



Libraries and Learning Services

University of Auckland Research Repository, ResearchSpace

Version

This is the Accepted Manuscript version of the following article. This version is defined in the NISO recommended practice RP-8-2008

<http://www.niso.org/publications/rp/>

Suggested Reference

Lu, Y., & Xu, X. (2017). A semantic web-based framework for service composition in a cloud manufacturing environment. *Journal of Manufacturing Systems*, 42, 69-81. doi: [10.1016/j.jmsy.2016.11.004](https://doi.org/10.1016/j.jmsy.2016.11.004)

Copyright

Items in ResearchSpace are protected by copyright, with all rights reserved, unless otherwise indicated. Previously published items are made available in accordance with the copyright policy of the publisher.

This is an open-access article distributed under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivatives](https://creativecommons.org/licenses/by-nc-nd/4.0/) License.

For more information, see [General copyright](#), [Publisher copyright](#), [SHERPA/RoMEO](#).

A semantic web-based framework for service composition in a cloud manufacturing environment

Yuqian Lu, Xun Xu*

Department of Mechanical Engineering, The University of Auckland, Auckland, 1010, New Zealand

ylu633@aucklanduni.ac.nz, x.xu@auckland.ac.nz

Abstract

Cloud manufacturing has been recognised as a transformative manufacturing paradigm to enable rapid production of highly customised products in a networked environment, through on-demand consumption of cloud-based manufacturing services. The fundamental issue of on-demand manufacturing service provision is service composition, where distributed manufacturing resources are mapped to personalised service requests. This paper examines knowledge-based service composition and adaptive resource planning in a cloud manufacturing environment. The intention is to develop an integrated networked environment, allowing fast resource allocation for a given service request, subject to governance policies, resource access policies, resource availability information and etcetera. The research challenge in this is to explore a feasible service composition method that facilitates easy mapping between service requests and manufacturing resources based on restrictive rule sets in the cloud and availability information about a resource. The research work in this paper analyses the relevant research challenges, proposes a practical approach and implements the solution in the form of a web-based system. The proposed system utilises distributed knowledge for intelligent service composition and adaptive resource planning. A case study is also presented to validate the performance of the proposed approach.

Keywords:

Cloud manufacturing, Semantic web, Ontology, Expert system, Service composition, Knowledge management

1. Introduction

Cloud manufacturing is a service-oriented, high efficiency and low consumption, knowledge-based new mode of networked manufacturing and it has been recognised as a transformative model for future manufacturing [1,2]. The purpose of cloud manufacturing is to move from production-oriented manufacturing processes to service-oriented manufacturing process networks by modelling single manufacturing assets as services and provide them to the variable demand of customers [3–5].

The fundamental issue of providing on-demand manufacturing services in the cloud is the mapping of distributed manufacturing resources with personalised service requests; this process is called service composition [4]. Recognising the need to develop a feasible service composition methodology, much effort has gone into applying well-established optimisation algorithms, such as genetic algorithm, particle swarm optimisation, and ant colony optimisation, to the task of selecting manufacturing resources for a given request. Quality of Service (QoS) is usually used as the optimisation goal in these methods. Parameters such as lead-time and cost make up the key parts of QoS. However, research outcomes from these efforts cannot easily cope with real business cases, as business situations are far more complicated. Most reported research work applied large-data scale numerical algorithms to production planning for real industrial settings where most of the metrics or parameters defined in these algorithms are hard to capture. In addition, many of the reported algorithms tended to simplify complex knowledge-based production activities to quantitative mathematical problems. In the published research outcomes, the common hypothesis is where there is a pool of feasible resources for a given task, the research question is to select the best combination of resources for the task. However, this is often not the case. A service provider has a clear description of a resource's capability boundary. A cloud environment needs to

screen the resource pool to find feasible resources for a given job, before generating an effective service scheme. Therefore, the service composition process in general consists of two phases: (1) capability assessment, which is to find feasible resources for a given task, based on the characteristics of the job and the capability of each unique resource, and (2) service recommendation, where economic analysis and sustainability analysis are carried out, after which an optimal set of manufacturing resources is recommended.

The research work reported in this paper proposes a systematic framework for capability assessment and service recommendation in a cloud manufacturing environment. The proposed solution utilises distributed knowledge for intelligent capability assessment and service recommendation. This unique approach makes the proposed system applicable to various engineering situations, as the main duty is simply maintaining the back-end knowledge base. As service composition in cloud manufacturing is relatively new, Section 2 discusses the unique features and requirements of capability assessment and service recommendation in a practical cloud environment. Section 3 reviews related technologies and Section 4 presents the overall architecture of the proposed solution. Section 5 presents the industrial implementation of the proposed environment with a case study. Conclusions are given in Section 6.

2 Service Composition in Cloud Manufacturing

As discussed in the preceding section, the fundamental issue in cloud manufacturing is to map personalised manufacturing requirements with distributed manufacturing resources. Sometimes, this process can be simple cross-checks between a service request and a service, while other times it requires an experienced engineer to investigate the technical difficulties in undertaking a job with certain resources. Both scenarios require the use of engineering knowledge to drive the decision-making process, though sometimes this knowledge can be as simple as comparison of numerical numbers. In addition, a service recommendation is not just a simple multi-objective optimisation. In an industry setting, the selection of the final service provider(s) can be derived from multiple unquantifiable factors that are critical to a business, such as IP (Intellectual Property) protection concerns, business partnerships and personal relationships. Even for quantifiable factors such as lead-

time and cost, it is not feasible to appoint a default coefficient for each variable that applies to every service consumer because different service consumers have different purchasing preferences. The industrial implementation of cloud manufacturing imposes special requirements on service composition that many proposed solutions fail to address.

2.1 Systematic Knowledge Utilisation

The process of mapping manufacturing jobs with an optimal set of manufacturing resources is a knowledge-intensive activity. This process often requires a manufacturer to reuse existing knowledge (such as drawings, assembly instructions, manufacturing processes and resource capability) to compose a sequence of activities, subject to specified constraints. Knowledge reuse is especially significant for the design and manufacturing process of highly personalised products. Human input is best minimised in this process when a good knowledge-based system can handle these activities.

The collaborative and distributed nature of cloud manufacturing makes knowledge utilisation more challenging. In cloud manufacturing, the product development process often involves contribution from multiple stakeholders concurrently. It, therefore, requires knowledge inputs from different manufacturers. The best practice of production process in a workshop often stems from years of evolution, subject to its in-house resources, and this knowledge could be very different in different workshops. This means using a standard knowledge base for processing manufacturing requests is not always feasible, as the resultant request-resource pairs can conflict with the best practice of some workshops. Hence, a systematic approach for capturing and utilising disparate knowledge from distributed manufacturers is required.

Subsequently, there is a call to design a feasible mechanism to integrate distributed knowledge from different parties, such as tooling suppliers, manufacturers, and regional governments, and use the right subset of knowledge in the process of mapping a service request with manufacturing resources. There are two main tasks to be undertaken: (1) generating a representation scheme for manufacturing

knowledge and (2) creating a mechanism to allow smooth knowledge integration and utilisation in any decision-making activities.

2.2 Dynamic Event Handling

Cloud manufacturing aggregates loosely-connected enterprises, and forms a dynamic marketplace network. In this network, the commencement of new projects and the closure of existing projects happen concurrently. This means a manufacturing resource is in a constant status of switching between being in use and idle. It is, therefore, necessary to consider the actual capacity and availability of a manufacturing resource during service composition.

In other words, cloud manufacturing needs to consider the actual resource capability instead of nominal resource information. Information such as availability, unique access policy, and unique operation of a manufacturing resource are some essential inputs. Service composition is thus required to be capable of processing this information and generating a feasible service plan, adaptive to the real shop-floor status[6].

2.3 Posteriori Articulation of Service Preference

Selection of a preferred service plan for a request is very similar to online shopping, in the sense that a service consumer selects his/her favourite service plan, based on personal experience rather than a default indicator pre-defined by the cloud system. In a cloud manufacturing environment, factors that contribute to the final decision include but are not limited to cost, lead-time, reputation, and transaction history. For a service consumer at a specific business situation, the dominant reason for final selection may be completely different. For a low-urgency order, a service consumer may be inclined to a service plan with low cost, whereas a high-urgency order may require a short lead-time. It is not very feasible to develop a normalised formula that can be applied to every service consumer for all business cases. Therefore, in an industrial setting, one of the basic requirements for service composition is that it outputs all reasonable service plans for a service consumer to select. A more advanced scenario would also allow a service consumer to input additional selection criteria to filter the returned results.

In summary, industry implementation of cloud manufacturing imposes special requirements on service composition because of the knowledge-intensive, collaborative, and web-based nature of cloud manufacturing. Service composition for cloud manufacturing needs systematic knowledge utilisation, dynamic event handling and posteriori articulation of service preferences. The next section reviews existing efforts on service composition.

3 Existing Methods and Tools

In cloud manufacturing, there are three mapping relationships between resources and services [1,7,8]: One-to-One mapping, when a manufacturing resource can only provide a single function and can directly be encapsulated as one service, One-to-Many for a resource with multiple functions or capabilities, which each matches different manufacturing requirements independently, and Many-to-One, when multiple resources are combined to create a more powerful resource.

One-to-One and One-to-Many mapping is technically the same as the service an end-user gets is from one resource, and the mapping process is to find a single resource to undertake a given task. Semantic similarity is recognised as a feasible approach for task and resource mapping. In order to create an intelligent matching process between supply and demand in a cloud manufacturing environment, an ontology method to facilitate unified modelling and the semantic description of manufacturing requirements and resources is presented in [9]. In the proposed four-step process, manufacturing resources and demand attributes are analysed using a semantic similarity algorithm. The output of this process is a sorted list of best matches between demand and resource. To achieve dynamic mapping between manufacturing requirement and manufacturing service, Gao, Yang, Liu and Hou [10] introduced cloud workflow into cloud manufacturing, proposing a conceptual model of a multi-agent business collaboration mechanism between manufacturing service demander, provider and operator. The model defines three critical stages: collaborative business process modelling and verification of cloud workflow, model instantiation with modelling and clustering of manufacturing services, and model execution, with the optimal matching of manufacturing service supplies and requirements. To facilitate more flexible, accurate and automated resource discovery for distributed manufacturing collaboration across ubiquitous virtual enterprises, Cai,

Zhang, Chen, Zhang and Li [11] presented a prototype intelligent system. A semantic web multidisciplinary manufacturing ontology was proposed, to convert resources into machine-understandable knowledge. An ontology-based multi-level knowledge retrieval model was devised to extend the traditional information retrieval approaches based on keyword search, with the integrated capabilities of graph search, semantic search, fuzzy search and automated reasoning, to achieve intelligent discovery of manufacturing resources. A prototype semantic web system (ManuHub) for administration and automatic retrieval of the required distributed manufacturing services was developed in [12]. In this approach, the use of ontology and constraint-based modelling supports semantic matching of manufacturing capabilities. Most recently, Sheng et al. proposed an intelligent service search engine by quantifying the semantic similarity between request and service represented in OWL-S [13].

Effective Many-to-One mapping or service composition will be of great importance for industry implementation of cloud manufacturing. In an industry setting, many manufacturing jobs need the collaboration of multiple resources, or even resources from multiple service providers. The responsibility of a cloud system is then to find and combine a mix of services, which may be heterogeneous and offered by different providers, to collaboratively do a complex job. The search of an optimised service plan for a given task is a global optimisation problem where shorter machining time, lower machining cost and better quality are very often set as the optimisation goals. Cao, Wang, Kang and Gao proposed a service selection model considering time, quality, cost and service as the criteria [14]. In this model, fuzzy decision-making theory is adopted to transform the matrix values into relative superiority degrees. They claimed that this method is different from the traditional linear weighted method of previous research, which results in large values of non-standardisation error. These four relative superiority degrees are then combined linearly into an overall objective. The weight coefficients are calculated through an analytic hierarchy process. After this, ant colony optimisation (ACO) is repurposed for service selection. Many service composition algorithms consider QoS to be an equivalent optimisation objective in a multi-criteria service selection scenario. For example, Tian, Liu, Xu and Yan proposed a Discrete Hybrid Bees Algorithm to produce a near-optimal solution with global optimal QoS [15]. Cui, Ren, Zhang and Wu proposed a service composition algorithm that combines Particle Swarm

Optimisation (PSO) algorithm and K-means clustering [16]. However, these QoS-based optimisation algorithms assume services to be composited are independent of each other, and correlation between services is not considered. Recognised this need, Xu et al. proposed a correlation-aware QoS model and developed an improved Discrete Bees Algorithm based on Pareto, to solve the correlation-aware optimisation problem [17]. Lartigau, Xu, Nie and Zhan presented a method based on QoS evaluation, along with the geo-perspective correlation from one cloud service to another, for a transportation impact analysis [18]. Since composition is an exhaustive process in terms of computational time consumption, the proposed method is optimised through an adapted Artificial Bee Colony (ABC) algorithm based on initialisation enhancement.

Service composition can become very complex when the scale of the resource pool is massive. Xiang, Jiang, Xu and Wang analysed the difficulties and solutions for service composition in a large data environment [19]. They proposed a two-phase service composition mechanism based on a case library. First, a similar case is retrieved from the case library for a given request. Then the case is used to initialise the associated optimisation algorithm, to solve the large-scale optimal selection problem.

It has been found that many researchers recognise that service composition is one of the most important elements in mapping manufacturing resources with service requests. The quality of the mapping results directly impacts the quality of service offering in the cloud. Significant research efforts have been put into generating optimal service plans for a given task. However, some practical issues have not been well-addressed in existing methods. For instance, engineering knowledge from involved parties is not utilised well in the service mapping process. Optimisation algorithms, such as neural networks and genetic algorithms, work on the assumption that the construction of feasible service plans is equivalent to selecting the best combination of processes from a list of feasible candidates. That means some knowledge-based decision support systems need to be used to eliminate invalid plans. Resource dynamics also need to be considered in the decision-making process, because the availability of a resource changes over time. A service plan should take this information into account. In addition, applying multi-objective

optimisation is not practical in a real industrial setting. Selecting a feasible service provider is not always a quantifiable process; there are unquantifiable factors in the decision-making process.

4 A Semantic Web-based Decision-making Framework

This section presents an integrated environment that links highly personalised manufacturing requirements with distributed manufacturing resources in the cloud. This framework is believed to be a novel solution, as the decision-making process makes use of distributed knowledge from various sources providers, and the service plan is adaptable to actual availability information streamed from the shop-floor.

Figure 1 presents the overall architecture of the proposed service composition framework. The overall process of finding a feasible service scheme for a personalised product is organised into three main stages:

- manufacturing requests are verified and parsed by a Product Parser
- series of queries is constructed, to query integrated knowledge base for a list of feasible manufacturing resources
- a Planning Agent takes in these manufacturing resources and evaluates the suggested service plans from the aspects of cost, time and other constraints

The remainder of this section discusses the overall framework in detail.

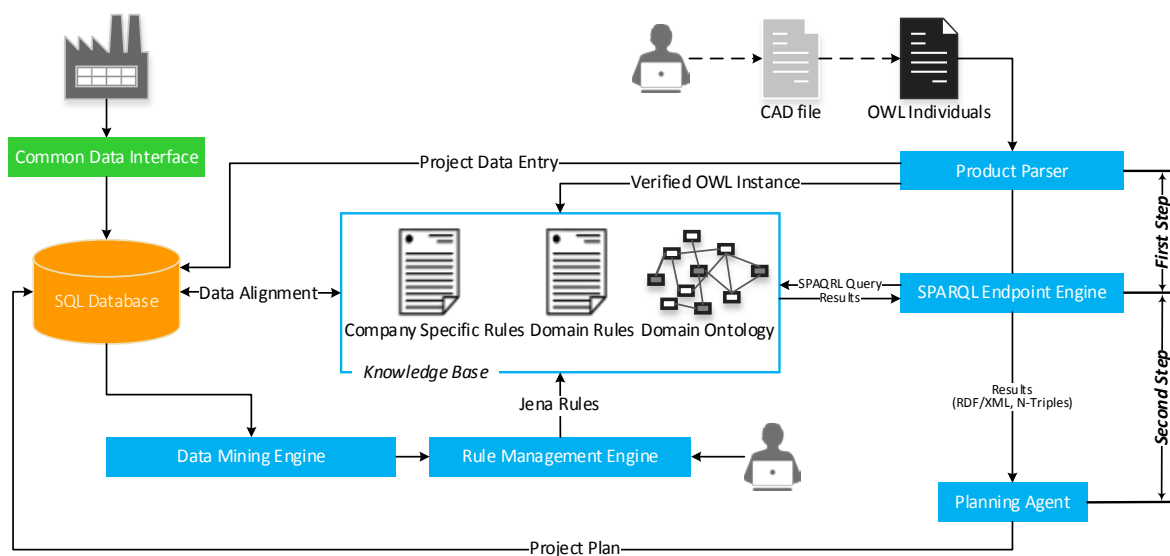


Figure 1: Knowledge-based service composition in a cloud manufacturing environment

4.1 Retrieve Service Request

The proposed system takes in service requests from a service consumer. In this research, service request is defined as a personalised project, which contains product specifications, quality constraints, cost expectations, etcetera. A service project is represented using pre-defined data objects from the developed integrated ontology ManuService and the project is presented as an OWL file (Ma). In the current version, the Product Parser supports direct product parsing from an uploaded OWL file through a web portal. In the future, service requests can be directly created from a CAD environment, such as SolidWorks and Autodesk Inventor. A native CAD file in a CAD environment can be directly converted to an OWL file and uploaded as a service request.

After a service request is parsed by the Product Parser, the verified OWL file is inserted into the central knowledge base and becomes part of the integrated domain knowledge. At the same time, the mirroring version of the data objects is inserted into the central SQL database. The SQL database is the main data repository for all data used in the system. The purpose of storing a replicated version of project data in a SQL database is because querying large amount of project data from the knowledge base using SPARQL is not very efficient. To keep data in the SQL database and the knowledge base in sync, a data alignment service is scheduled to run at set intervals.

4.2 Knowledge-based Decision-making Mechanism

After a service request is successfully uploaded into the system, the SPAQRL Endpoint Engine starts creating queries to question the knowledge base. This process outputs a list of available manufacturing resources for the given service request. Three key components are required in this process: (1) an integrated knowledge base, (2) an inference engine, and (3) a suitable query language.

A knowledge base built for implementing resource selection requires three key components:

1. a formal representation of the domain (an ontology) that represents various aspects of a cloud-based manufacturing business, including but not limited to: product characteristics, quality constraints, manufacturing processes, organisation information, business processes, service capability, and resource specifications
2. product instances – the customised product models input by consumers that describe product information following the definitions in the ontology
3. various rules that specify constraints should be obeyed in the cloud environment. These rules can include regional regulations, unique know-how about a specific resource, and customised access policies set by a service provider.

The ontology being used in this knowledge base (ManuService) includes all the base concepts for service request, business entity, manufacturing resource and business transactions. The proposed environment also allows manufacturers to insert their own know-how into the system. This mechanism gives them the ability to accurately describe their service capabilities. In addition, resource access policies for a resource in the cloud are part of the knowledge source.

In summary, in this knowledge base, an integrated ontology (ManuService) serves as a generic data model that governs overall data structure in the cloud. Personalised products are created, based on the concepts defined in ManuService. After this, product instances (ABox) are input into the knowledge base. In addition, company-specific rules from manufacturers are sourced to the knowledge base after validation by the Rule Management Engine. Alternatively, engineering data stored in the SQL Database is processed by the Data Mining Engine, which outputs Jena rules, and these rules are further transferred as explicit engineering knowledge to the knowledge base. The knowledge base thus stores terminological (TBox) knowledge, assertive (ABox) knowledge (instances specific of the processed service request), and Jena rules (RBox).

A Jena rule is defined by a Java *Rule* Object with a list of body terms (premises), a list of head terms (conclusions), an optional name and optional direction. Each term is either a triple pattern, and extended triple pattern or a call to a built-in primitive. A

rule set is simply a list of Rules. An informal description of the simplified text rule syntax is as follows:

```

Rule      :=  bare-rule
           or  [ bare-rule ]
           or  [ ruleName : bare-rule ]

bare-rule :=  term, ... term -> hterm, ... hterm    // forward rule
           or  bhterm <- term, ... term             // backward rule

hterm     :=  term
           or  [ bare-rule ]

term      :=  (node, node, node)                   // triple pattern
           or  (node, node, functor)               // extended triple pattern
           or  builtin(node, ... node)             // invoke procedural primitive

bhterm    :=  (node, node, node)                   // triple pattern

functor   :=  functorName(node, ... node)          // structured literal

node      :=  uri-ref                             // e.g. http://foo.com/eg
           or  prefix:localname                   // e.g. rdf:type
           or  <uri-ref>                          // e.g. <myscheme:myuri>
           or  ?varname                           // variable
           or  'a literal'                         // a plain string literal
           or  'lex'^^typeURI                     // a typed literal, xsd:* type names supported
           or  number                             // e.g. 42 or 25.5

```

Figure 2: Jena rule syntax

Below is a simple example of Jena rules. Line 1 defines a prefix that can be used in the subsequent rule set. Rule 1 means if something is a lathe, it can perform turning operations. Rule 2 means if something is a twist drill, it can drill a hole. Rule 3 means if something is a conical bottom hole, print out 'A twist drill is needed'.

```

1  @prefix drc: <http://www.codesupreme.com/#>
2
3  [rule1: (?u rdf:type drc:lathe) -> (?u drc:canPerform drc:turningOperations)]
4  [rule2: (?u rdf:type drc:twistDrill) -> (?u drc:canDrill drc:hole)]
5  [rule3: (?x rdf:type drc:conicalBottomHole) -> print('A twist drill is needed')]

```

Figure 3: Simple Jena rule example

Jena supports a range of inference engines or reasoners to be plugged into Jena [20]. It includes a generic rule engine that can be used for many RDF processing or

transformation tasks. In this research, the native rule engine is used to facilitate rule-based inference against RDF graphs. The output from the rule engine is an InfoGraph, which is a collection of triple objects converted from terminological knowledge, assertive knowledge and Jena rules. Then the SPARQL language is used to query the knowledge graph for matched patterns.

The knowledge-based resource selection mechanism allows distributed rules to be plugged into the system for finding a request-resource match. This mechanism ensures the request-resource pairing complies with regional regulations, generic recommendations from international standards and user-defined policies. The resultant knowledge base contains, (1) integrated ontology that incorporates the base concepts for describing highly customised service requests, and base concepts for describing manufacturing resources and capabilities, (2) regional regulations and special know-how, and (3) access policies set by service providers (Figure 4). These rules are all stored and indexed in the central SQL database and they can easily be updated by authorised parties.

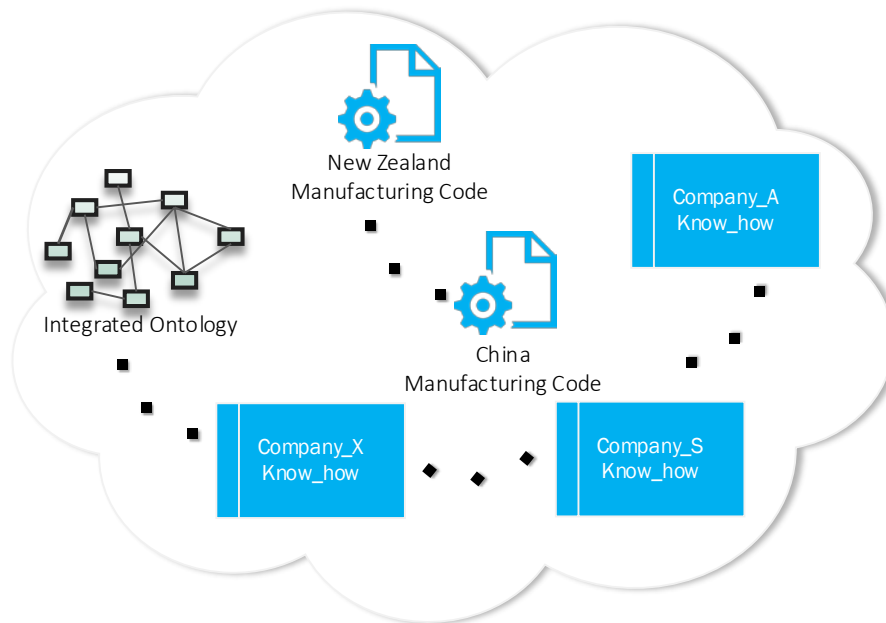


Figure 4: Dynamics of knowledge base in the semantic web

4.3 Adaptive Service Composition

The list of available resources that can be retrieved from the knowledge base is a refined list of services a project can use for production. Till now, the feasibility of a

potential service plan from the perspectives of cost, lead time and sustainability has not been considered. To economically construct a feasible service plan, the system introduces a Planning Agent to construct feasible service plans for a given project, using the available resources and services returned from the SPARQL Endpoint Engine.

The Planning Agent takes a set of data where individual elements are tuples (key/value pairs). In a tuple, key is Task Id and value is Resource Id. For each tuple, the Planning Agent further queries the SQL database for a resource's owner, availability information, etc. With this information at hand, the Planning Agent starts constructing a service plan. In the Planning Agent, cost and lead-time are set as the optimisation targets when constructing a service plan. Provider credit grade and proximity to the consumer are also listed as controlling parameters for a user to filter end results. This feature responds to the need to give service consumers the option of making their own selection from a list of feasible service plans, based on their own selection criteria. At this stage, all the potential service providers will be evaluated on service quality, lead-time, cost, and other factors. A service consumer can filter different service plans based on customising criteria. Once several service plans are selected for detailed quote, the target service providers will receive quotation requests. From there, other downstream activities commence such as contract signing and service delivery management and tracking.

4.4 Service Evaluation Matrix

As discussed, the service mapping process consists of two stages: retrieval of feasible resources for a project (Stage A) and adaptive service generation based on real-time availability information (Stage B). Stage A uses semantic web reasoning to retrieve possible manufacturing resources. In this process, the domain ontology, resource capability descriptions and access rules from service providers, recommendations from third-parties, and regional regulations are all integrated into a knowledge graph. The semantic reasoning engine can then construct SPARQL queries to question the resultant knowledge graph for valid manufacturing resources for given predicates. Once these manufacturing resources are returned from the central knowledge base, the workload of each resource is further queried from the main SQL database for further lead-time calculations.

To quantitatively evaluate QoS, three indexes were introduced. The first index is Service Coverage: it refers to the proportion of unique workpieces that a service provider can assign specific manufacturing resources to. It is defined as follows:

$$SC = S_1/S \times 100\% \quad (1)$$

where,

SC – Service coverage of a service provider,

S_1 – The number of unique workpieces which a service provider can assign at least one specific resource to,

S – The sum of unique workpieces in a service project.

Service Coverage assesses the capability of a service provider for undertaking a project. Very often, a manufacturer claims the capability of providing certain services in a description using natural languages. However, this kind of marketing information does not often reflect the actual capability of a service provider. Moreover, the difference in understanding of a common service offering leads to inefficient repeated talks between service consumers and service providers. Hence, Service Coverage was introduced, to evaluate the technical capability of a service provider. For each unique workpiece in a service project, the semantic reasoning engine queries feasible manufacturing resources from each company. If at least one cloud resource from the service provider can process the workpiece, this workpiece is counted as a soluble workpiece.

The second index is Lead Time, which is used to represent the latency between the placement of a service order and delivery of the requested service. In manufacturing companies, lead-time is often capacity-dependent, and varies as the production capacity changes. In this research, the maximum lead-time is set to 80 days, which is long enough for a small-to-medium-sized service project.

The third index, Service Reliability, is used to represent the probability that a company will perform the agreed service adequately in a defined environment without failure. A service offering can be delayed or suspended because of an unexpected resource breakdown. This would require backup resources to be

employed for the timely delivery of the promised service. The amount of backup resources for a given service offering can reflect a service provider's strength of service capability.

To systematically define service reliability for a highly customised project, the following model was introduced. For a required product in a service project, there are m items in a product, each item is an undetachable object that can either be a workpiece or a sub-assembly. Therefore, a product is represented as:

$$W = \{w_1, w_2, \dots, w_i, \dots, w_m\} \quad (1 \leq i \leq m) \quad (2)$$

where,

m – The number of items in a product structure,

w_i – The i th item in a product.

There are n service providers in the cloud. The company pool can be defined as follows:

$$C = \{c_1, c_2, \dots, c_j, \dots, c_n\} \quad (1 \leq j \leq n) \quad (3)$$

where,

n – The number of service providers in the cloud,

c_j – The j th company in the service provider pool.

For each item in the product, we define a service reliability index to represent a company's ability to deliver the service. This index can be classified into three categories. For an off-the-shelf item such as a motor and bearing, if a service provider has stock for it or can source it from its suppliers, the service reliability for this item is set as 1. Strictly speaking, the ability to source a purchased item varies between companies because of different robustness in supply chain. In this research, supply chain dynamics were not considered. When a company has no matching resource for a fabricated item or no stock for an off-the-shelf item, the company's service reliability for this item is zero. For a fabricated item, a company's service reliability is calculated by evaluating its matching resources for each machining

feature in the item. If we assume the i th item in the product is a fabricated item, it can be represented as:

$$F_i = \{f_{i1}, f_{i2}, \dots, f_{ik}, \dots, f_{ip}\} \quad (1 \leq k \leq p) \quad (4)$$

where,

p – The number of features in the i th item,

f_{ik} – The k th feature in the i th item.

For company c_j , the number of matching resources for the k th feature in item w_i is defined as r_{ijk} . Hence, company c_j 's matching resource set for item w_i is:

$$R_{ij} = \{r_{ij1}, r_{ij2}, \dots, r_{ijk}, \dots, r_{ijp}\} \quad (1 \leq k \leq p, r_{ijk} \in \mathbb{N}^0) \quad (5)$$

If there is an element in R_{ij} that equals zero, the company is not a valid service provider. This is because in common practice, the most granular manufacturing job is at workpiece level, and it is not common that multiple companies collaborate on one workpiece. In this case, the service reliability is zero. When there is at least one matching resource for each feature in w_i , service reliability is calculated by evaluating the addition of all valid manufacturing resources for all the features in w_i . In this research, the service reliability is further scaled into a number between 0 and 1. Therefore, company c_j 's service reliability for item w_i is represented as:

$$SR_{ij} = \begin{cases} \frac{R_{ij,max} - \sum_{k=1}^p r_{ijk}}{R_{ij,max} - R_{ij,min}}, & R_{ij,max} - R_{ij,min} \neq 0, \forall x \in R_{ij}: x \neq 0 \\ 1, & R_{ij,max} - R_{ij,min} = 0, \forall x \in R_{ij}: x \neq 0 \\ 0, & \exists x \in R_{ij}: x = 0 \end{cases} \quad (6)$$

where $R_{ij,max}$ and $R_{ij,min}$ denote the maximum and minimum number of matching resources a company can offer for item w_i among all service providers. After obtaining all service reliability levels for each item in a project, the sum of company c_j 's service reliability for a project is calculated as: $\sum_{i=1}^m SR_{ij}$. Again, a company's service reliability for a project is normalized into a value between 0 and 1:

$$SR_j = \begin{cases} \frac{(\sum_{i=1}^m SR_{ij})_{max} - \sum_{i=1}^m SR_{ij}}{(\sum_{i=1}^m SR_{ij})_{max} - (\sum_{i=1}^m SR_{ij})_{min}}, & (\sum_{i=1}^m SR_{ij})_{max} - (\sum_{i=1}^m SR_{ij})_{min} \neq 0 \\ 1, & (\sum_{i=1}^m SR_{ij})_{max} - (\sum_{i=1}^m SR_{ij})_{min} = 0 \end{cases} \quad (7)$$

The following pseudocode illustrates the overall process, from acceptance of a service project to displaying a list of service plans:

```

START
  FOREACH Item in a Product
    IF the Item is an off-the-shelf item
      Query RDFMatch to find all available service providers
      Mark each service provider as a valid service provider for the Item
    IF the Item is a fabricated item
      FOREACH Feature in the Item
        Query RDFMatch to get all feasible resources for the Feature
      ENDFOR
      Mark a service provider that has at least one matching resource for each Feature as a valid service provider
    ENDFOR
  FOREACH Item in a Product
    IF the Item is a fabricated item
      FOREACH Matching Resource in Resource List
        Check its availability
        IF the lead time is longer than expected project duration
          Remove the Matching Resource from Resource List
        ENDFOR
      ENDFOR
    FOREACH Service Provider in Provider List
      Calculate Service Coverage of the Service Provider
      Calculate Service Reliability of the Service Provider
    ENDFOR
  END

```

Figure 5: Detailed service composition process

In the above process, RDFMatch is the knowledge-building service in the proposed cloud environment, which takes the responsibility of “answering questions” based on the knowledge base at the backend. Note that a service provider can only be treated as a valid service provider for an item when it can offer at least one matching resource for each feature in the item; if the item is a fabricated item, or has stock for the item when it is an off-the-shelf item. This is because it is unrealistic that an item could be scheduled to be machined on different machines from different companies.

5 Case Study

A practical case study was undertaken to validate the proposed service composition mechanism. The industry partner for this case study is a world-leading provider of high-performance mechanical seals for challenging working conditions. An engineer-

to-order production model is adopted by the company. At the start, a sales engineer raises an order after analysing a customer's requirements. Then the design department proposes a customised product design, tailored to the particular working conditions. After this, a design file is sent to its global workshops and suppliers for quoting. Once the customer is happy with the final quote, the actual production and sub-contracts commence.

The task is to identify service providers and select proper manufacturing resources, through which the production of a highly customised mechanical seal solution can be determined. This is carried out by matching the results from the intelligent service composition module in this research. Service composition, or production planning in this case, has been identified as critical business processes that directly schedule a solution design into the process of delivering it. The company has multiple manufacturing sites in different countries, each of which has its unique production specialities. The company currently does not track any data related to production workload and real-time capability in each manufacturing site. Therefore, the sales department will usually need to determine the capacity in each workshop through emails, and plan accordingly after a tailored sealing solution is finalised.

At present, the company attempts to streamline the process of production planning through an integrated IT system with minimum human involvement. Therefore, this case study uses the proposed two-stage approach to configuring a knowledge-based decision-making process, to automatically select the best service providers to produce a highly customised seal project. More specifically, it is desirable that the service composition module be used to support global operation planning in a distributed enterprise, and even in a loosely connected manufacturing network.

5.1 Set Up Testing Data

The manufacturing project used in this case study is the mechanical seal example as shown in Figure 6. This mechanical seal consists of off-the-shelf items and fabricated items (Figure 7). This project was submitted to the cloud environment by a sales engineer from their global project office in Australia. As mentioned before, project specification is in RDF/XML format and is compliant with ManuService ontology. A project specification at the current stage needs to be created semi-

automatically, with the assistance from ontology editors such as Protégé. A snapshot of the serialised project specification in RDF/XML format is illustrated in Figure 8. For detailed definition of data objects used in this case study, please refer to ManuService specification [21].



Figure 6: An example of a customised mechanical seal solution

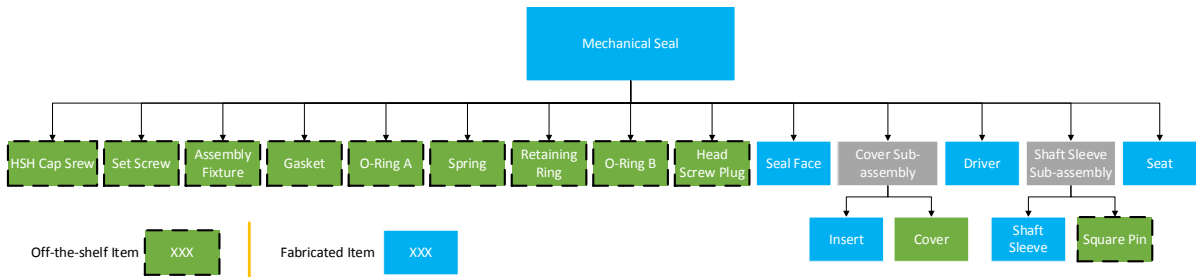


Figure 7: BOM structure of a mechanical seal

```

<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.manu.network.com/manu.service/v1#"
xml:base="http://www.manu.network.com/manu.service/v1"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:ms="http://www.manu.network.com/manu.service/v1#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
<owl:Ontology rdf:about="http://www.manu.network.com/manu.service/v1#">
.....
<owl:NamedIndividual rdf:about="&ms;TrelleBurgmann">
<rdf:type rdf:resource="&ms;BusinessEntity"/>
<telephone rdf:datatype="&xsd:string">+64-09-3285001</telephone>
<hasSICv4 rdf:datatype="&xsd:int">2819</hasSICv4>
<hasNAICS rdf:datatype="&xsd:int">339991</hasNAICS>
<name rdf:datatype="&rdfs;Literal">TrelleBurgmann NZ</name>
<legalName rdf:datatype="&rdfs;Literal">TrelleBurgmann New Zealand Ltd.</legalName>
<description rdf:datatype="&rdfs;Literal">TrelleBurgmann New Zealand, a world leader of seal solution</description>
<category rdf:datatype="&rdfs;Literal">manu facturer</category>
<hasAddress rdf:resource="&ms;TrelleBurgmannAdd"/>
<hasBillingAddress rdf:resource="&ms;TrelleBurgmannAdd"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="&ms;TrelleBurgmannAdd">
<rdf:type rdf:resource="&ms;Address"/>
<postalCode rdf:datatype="&rdfs;Literal">0622</postalCode>
<streetAddress rdf:datatype="&rdfs;Literal">21 Northco te Rd</streetAddress>
<addressLocality rdf:datatype="&rdfs;Literal">Auckland</addressLocality>
<addressCountry rdf:datatype="&rdfs;Literal">New Zealand</addressCountry>
<addressRegion rdf:datatype="&rdfs;Literal">Takapuna</addressRegion>
<hasGeoLocation rdf:resource="&ms;EagleBurgmannGeoCoordinates"/>
</owl:NamedIndividual>
.....
<owl:NamedIndividual rdf:about="&ms;MS-T10436000">
<rdf:type rdf:resource="&ms;Project"/>
<version rdf:datatype="&xsd:string">1.1.2</version>
<dateOriginated rdf:datatype="&xsd;dateTime">2014-06-30T09:30:10Z</dateOriginated>
<dateLastModified rdf:datatype="&xsd;dateTime">2014-11-02T12:30:10Z</dateLastModified>
<name rdf:datatype="&rdfs;Literal">K32665. Lawrie&apos;s Pumps.</name>
<description rdf:datatype="&rdfs;Literal">Mechanical seal, 25 bar. -40C-220C</description>
<jobNumber rdf:datatype="&xsd:string">T232453221</jobNumber>
<status rdf:datatype="&xsd:string">unassigned</status>
<hasOwner rdf:resource="&ms;TrelleBurgmann"/>
<hasCostExpectation rdf:resource="&ms;MSCostExpectation"/>
<hasProduct rdf:resource="&ms;MechanicalSeal"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="&ms;MechanicalSeal">
<rdf:type rdf:resource="&ms;Product"/>
<drawingNumber rdf:datatype="&xsd:string">02-SESUM-SEMP/2.500-00</drawingNumber>
<quantity rdf:datatype="&xsd:int">1</quantity>
<name rdf:datatype="&rdfs;Literal">Modified sesum dual seal</name>
<partNumber rdf:datatype="&xsd:string">T10436000</partNumber>
<hasUnitOfMeasurement rdf:datatype="&xsd:string">suit</hasUnitOfMeasurement>
<hasDrawing rdf:resource="&ms;ProductDesignFile"/>
<hasWorkpiece rdf:resource="&ms;Seal"/>
.....
</owl:NamedIndividual>
.....
<owl:NamedIndividual rdf:about="&ms;Seal">
<rdf:type rdf:resource="&ms;Workpiece"/>
<quantity rdf:datatype="&xsd:int">1</quantity>
<partNumber rdf:datatype="&xsd:string">579985146</partNumber>
<name rdf:datatype="&rdfs;Literal">Seal</name>
<drawingNumber rdf:datatype="&xsd:string">ST271-2.500</drawingNumber>
<includesFeature rdf:resource="&ms;SealCutOff"/>
<includesFeature rdf:resource="&ms;SealIDThroughHole"/>
<includesFeature rdf:resource="&ms;SealODProfile"/>
<includesFeature rdf:resource="&ms;SealPlanarSurface"/>
<includesFeature rdf:resource="&ms;SealSlot"/>
<fromMaterial rdf:resource="&ms;SiliconCarbide"/>
<hasDrawing rdf:resource="&ms;SealFaceDrawing"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="&ms;SealCutOff">
<rdf:type rdf:resource="&ms;ODCutOffTube"/>
<maxWidthOfRemovedMaterialZone rdf:datatype="&rdfs;Literal">5</maxWidthOfRemovedMaterialZone>
<diameterOfInnerHole rdf:datatype="&rdfs;Literal">66</diameterOfInnerHole>
<diameterOfPlanarSplitFace rdf:datatype="&rdfs;Literal">85</diameterOfPlanarSplitFace>
<name rdf:datatype="&rdfs;Literal">Cut off part</name>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="&ms;SealDrawing">
<rdf:type rdf:resource="&ms;Drawing"/>
<fileSize rdf:datatype="&xsd:string">2.3M</fileSize>
<fileFormat rdf:datatype="&xsd:string">dx</fileFormat>
</owl:NamedIndividual>
.....
<owl:NamedIndividual rdf:about="&ms;SealSlot">
<rdf:type rdf:resource="&ms;StraightSlot"/>
<minWidthOfSlot rdf:datatype="&rdfs;Literal">4</minWidthOfSlot>
<widthOfSlot rdf:datatype="&rdfs;Literal">4</widthOfSlot>
<maxWidthOfSlot rdf:datatype="&rdfs;Literal">4.3</maxWidthOfSlot>
<depthOfSlot rdf:datatype="&rdfs;Literal">4.8</depthOfSlot>
<angleOfWallValue rdf:datatype="&rdfs;Literal">90</angleOfWallValue>
<qualityOfWallSurface rdf:datatype="&rdfs;Literal">N9</qualityOfWallSurface>
<name rdf:datatype="&rdfs;Literal">Straight slot</name>
<hasCornerStyle rdf:resource="&ms;SealCorner"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="&ms;SealCorner">
<rdf:type rdf:resource="&ms;CornerRadius"/>
<minFloorRadiusValue rdf:datatype="&rdfs;Literal">0.4</minFloorRadiusValue>
<floorRadiusValue rdf:datatype="&rdfs;Literal">0.5</floorRadiusValue>
<maxFloorRadiusValue rdf:datatype="&rdfs;Literal">0.6</maxFloorRadiusValue>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="&ms;SiliconCarbide">
<rdf:type rdf:resource="&ms;Material"/>
<name rdf:datatype="&rdfs;Literal">Silicon carbide</name>
</owl:NamedIndividual>
.....
</rdf:RDF>

```

Figure 8: Snapshot of the project specification for a customised mechanical seal in RDF/XML format

As mentioned before, the case company has distributed subsidiaries in Austria, New Zealand, Singapore, Japan, US, etcetera. In the Asia-Pacific region, production orders are distributed from project management office in Sydney. Production capacities from workshops in Australia, New Zealand, and Singapore are queried through emails between project managers and local production managers. In the developed cloud environment, we aimed to demonstrate the flexibility of making accurate production planning decisions based on connected workshops and resources in the cloud environment.

Based on the virtualisation methods introduced in [22], three local workshops were virtualised in the cloud environment. Take the New Zealand workshop as an example. It owns several advanced CNC machining centres, such as Mazak Quick Turn Nexus 100-II MS and Mazak Vertical Centre Nexus 510C-II. Each manufacturing resource has a dedicated scheduler associated with it (Figure 9). A production planner can manually edit production events on a manufacturing resource (Figure 10). Alternatively, a machine-tool monitoring system can be connected to the cloud and the scheduler can be auto-refreshed with real-time resource dynamics. This availability information will be used in the service composition process.

Selected Resource:

OKUMA-1011

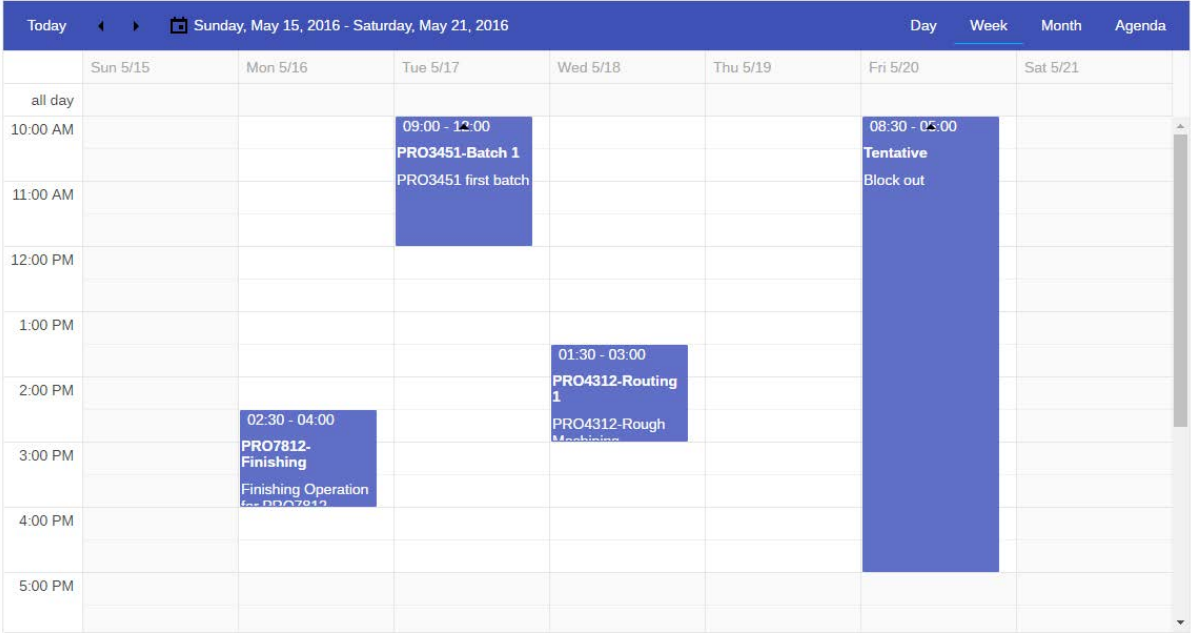


Figure 9: Screenshot of a resource schedule

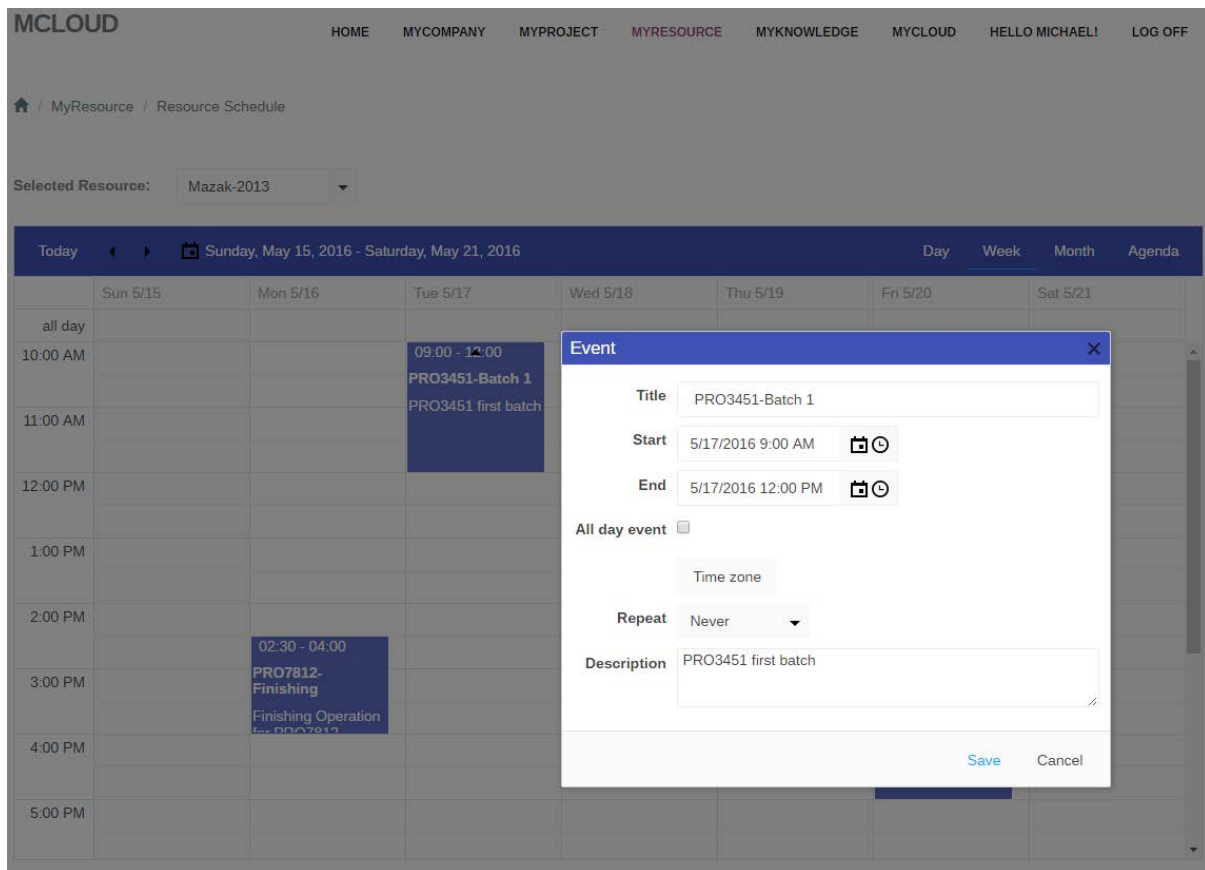


Figure 10: Screenshot of editing a production event of a machine tool

In this test case, some essential domain knowledge was sourced from governance bodies and recognised machine tool vendors, and used as baseline rules when mapping manufacturing resources and the given project. For example, a workpiece size should be less than the maximum working space of a machine tool. Taking Mazak Quick Turn Nexus 100-II MS as an example, the diameter of the workpiece should be smaller than the maximum machining diameter of the machine tool, and the workpiece length should be shorter than the maximum machining length of the machine tool.

Rule: `[(?x ms:hasAccessTo ?y) <- (?x rdf:type ms:Workpiece), (?y rdf:type ms:MazakQTN100IIMS), (?x ms:workpieceDiameter ?d), (?x ms:workpieceLength ?l), (?y ms:maxMachiningDiameter ?maxD),(?y ms:maxMachiningLength ?maxL), ge(?maxD,?d), ge(?maxL, ?l)]`

Another essential rule set is the rules for mapping a specific type of machining feature with a machine tool. For example, a machine tool can process PlanarSurfaceFromTube feature if the machine tool has a turning function. All these rules are stored to the central Microsoft SQL Server at the back end.

```
Rule:[(?x ms:canProcess ?y) <-(?x rdf:type ms:MachiningResource)(?x ms:hasFunction ms:Turning)(?y rdf:type ms:PlanarSurfaceFromTube)]
```


5.2 Service Composition Process


Service composition starts with the project manager from Australia clicking the “Upload” button (Figure 11). The Product Parser will then start parsing the uploaded file, search for feasible manufacturing resources for each fabricated item, and search for inventory for off-the-shelf items from each service provider. It should be noted that a manufacturing resource will not be counted as an available resource if a manufacturing resource is not available within the expected time frame. The service matrix will then be calculated for each available service provider. The mapping result is displayed on the following screenshot (Figure 12). On this page, Service Coverage, Lead Time and Service Reliability are listed for each service provider. A user can adjust the scale bar of each matrix to short-list the available service providers for the project. In the current setting of the proposed method, maximum lead-time is set to 80 days. Service Coverage and Service Reliability are both numbered as a percentage. Once the project manager is happy with a service provider in the list, he/she can click the “Get in Touch” button to ask for more accurate information. The service provider will receive a notification in from the cloud environment as well as an email. Then the typical quoting process begins. If a quote is accepted, production of the project will commence.

- The uploaded project file must conform to the service specifications in ManuService.
- Want to know more about how to create a service request? [Click Here](#).

Select file

Upload


Project Creation Tutorial


ManuService Specifications



Project Examples

Figure 11: User interface for creating a service project

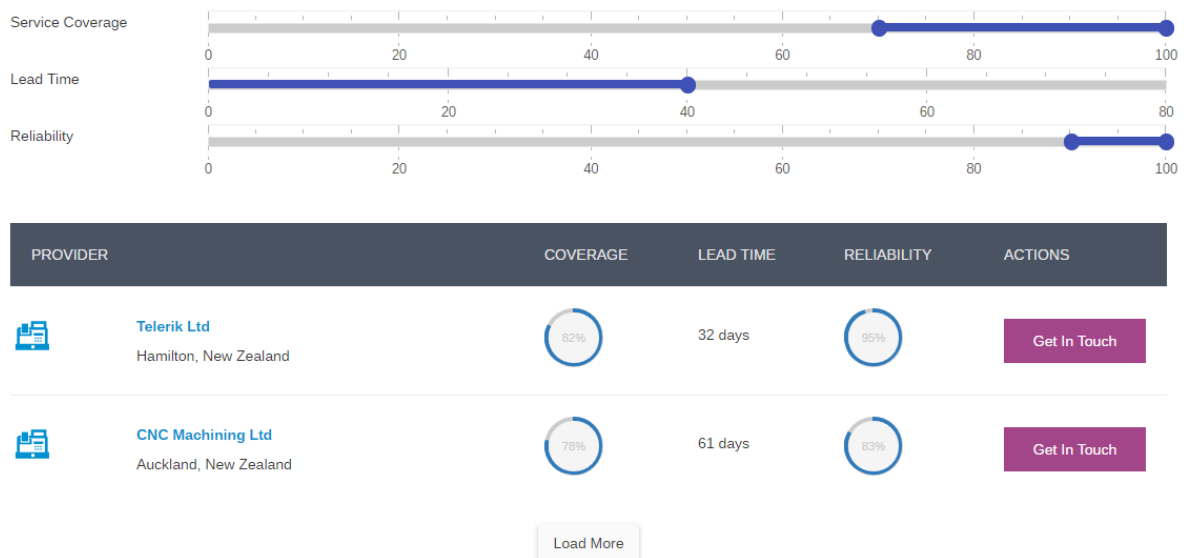


Figure 12: Screenshot of service recommendations for a project

5.3 Discussions

Based on these case study results, it can be said that the knowledge-based service composition module exhibits acceptable accuracy for mapping customised service requests with distributed manufacturing capabilities. This integrated service

composition module successfully connects sales engineers with isolated manufacturing resources from multiple workshops. It enables real-time querying of production capacity, and provides accurate capability information for sales engineers or production planners to forecast and schedule production jobs.

For the demonstrated case study, the successful application of the proposed service composition method was attributable to two key factors: explicit representation of domain knowledge and company rules, and real-time resource availability information. Although it may not always be easy to get a good set of engineering rules from each service provider, it is essential to have expert input throughout the data preparation process. Good quality data is essential for the successful application of the proposed method. It is acceptable if a service provider does not have the expertise to summarise its unique engineering knowledge. It is anticipated that with the running of the developed cloud environment, more and more recommended engineering knowledge will be sourced from machine tool vendors, governance bodies and developers.

Positive feedback was received from the industry partner with regard to the application of semantic-web based service mapping, with the assistance from a dedicated expert system. It was found that a centralised resource management environment is beneficial to a distributed enterprise. This is because it provides distributed manufacturing companies with great flexibility in managing manufacturing resources. The specification of a customised service project in ManuService, and the resource virtualisation process, is a relatively time-consuming process; the generation of ManuService files for service requests and resources, and Jena rules for engineering knowledge can take some time, and require professionals with semantic-web knowledge. However, this is a one-off process and modification to existing records is quite straight forward. In addition, there are a growing number of tools on the Internet to help users with this data input process.

6 Conclusions

Service composition is one of the key research issues for facilitating on-demand service provision in a dynamic cloud environment. Developing a systematic and easy-to-implement service composition mechanism is a critical research challenge in

this regard. The research work proposed a knowledge-based service composition mechanism which allows accurate mapping between distributed manufacturing resources and dynamic service requests. The proposed approach also draws inspiration from past service composition methods to provide a quantitative service evaluation matrix for each mapping result. In this research, the concepts of Service Coverage, Lead Time and Service Reliability are proposed, to evaluate the QoS. It is anticipated that these three indexes will systematically evaluate each service result from different perspectives.

In terms of the service composition process, a two-staged approach is used. The first step is to use engineering knowledge from distributed sources such as governance bodies, recognised machine tool vendors, and individual service providers. These rules are converted to a knowledge map and used as decision-making policies in the process of mapping service requests with manufacturing resources. The second step is to filter returned results against real-time availability information from a resource schedule. This method ensures only available manufacturing resources are sourced for a given job.

The authors believe that the proposed service composition method is easy to implement in real industry settings. The implementation of this method does not require large-scale historical production data from a company, the only input from a company is engineering knowledge from the engineers or management team. The service composition system will become increasingly accurate, once the company starts capturing and storing more and more engineering knowledge.

References

- [1] Xu X. From cloud computing to cloud manufacturing. *Robot Comput Integr Manuf* 2012;28:75–86.
- [2] Wu D, Rosen DW, Wang L, Schaefer D. Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation. *Comput Des* 2015;59:1–14. doi:10.1016/j.cad.2014.07.006.
- [3] Wu D, Greer MJ, Rosen DW, Schaefer D. Cloud manufacturing: Strategic vision and state-of-the-art. *J Manuf Syst* 2013;32:564–79. doi:1.
- [4] Lu Y, Xu X, Xu J. Development of a Hybrid Manufacturing Cloud. *J Manuf Syst* 2014;33:551–66.
- [5] Buckholtz B, Ragai I, Wang L. Cloud Manufacturing: Current Trends and Future Implementations. *J Manuf Sci Eng Trans ASME* 2015;137. doi:10.1115/1.4030009.
- [6] Mourtzis D, Vlachou E, Xanthopoulos N, Givehchi M, Wang L. Cloud-based adaptive process planning considering availability and capabilities of machine tools. *J Manuf Syst* 2016;39:1–8. doi:10.1016/j.jmsy.2016.01.003.
- [7] Ren L, Zhang L, Wang L, Tao F, Chai X. Cloud manufacturing: key characteristics and applications. *Int J Comput Integr Manuf* 2014. doi:10.1080/0951192x.2014.902105.
- [8] Tao F, Zuo Y, Da Xu L, Zhang L. IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing. *Ind Informatics, IEEE Trans* 2014;10:1547–57.
- [9] Tai LJ, Hu RF, Chen CW, Huang YD. Manufacturing Resources and Demand Intelligent Matching in Cloud Manufacturing Environment. *Adv Mater Res* 2012;616–618:2101–4. doi:10.4028/www.scientific.net/AMR.616-618.2101.
- [10] Gao X, Yang M, Liu Y, Hou X. Conceptual model of multi-agent business collaboration based on cloud workflow. *J Theor Appl Inf Technol* 2013;48:108–12.

- [11] Cai M, Zhang WY, Chen G, Zhang K, Li ST. SWMRD: a Semantic Web-based manufacturing resource discovery system for cross-enterprise collaboration. *Int J Prod Res* 2010;48:3445–60.
- [12] Cai M, Zhang WY, Zhang K. ManuHub: A semantic web system for ontology-based service management in distributed manufacturing environments. *IEEE Trans Syst Man, Cybern Part A Systems Humans* 2011;41:574–82. doi:10.1109/tsmca.2010.2076395.
- [13] Sheng B, Zhang C, Yin X, Lu Q, Cheng Y, Xiao T, et al. Common intelligent semantic matching engines of cloud manufacturing service based on OWL-S. *Int J Adv Manuf Technol* 2015:1–16.
- [14] Cao Y, Wang S, Kang L, Gao Y. A TQCS-based service selection and scheduling strategy in cloud manufacturing. *Int J Adv Manuf Technol* 2016;82:235–51.
- [15] Tian S, Liu Q, Xu W, Yan J. A Discrete Hybrid Bees Algorithm for Service Aggregation Optimal Selection in Cloud Manufacturing. *Intell. Data Eng. Autom. Learn.* 2013, Springer; 2013, p. 110–7.
- [16] Cui J, Ren L, Zhang L, Wu Q. An Optimal Allocation Method for Virtual Resource Considering Variable Metrics of Cloud Manufacturing Service. *ASME 2015 Int. Manuf. Sci. Eng. Conf., American Society of Mechanical Engineers*; 2015, p. V002T04A013-V002T04A013.
- [17] Xu W, Tian S, Liu Q, Xie Y, Zhou Z, Pham DT. An improved discrete bees algorithm for correlation-aware service aggregation optimization in cloud manufacturing. *Int J Adv Manuf Technol* 2015:1–12.
- [18] Lartigau J, Xu X, Nie L, Zhan D. Cloud manufacturing service composition based on QoS with geo-perspective transportation using an improved Artificial Bee Colony optimisation algorithm. *Int J Prod Res* 2015:1–25.
- [19] Xiang F, Jiang G, Xu L, Wang N. The case-library method for service composition and optimal selection of big manufacturing data in cloud manufacturing system. *Int J Adv Manuf Technol* 2015:1–12.

- [20] Jena A. Reasoners and rule engines: Jena inference support. Apache Softw Found 2013.
- [21] Lu Y, Wang H, Xu X. ManuService ontology: a product data model for service-oriented business interactions in a cloud manufacturing environment. J Intell Manuf 2016;1–18. doi:10.1007/s10845-016-1250-x.
- [22] Lu Y, Shao Q, Singh C, Xu X, Ye X. Ontology for manufacturing resources in a cloud environment. Int J Manuf Res 2014;9:448–69.