquired, applied, or tested related to the total amount of knowledge in nodes that is applicable for a task. Thirteen nodes are part of the example representation of Fig. 5. The level on which knowledge is applicable for a task can be measured by the following function [21]:

$$\text{Applicable} : \mathcal{TI} \times \mathcal{ND} \rightarrow [0, 1] \tag{17}$$

The set $\mathcal{TI}$ contains task instances. Those nodes that contain applicable knowledge are part of the set $\mathcal{AK}$. This set can be defined as follows:

$$\mathcal{AK} = \{x \in \text{Pop}(r) | x > 0\} \tag{18}$$

Role *r* as part of the ORM model of Fig. 7 contains the applicability values of node instances. Those instances of role *r* need to be counted that equate to values greater than 0. Results for the NC metric can be computed by dividing the number of nodes that contain applicable knowledge with the total number of nodes in a graph:

$$\text{NC} = \frac{|\mathcal{AK}|}{|\mathcal{ND}|} \tag{19}$$

Applying this metric on the example navigation graph of Fig. 5 results in $\text{NC} = \frac{14}{14} = 1$.

7.7. Coupling between characteristics metric

This metric is a measure of the number of characteristics whose application depends on the application of other characteristics, divided by the total number of characteristics. Results for this metric can be found by means of the system dynamics model of cognitive characteristics. Dependencies between characteristics are indicated by the connectors in the system dynamics model. If a cognitive characteristic is dependent of another this can be modeled as follows:

$$\text{Dependency} : \mathcal{CC} \times \mathcal{CC} \rightarrow \mathbb{N} \tag{20}$$

The system dynamics model of Fig. 4 shows that the applicability characteristic is dependent of the sentience characteristic. This can be expressed as follows: $\text{Dependency}(\texttt{applicability}, \texttt{sentience}) = 1$, which indicates that there is exactly one dependency between the two characteristics. Results for the CBC metric can be computed by the following function:

$$\text{CBC} = \frac{|\text{Pop}(s)|}{|\mathcal{CC}|} \tag{21}$$

Role *s* includes the instances of the set of cognitive characteristics that are dependent of other characteristics. The population of role *s* contains five instances in case of the process leading to the example representation. Subsequently, the metric results in $\text{CBC} = \frac{5}{7}$.

7.8. Characteristic modification metric

This metric is a measure of the number of applied characteristics that are potentially influenced in response to the modification of the level on which another characteristic is applied. The dependency equation can be used to find out which characteristics are influenced when a supply level of a characteristic is changed. The total population of role *t* in Fig. 7 includes the cognitive characteristics which other characteristics depend on. The characteristics as part of role *t* may, therefore, be potentially influenced. Results for the CM metric can be calculated by the following function:

$$\text{CM} = \frac{|\text{Pop}(t)|}{|\mathcal{CC}|^2 - |\mathcal{CC}|} \tag{22}$$

When applying this metric on the example cognitive process, role *t* contains 34 instances of characteristics. Dividing this by the total number of characteristics results in $\text{CM} = \frac{34}{42}$.

Please cite this article in press as: S.J. Overbeek et al., Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.002

7.9. Consistency of characteristics metric

This metric is a count of the characteristics that have been insufficiently applied during task execution, divided by the total number of characteristics. The framework for cognitive and qualitative matchmaking that has been introduced in Section 3 includes a supply function to show on what level a, in this case, cognitive characteristic is supplied by an actor. The signature of this supply function can be modeled as follows [22]:

$$\mathsf{Supply} : \mathcal{AT} \rightarrow (\mathcal{CC} \rightarrow \mathcal{CRN}) \tag{23}$$

The set $\mathcal{CRN}$ includes values from the *characteristics rank domain*. These values are natural numbers that vary from 0 up to and including 10. The expression $\mathsf{Supply}_{\texttt{expert}}(\texttt{volition}) = 10$ shows that an actor characterized by the expert type offers the volition characteristic and is capable to perform this characteristic at level 10. A demand function that returns a value expressing to what extent a cognitive characteristic is *required* for a certain task can be modeled as follows:

$$\mathsf{Demand} : \mathcal{TT} \rightarrow (\mathcal{CC} \rightarrow \mathcal{CRN}) \tag{24}$$

The expression $\mathsf{Demand}_{\texttt{synthesis}}(\texttt{volition}) = 10$ indicates that the volition characteristic is required at the highest level in order to fulfill a task of the synthesis type. Supply and demand levels of the characteristics that are involved when matching an expert actor and a synthesis task are shown in Fig. 6. Results for the COC metric can be computed by first counting the number of characteristics that are supplied on a level that is lower than the demand level. For the example in Table 4 this was the case for 2 out of 7 characteristics in total. This count should then be divided by the total number of characteristics. The characteristics that have been insufficiently applied are part of the set $\mathcal{IC}$. This set can be defined as follows:

$$\mathcal{IC} = \{x \in \mathcal{CC} | \exists_{i \in \mathcal{AT}} \exists_{j \in \mathcal{TT}} [\mathsf{Supply}_i(x) < \mathsf{Demand}_j(x)]\} \tag{25}$$

As such, the formula for the COC metric can be modeled as follows:

$$\mathrm{COC} = \frac{|\mathcal{IC}|}{|\mathcal{CC}|} \tag{26}$$

Fig. 6 shows that two characteristics are insufficiently supplied in case of the example cognitive process. This leads to a metric result of $\mathrm{COC} = \frac{2}{7}$.

7.10. Robustness of characteristics metric

The ROC metric is the number of applied cognitive characteristics that mismatch the level on which they are demanded. The formula to compute results for this metric can almost be equated to that of the COC metric. However, the numerator should provide the number of those characteristics that do not have equal supply and demand levels. The characteristics that have unequal supply and demand levels are part of the set $\mathcal{UC}$. This set can be defined as follows:

$$\mathcal{UC} = \{x \in \mathcal{CC} | \exists_{i \in \mathcal{AT}} \exists_{j \in \mathcal{TT}} [\mathsf{Supply}_i(x) \neq \mathsf{Demand}_j(x)]\} \tag{27}$$

The following equation can be used to calculate with the ROC metric:

$$\mathrm{ROC} = \frac{|\mathcal{UC}|}{|\mathcal{CC}|} \tag{28}$$

Also note that $\mathcal{IC} \subseteq \mathcal{UC}$. The set $\mathcal{UC}$ contains three characteristics in case of the example cognitive process. This leads to a metric result of $\mathrm{ROC} = \frac{3}{7}$.

7.11. Characteristic cohesion metric

This metric is the degree of relatedness between applied characteristics. It can be measured through counting the number of dependencies in the system dynamics model of cognitive characteristics, divided by the total number of dependencies. Recall that these dependencies are also dubbed as connectors (represented by simple arrows) and represent the cause and effects within the model structure. The set $\mathcal{CN}$ includes the connectors of a system dynamics model. A connector that is related to a cognitive characteristic can be modeled as follows:

$$\mathsf{Connector} \subseteq \mathcal{CC} \times \mathcal{CN} \tag{29}$$

Assume that $c \in \mathcal{CC}$ and $i \in \mathcal{CN}$. The expression $(c, i) \in \mathsf{Connector}$ denotes that connector i is related to cognitive characteristic c. The following formula can be used to calculate with the CC metric:

$$\mathrm{CC} = \frac{\sum_{x,y \in \mathcal{CC}} \mathsf{Dependency}(x, y)}{|\mathcal{CN}|} \tag{30}$$

This formula can be used to summate the number of dependencies between the cognitive characteristics shown in Fig. 4. This summation is divided by the total number of connectors in the system dynamics model of cognitive characteristics. In case of the example cognitive process this metric results in $\mathrm{CC} = \frac{8}{24}$.

Please cite this article in press as: S.J. Overbeek et al., Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.002

7.12. Characteristic redundancy metric

The characteristic redundancy metric is a count of the applied characteristics that can have the same effect on the outcome of the task result, divided by the total number of characteristics. Fig. 3 shows the ORM model of the cognitive characteristics, which includes all object types and fact types that formally define the characteristics. Using this ORM model, it is possible to determine which object types and fact types formalize a characteristic. This can be done by using the following equation:

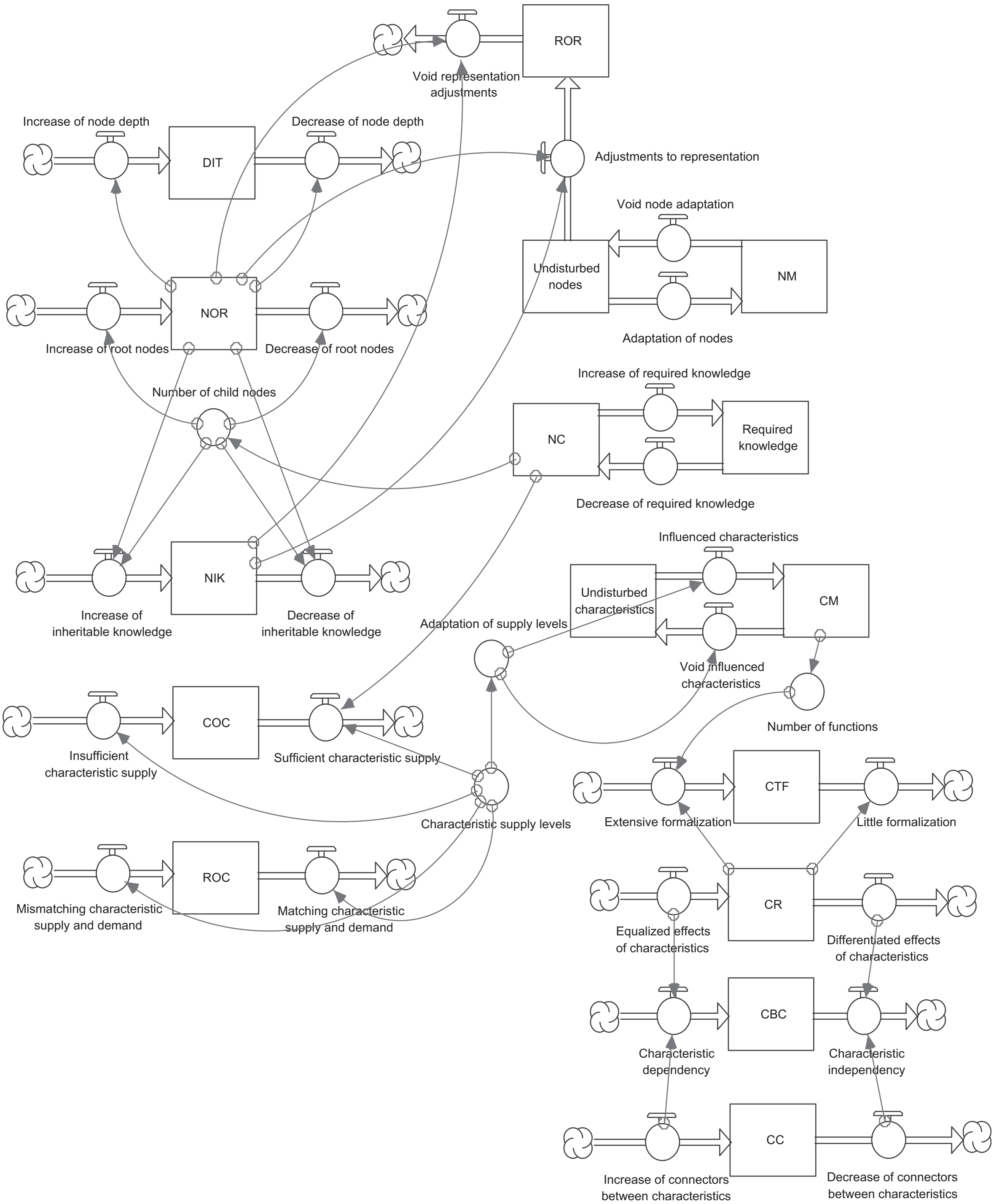

Fig. 8. Structure of the system dynamics model for quality factor metrics.

Please cite this article in press as: S.J. Overbeek et al., Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.002

$$\text{Characteristic} : \wp(\mathcal{OT}) \times \wp(\mathcal{FT}) \rightarrow \mathcal{CC} \tag{31}$$

For instance, the following object types and fact types constitute the volition characteristic:

$$\text{Characteristic}(\{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{AC}, \mathcal{TI}, \wp(\mathcal{TI}), \mathcal{AS}, \mathcal{MO}\}, \{\text{Fulfillment}, \text{Motivation}\}) = \texttt{volition}$$

The set $\mathcal{EC}$ contains those characteristics that are formalized by the same object types and fact types. This set can be defined as follows:

$$\mathcal{EC} = \{z \in \mathcal{CC} | \exists_{I,X \subseteq \mathcal{OT}} \exists_{J,Y \subseteq \mathcal{FT}} [z = \text{Characteristic}(I, J) = \text{Characteristic}(X, Y)]\} \tag{32}$$

Eventually, the formalization of the CR metric can be modeled as follows:

$$\text{CR} = \frac{|\mathcal{EC}|}{|\mathcal{CC}|} \tag{33}$$

The set of equally formalized characteristics contains 0 instances in case of the example cognitive process. There are no characteristics that are formalized by the same object types and fact types. This implies CR = 0.

7.13. Characteristic complexity metric

The final metric is a count of those applied characteristics that are defined by using three or more functions, divided by the total number of characteristics. The characteristic function can also be used to compute results for this metric. In fact, the powerset of fact types as part of the characteristic function include the functions that have been used to formalize a cognitive characteristic. The set $\mathcal{DC}$, which includes those characteristics that are defined by using three or more functions, can be defined as follows:

$$\mathcal{DC} = \{\text{Characteristic}(X, Y) | X \subseteq \mathcal{OT}, Y \subseteq \mathcal{FT}, |Y| \geqslant 3\} \tag{34}$$

This leads to the following equation that can be used to calculate with the CTF metric:

$$\text{CTF} = \frac{|\mathcal{DC}|}{|\mathcal{CC}|} \tag{35}$$

Five characteristics are defined by using three or more functions in case of the example cognitive process. Therefore, the CTF metric results in $\frac{5}{7}$.

Table 6
Dependencies between cognitive characteristics and quality factors.

Quality factor metric	Dependent cognitive characteristics	Description
DIT	Applicability Correctness Rectification	The DIT metric is related with an increase of nodes in a graph-based knowledge representation. The supply levels of the applicability, correctness, and rectification characteristics influence the DIT quality. The more knowledge is produced or rectified, the more the DIT quality level increases and vice versa
NOR	Applicability Correctness Rectification	The NOR metric depends on the same characteristics as the DIT metric. The more knowledge is produced or rectified, the more the NOR quality level increases and vice versa
NIK	Applicability Causability Correctness Rectification	The NIK metric depends on the amount of knowledge production and rectification. It is also dependent of the causability characteristic. This is because a certain knowledge type may be more suitable for inheritance than another
NM	Causability Rectification	The NM metric is dependent of the adaptation of nodes in a graph-based knowledge representation. The causability and rectification characteristics can, therefore, influence this metric
ROR	Causability Rectification	Adjustments to an overall knowledge representation can also be influenced by the causability and rectification characteristics
NC	Relevance Satisfaction	The NC metric is related to an actor's required knowledge. This implies that this metric depends on the relevance and satisfaction characteristics
CBC	All characteristics	The CBC metric is related to all characteristics, because every characteristic may have dependencies with others
CM	Improvability Volition	The CM metric can be influenced by the level on which the improvability and volition characteristics are supplied. Both characteristics are related with an actor's cognitive capabilities
ROC	Improvability Volition	The ROC metric is also related with the levels on which characteristics are supplied. Therefore, it is dependent of improvability and volition
COC	Improvability Volition	Insufficient characteristic supply is also related with improvability and volition
CC	All characteristics	The CC metric is related to the number of connectors between all the cognitive characteristics. Therefore, it is dependent of all characteristics
CR	All characteristics	The CR metric is related to equal effects on task fulfillment caused by the application of cognitive characteristics. CR is dependent of all characteristics
CTF	All characteristics	The CTF metric is concerned with the extend to which each characteristic is formalized. Therefore, it is dependent of all characteristics

Please cite this article in press as: S.J. Overbeek et al., Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.002

8. System dynamics model of quality factor metrics

The formal elaboration of the example quality factor metrics, including the ORM model of Fig. 7, can be used to conceive a system dynamics model of the metrics. All 13 quality factor metrics are shown in the model. Notice that the quality levels of the metrics are influenced by variables denoted by the valve symbols. First, it is possible to model dependencies between the quality factor metrics by means of the model. Second, the causal links that influence the level on which a quality factor metric is applied is modeled. Third, the effects of changes in the model and the relationships between the metrics can be determined as well. The meaning of this model can be explained by the system dynamics of, for example, the *node completeness* metric as follows. The level of node completeness (stock) is dependent of the number of child nodes (converter) and the level of sufficient characteristic supply (flow). Obviously, an increase or a decrease of required knowledge also influences the level of node completeness.

The reasoning framework depicted in section 2 shows that there exist dependencies between cognitive characteristics and quality factors. Dependencies between the cognitive characteristics on the one hand and the quality factors on the other hand have already been visualized in the separate system dynamics models of Figs. 4 and 8. The dependencies between the two separate models can be found by studying which effect the supply of each characteristic has on the quality factors. These additional dependencies are explained in Table 6. It can be inferred from Table 6 that the level of every quality factor metric depends on how successful one or more cognitive characteristics have been applied during the fulfillment of a task. For example, it can be seen that the result of the COC metric is dependent of the application of the improvability and volition characteristics. After all, whether cognitive characteristics are sufficiently or insufficiently supplied is dependent of an actor's motivation to perform a task and an actor's cognitive capabilities.

9. Discussion

Cognition and quality are two key concepts that enable to bridge the gap of supply and demand for knowledge intensive tasks. In this context, actors supply cognitive characteristics and stakeholders demand quality. Nevertheless, other approaches of bridging supply and demand for knowledge intensive tasks can be found in the literature. The KnowMore approach discussed in [1] focuses on the support of actors who work on knowledge intensive tasks by automatic delivery of context-sensitive and relevant knowledge. In the case of KnowMore, it can be seen that context-sensitive and goal-specific knowledge is supplied to actors who may demand this when performing tasks. An example of context-sensitive knowledge is the provision of knowledge about earlier conducted tasks. These earlier conducted tasks should, of course, relate to the task that is performed by an actor when the context-sensitive knowledge is provided. The KnowMore study is actor centered. However, possible stakeholders of knowledge intensive tasks may have quality requirements. The provision of context-sensitive knowledge to actors may certainly improve the quality of knowledge intensive task fulfillment. Adding the notion of quality to the KnowMore study can further illustrate the relevance and added value of such context-sensitive knowledge provisions.

A collaborative task-based workplace facilitating knowledge retrieval and sharing among actors has been introduced in [11]. This workplace is referred to as a Knowledge Support ($\mathcal{K}$-Support) system. An adaptive task-based profiling approach is proposed to model dynamic knowledge needs of actors based on their access behaviors or relevance feedbacks on knowledge assets. Task-based knowledge support can then be facilitated to assist actors to access and disseminate task-relevant knowledge. In terms of supply and demand, the $\mathcal{K}$-Support system supplies task-relevant knowledge to actors demanding knowledge to fulfill their tasks. In order to deliver relevant knowledge, the system demands access behaviors and relevance feedbacks. These need to be supplied by actors. The system can also form groups of actors based on similarities in their knowledge needs. Knowledge exchange in these peer groups may additionally facilitate task fulfillment. How the $\mathcal{K}$-Support system is able to improve the quality of task fulfillment is not made clear in [11]. An actor's cognitive load may also be diminished by utilizing such a system due to adequate delivery of knowledge during task performance. Insight in the cognitive characteristics of actors using the system is necessary to understand if and on what level an actor's cognitive load is reduced. The cognitive parts of the proposed models in our study may assist in clarifying this.

To our knowledge, approaches that utilize both the notions of cognition and quality to bridge supply and demand for knowledge intensive tasks are not discussed in existing literature. Most existing work, however, is concentrated on supporting knowledge processing actors by providing (automated) assistance in several ways. By including the concepts of cognition and quality we have provided more insight in possible cognitive requirements of knowledge intensive task types and quality demands of stakeholders. The field of cognitive task analysis, however, mentions the *bracketing heuristic* [9]. The bracketing heuristic is a way to obtain performance predictions for knowledge intensive tasks to be fulfilled. The heuristic requires an elaboration of mandatory and optional task demands so that the bounds on the possible ways of doing the task can be defined. By applying this heuristic insight can be gained in what about the task absolutely must be done. This requires a more thoughtful analysis than simply observing and recording how actors perform a task. In terms of our study, such a heuristic can be helpful to 'bracket' the cognitive, qualitative, and knowledge-related demands for knowledge intensive tasks. For example, implementing such a heuristic in the framework for cognitive and qualitative matchmaking shown in Section 3 and the matchmaking prototype may improve matchmaking processes between supply and demand for knowledge intensive tasks.

Please cite this article in press as: S.J. Overbeek et al., Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.002

10. Conclusions and future work

This paper bridges supply and demand for knowledge intensive tasks based on cognition and quality. First, a framework for cognitive and qualitative matchmaking is elaborated. Actors can supply cognitive characteristics that are demanded by knowledge intensive tasks. The suitability of an actor to fulfill a task can be calculated if the framework is used to match supply and demand of cognitive characteristics. Quality factors are supplied by knowledge intensive tasks and demanded by the stakeholders of those tasks. The certainty that a fulfilled task satisfies quality expectations can be calculated if the framework is used to match supply and demand of quality factors. Next, conceptual models of cognitive characteristics and quality factors are discussed. These models show how actor types, task types, and stakeholders can be related with the concepts of cognitive characteristics and quality factors. This provides insight in the way how object types that conceptualize cognitive characteristics and quality factors are related with each other. The roles that these object types play in knowledge intensive task fulfillment are also clarified. Finally, system dynamics models are introduced to understand how the supply of a certain characteristic influences the supply of a certain quality factor. Particularly, the effects of changes to the supply and demand levels and the dependencies between cognitive characteristics and quality factors are modeled by these dynamic models. This also illustrates which cognitive characteristics affect which quality factors.

Future work is concentrated on evaluation of the conceptual and dynamic models in a case study and implementing the framework extension for qualitative matchmaking of Section 3 in the prototype system. The inductive-hypothetical research strategy (see e.g. [27]) has been successfully applied in a case study that we have conducted in previous work [19]. This strategy can be applied when designing a new case study. The case study to evaluate the models as part of the reasoning framework depicted in Fig. 1 may then consist of the following phases: (1) Description of actor types, knowledge intensive task types and stakeholders as part of a studied domain and their relations. (2) Abstraction of the results of phase 1 of the research strategy to our general model of actor types, task types and stakeholders. Actor types and task types have already been characterized by means of cognitive characteristics in [19,21]. Additionally, it is needed to characterize stakeholders by the quality factors they demand in this phase. (3) Formulation of relations and dependencies between the involved cognitive characteristics and quality factors in a studied domain by using the conceptual and dynamic models. (4) Analysis of phase 3 by examining task results to reveal delivered quality and, from a cognitive point of view, how actors have performed their tasks. (5) Evaluation by comparing phase 1 with phase 4.

Recall from Section 3 that the matchmaking framework includes a *Weigh* function. This function weighes the match result of every cognitive characteristic or quality factor. This enables to stress that the supply of certain cognitive characteristics or quality factors are more (or less) important than others. Up till now, these weigh factors were manually provided in the matchmaking prototype. For example, if the user of the matchmaker system decides that volition is more important than sentience for an actor to supply, then he or she can attach more importance to the volition characteristic. The conceptual and dynamic models introduced in this paper have provided knowledge about relations and dependencies between cognitive characteristics and quality factors. This knowledge can be used in future research to support *weigh factor analysis*. That is, it may be possible to reason about the provision of weigh factors by knowing which supply of cognitive characteristics affect which quality factors and on what level. For instance, a low supply of the relevance characteristic may negatively influence the node completeness quality metric. If a stakeholder of a knowledge intensive task attaches value to this quality metric, then the relevance characteristic should be supplied at an adequate level. In case the relevance characteristic is adequately supplied by an actor it can be substantially weighed.

References

[1] A. Abecker, A. Bernardi, H. Maus, M. Sintek, C. Wenzel, Information supply for business processes: coupling workflow with document analysis and information retrieval, Knowledge-Based Systems 13 (5) (2000) 271–284.
[2] J. Albus, A model of computation and representation in the brain, Information Sciences 180 (9) (2010) 1519–1554.
[3] J. Anderson, Rules of the Mind, Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1993.
[4] M. Chaerul, M. Tanaka, A. Shekdar, A system dynamics approach for hospital waste management, Waste Management 28 (2) (2008) 442–449.
[5] H. de Koning, Scientific grounding of lean six sigma's methodology, Ph.D. Thesis, University of Amsterdam, The Netherlands, EU, 2007.
[6] T. Halpin, Information Modeling and Relational Databases from Conceptual Analysis to Logical Design, Morgan Kaufmann, San Mateo, CA, USA, 2001.
[7] R. Hertwig, G. Barron, E. Weber, I. Erev, The role of information sampling in risky choice, in: K. Fiedler, P. Juslin (Eds.), Information Sampling and Adaptive Cognition, Cambridge University Press, New York, NY, USA, 2006, pp. 72–91.
[8] E. Kako, Thematic role properties of subjects and objects, Cognition 101 (1) (2006) 1–42.
[9] D. Kieras, D. Meyer, The role of cognitive task analysis in the application of predictive models of human performance, in: J. Schraagen, S. Chipman, V. Shalin (Eds.), Cognitive Task Analyis, Lawrence Erlbaum Associates, Mahway, NJ, USA, 2000, pp. 391–438.
[10] D. Koehler, Explanation, imagination, and confidence in judgment, Psychological Bulletin 110 (3) (1991) 499–519.
[11] D. Liu, I. Wu, K. Yang, Task-based K-Support system: disseminating and sharing task-relevant knowledge, Expert Systems with Applications 29 (2) (2005) 408–423.
[12] N. Meiran, Modeling cognitive control in task-switching, Psychological Research 63 (3–4) (2000) 234–249.
[13] D. Nabil, A. EL-Korany, A. Eldin, Quality measuring model for KADS-based expert systems, in: M. Hamza (Ed.), IASTED International Conference on Computational Intelligence, Calgary, Alberta, Canada, IASTED/ACTA Press, Calgary, Alberta, Canada, 2005.
[14] D. Nabil, A. EL-Korany, A. Eldin, Towards a suite of quality metrics for KADS-domain knowledge, Expert Systems with Applications 35 (3) (2008) 654–660.
[15] I. Nonaka, H. Takeuchi, The Knowledge Creating Company, Oxford University Press, New York, NY, USA, 1995.
[16] S. Overbeek, Bridging supply and demand for knowledge intensive tasks: Based on knowledge, cognition, and quality, Ph.D. Thesis, Radboud University Nijmegen, The Netherlands, EU, 2009.

Please cite this article in press as: S.J. Overbeek et al., Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.002

[17] S. Overbeek, M. Janssen, P. van Bommel, Integrating markets to bridge supply and demand for knowledge intensive tasks, in: T. Di Noia, F. Buccafurri (Eds.), E-Commerce and Web Technologies: 10th International Conference, EC-Web 2009 Linz, Austria, August 31–September 4, 2009, Proceedings, Lecture Notes in Computer Science, Linz, Austria, EU, vol. 5692, Springer, Berlin, Germany, EU, 2009.

[18] S. Overbeek, P. van Bommel, H. Proper, Information systems engineering supported by cognitive matchmaking, in: Z. Bellahsène, M. Léonard (Eds.), 20th International Conference on Advanced Information Systems Engineering, CAiSE 2008, Montpellier, France, June 16–20, 2008, Proceedings, Lecture Notes in Computer Science, Montpellier, France, EU, vol. 5074, Springer, Berlin, Germany, EU, 2008.

[19] S. Overbeek, P. van Bommel, H. Proper, Matching cognitive characteristics of actors and tasks in information systems engineering, Knowledge-Based Systems 21 (8) (2008) 764–785.

[20] S. Overbeek, P. van Bommel, H. Proper, Embedding knowledge exchange and cognitive matchmaking in a dichotomy of markets, Expert Systems with Applications 36 (10) (2009) 12236–12255.

[21] S. Overbeek, P. van Bommel, H. Proper, D. Rijsenbrij, Characterizing knowledge intensive tasks indicating cognitive requirements – Scenarios in methods for specific tasks, in: J. Ralyté, S. Brinkkemper, B. Henderson-Sellers (Eds.), Proceedings of the IFIP TC8/ WG8.1 Working Conference on Situational Method Engineering: Fundamentals and Experiences, Geneva, Switzerland, vol. 244, Springer, Boston, USA, 2007.

[22] S. Overbeek, P. van Bommel, H. Proper, D. Rijsenbrij, Matching cognitive characteristics of actors and tasks, in: R. Meersman, T. Zari (Eds.), On the Move to Meaningful Internet Systems 2007: DOA, CoopIS, ODBASE, GADA, and IS, Vilamoura, Portugal, November 25–30, 2007, Proceedings, Part I, Lecture Notes in Computer Science, Vilamoura, Portugal, EU, vol. 4803, Springer, Berlin, Germany, EU, 2007.

[23] C. Pang, R. Zhang, Q. Zhang, J. Wang, Dominating sets in directed graphs, Information Sciences 180 (19) (2010) 3647–3652.

[24] E. Reid, Identifying a company's non-customer online communities: a proto-typology, In: 36th Hawaii International Conference on System Sciences (HICSS-36 2003), vol. 7, Big Island, HI, USA, IEEE Computer Society, Los Alamitos, CA, USA, 2003.

[25] D. Richards, A social software/web 2.0 approach to collaborative knowledge engineering, Information Sciences 179 (15) (2009) 2515–2523.

[26] E. Smith, Mental representations and memory, in: D. Gilbert, S. Fiske, G. Lindzey (Eds.), The Handbook of Social Psychology, fourth ed., vol. 1, McGraw-Hill, Boston, MA, USA, 1998, pp. 391–445.

[27] H. Sol, Simulation in information systems, Ph.D. Thesis, University of Groningen, The Netherlands, EU, 1982.

[28] S. Staab, R. Studer, H. Schnurr, Y. Sure, Knowledge processes and ontologies, IEEE Intelligent Systems 16 (1) (2001) 26–34.

[29] A. ter Hofstede, H. Proper, T. van der Weide, Formal definition of a conceptual language for the description and manipulation of information models, Information Systems 18 (7) (1993) 489–523.

[30] P. van Bommel, A. ter Hofstede, T. van der Weide, Semantics and verification of object-role models, Information Systems 16 (5) (1991) 471–495.

[31] W. van der Aalst, A. ter Hofstede, Verification of workflow task structures: a Petri-net-based approach, Information Systems 25 (1) (2000) 43–69.

[32] B. van Gils, H. Proper, On the quality of resources on the web: an information retrieval perspective, Information Sciences 177 (21) (2007) 4566–4597.

[33] B. van Gils, H. Proper, P. van Bommel, T. van der Weide, Typing and transformational effects in complex information supply, International Journal of Cooperative Information Systems 6 (2) (2007) 229–279.

[34] C. Weir, J. Nebeker, L. Bret, R. Campo, F. Drews, B. LeBar, A cognitive task analysis of information management strategies in a computerized provider order entry environment, Journal of the American Medical Informatics Association 14 (1) (2007) 65–75.

Please cite this article in press as: S.J. Overbeek et al., Statics and dynamics of cognitive and qualitative matchmaking in task fulfillment, Inform. Sci. (2010), doi:10.1016/j.ins.2010.09.002