# ODERU: Optimisation of Semantic Service-Based Processes in Manufacturing

Luca Mazzola[1], Patrick Kapahnke[1], Matthias Klusch[1]

DFKI - German Research Center for Artificial Intelligence
Saarland Informatics Campus D3.2, 66123 Saarbrücken, Germany
{Luca.Mazzola, Patrick.Kapahnke, Matthias.Klusch}@dfki.de

**Abstract.** A new requirement for the manufacturing companies in Industry 4.0 is to be flexible with respect to changes in demands, requiring to react rapidly and efficiently on the production capacities. Coupling it with the affirmed Service-Oriented Architectures (SOA) induces a need for agile collaboration among supply chain partners, but also between different divisions or branches of the same company. To this end, we propose a novel pragmatic approach for automatically implementing service-based manufacturing processes at design and run-time, called ODERU. It provides an optimal plan for a business process model, relying on a set of semantic annotations and a configurable QoS-based constraint optimisation problem (COP) solving. The additional information encoding the optimal process service plan produced by means of pattern-based semantic composition and optimisation of non-functional aspects, are mapped back to the BPMN 2.0 standard formalism, through the use of extension elements, generating an enactable optimal plan. This paper presents the approach, the technical architecture and sketches two initial real-world industrial application in the manufacturing domains of metal press maintenance and automotive exhaust production.

## 1 Introduction

As every other aspect of the everyday life, also the manufacturing domain is strongly influenced by innovations in the Information and Communication Technologies (ICT). Companies need to flexibly react to changing demands to remain competitive in a dynamic market. The impact of ICT in this domain is broadly known as Industry 4.0 and ranges from the application of artificial intelligence in robot-assisted production to the usage of Internet of Things (IoT) devices, always connected and controllable just-in-time. Along the same line, manufacturing business processes have to be designed and executed in a more dynamic production context, thus creating the need for adaptation and optimisation at design time as well as at run-time. As a consequence, the design of process models for business applications need to comprise representations for functional and non-functional requirements beyond what can be specified in traditional Business Process Modelling (BPM) systems, such as semantic representations of product models and manufacturing services as well as Key Performance Indicator (KPI) requirements and Quality of Service (QoS) aspects. Moreover, the tools need to be able to provide effective composition of services in the context of SOA and XaaS (Everything-as-a-Service) systems and reliable model optimisation to achieve the best executable service plans for business processes. Eventually, the provided process service plans (PSP) should be designed to support effectively a run-time incremental re-planning, in case an included service is temporarily failing or becomes unavailable.

Due to the unavailability of solutions to tackle these issues in an integrated way, we developed a novel pragmatic approach called ODERU (**O**ptimisation tool for **DE**sign and **RU**n-time). It is able to select the set of compliant services, available to implement the tasks, and subsequently to compose functionally correct plans based on semantic annotations, while optimising their non-functional aspects formalised in terms of a Constrained Optimisation Problem (COP). The resulting complete service plan (services used, their order, the variable bindings and the optimal environmental variables assignment) is encoded back into specifically developed BPMN 2.0 extensions, partially bridging the gap between models and executable plans, providing at the same time the best variable assignments to optimise the outcome of the plan execution.
The rest of paper is organised as follows: In Section 2, related work is briefly presented, then we describe the ODERU basics and algorithm in Section 3; while Section 4 introduces two use cases adopted as applications of ODERU. For each of them a short overview of the scenario is given, followed by a brief description of the design and runtime behaviours. The conclusions are given in Section 5.

## 2 Related Work

Process models are automatically implemented with semantic services by applying techniques of semantic service selection and composition planning , as for Semantic SOA (SemSOA). The key idea is to enable automated understanding of task requirements and services by providing semantic descriptions in a standardised machine-understandable way by using formal ontological definitions [1], for example in OWL2[1]. In [2], the authors propose SBPM, a framework to combine semantic web services and BPM to overcome the problem of automated understanding of processes by machines in a dynamic business environment. Similarly, the authors of [3] propose sBPMN, which integrates semantic technologies and BPMN to overcome the obvious gap between abstract representation of process models and actual executable descriptions in BPEL. [4] follows the same track with the proposal of BPMO, an ontology, which partly is based on sBPMN, while [5] takes sBPMN as basis for the Maestro tool, which implements the realisation of semantically annotated business tasks with concrete services by means of automatic discovery and composition. In [6], a reference architecture for semSOA in BPM is proposed, which aims to address the representation discrepancy business expertise and IT knowledge by making use of semantic web technologies. All of these proposals rely on formalisation different from (although based on)

[1] W3C standard; https://www.w3.org/TR/owl2-overview/

BPMN or do not aim for a full integration from a formalism point of view. In the work [7] the authors propose an approach that uses BPMN extensions to add semantic annotations for automatic composition of process plan and to verify their soundness, but this approach does not consider QoS-aware or run-time optimisation. Adopting a similar approach, ODERU proposes a set of BPMN extensions that not only enable interoperability by offering process model composition, task service selection and process execution, but also provide a way to represent the best values to optimise the QoS and the quality values achieved.

ODERU applies state of the art semantic service selection technologies [8] for implementing annotated process tasks. Non-functional criteria, often referred to as QoS (e.g. costs, execution time, availability), can additionally be considered to find matching services in terms of functional *and* non-functional requirements [9, 10]. In ODERU, optimality with respect to the non-functional QoS specifications is achieved on the process model level by solving (non-)linear multi-objective COP (muCOP) as an integrated follow-up to the pattern-based composition.

Most existing approaches to process service plan composition do not cover the combination of functional (semantic) aspects and non-functional (QoS-aware) optimisation. For example, [5, 11, 12] consider functional semantic annotations to implement business processes by means of a service composition plan. [13] provides a survey giving an overview of existing approaches and initiatives in this direction and highlights research questions. Integrated functional and non-functional optimisation has rarely been considered, with the notable exception of [14]. While composition typically includes the computation of possible data flows, ODERU additionally finds optimal service variable assignments that are also required for executing the resulting plans. This is a feature not yet considered by existing work. Moreover, ODERU performs re-optimisation of process service plans at run-time upon request, which is also a novel feature.

## 3 ODERU: architecture and overview

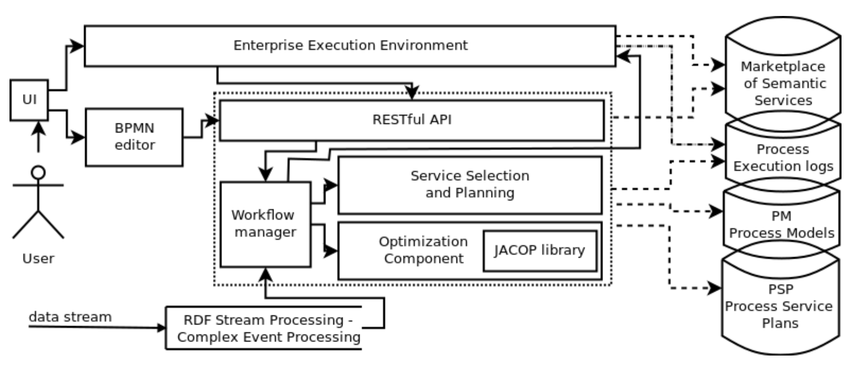


**Fig. 1.** The infrastructure and its interactions with a Process Enterprise Environment.

Given the problem at hand, we identified three main requirements for ODERU. It should support service selection and composition with non-functional optimisation based on flexible measures and objective functions, creating as output a complete plan, directly enactable by an execution environment. Eventually, the format used should simplify the re-use and adaptation of the created plans in a dynamic environment, at run-time. ODERU is a JAVA-based software implemented as a RESTful service. Fig. 1 depicts its basic components and the interactions required for a fully functional Process Enterprise Execution Platform. Only the part enclosed inside dotted line composes the ODERU solution. To provide an optimal solution out of the set of possible functionally valid solutions, ODERU has to make particular choices driven by non-functional requirements, which are expressed as functions of the QoS measures provided by the services. Moreover, it computes concrete settings of service input parameter values, which yield optimal results in terms of the optimisation criteria.

Analogous to the semantic service descriptions themselves, these process model annotations are structured in terms of IOPE and refer to domain knowledge in OWL2. Moreover, the BPMN should specify what QoS measures are to be optimised and how they are defined. This is done by specifying a COP at the process model level, whose solutions dictates what services to choose from and what parameter settings to use when calling services. The COP formulation includes information on how to map optimal parameter values to service inputs and service QoS to COP constants. The outputs produced by ODERU are process service plan encoded in the original BPMN itself by making use of extensions again. Besides the optimal services and input values for calling the services as described above, this also includes possible data flows with parameter bindings among services. Such a process service plan implementing the process model can then be instantiated at run-time by a process plan execution environment. To achieve this, ODERU works following two steps in a sequential manner: first it performs a (A) Pattern-based composition using semantic service selection for all semantically annotated process tasks and the computation of possible data flows. Then, ODERU executes a (B) QoS-aware non-functional optimisation by means of COP solving on the process model level. This second step selects particular services out of sets of functionally fitting services per tasks previously identified, and provides the optimal settings for service inputs.

### 3.1 Semantic Annotation of Tasks and Services

In order to be able to automatically compose functionally valid process service plans given a process model, we assume process tasks to be equipped with structured semantic descriptions. Following the SemSOA approach, IOPE of tasks are described in terms of formalised ontological domain knowledge. For the use cases described in this paper, we propose a reference domain ontology called CDM-Core [15], which provides OWL2 descriptions of concepts from the manufacturing domain, in particular for hydraulic metal press maintenance and car exhaust production. The semantic annotations are embedded in the BPMN model by making use of extension elements at the task level. Similarly, we assume that

**Algorithm 1:** The pseudocode for the process service plan composition

```
Input: PM: semantically annotated BPMN model, S: set of available services
parameter : Sim_min: minimal similarity value accepted
Output: PSP: the computed process service plan
forall s ∈ S do
    IOPE_s → IOPE_S;
end
forall task ∈ PM do
    task → T;
end
forall t ∈ T do
    forall s ∈ S do
        if SIM(IOPE_t, IOPE_s) >= Sim_min then
            s → CANDIDATES_t;
        end
    end
end
forall t ∈ T do
    forall s ∈ CANDIDATES_t do
        forall QoS ∈ T do
            QoS → Parameters_{s_t};
        end
    end
end
Solutions = COPsolver(Parameters);
forall Solution ∈ Solutions do
    ComposeVariableBindings(Solution) → Plans;
end
PSP=MergePMwithSolution(PM, Plans[0]);
return PSP;
```

all services come with semantic annotations of IOPE. For this, the W3C recommendation OWL-S [16] is used, providing means for not only IOPE annotations, but also for the QoS aspect required for the non-functional optimisation.

### 3.2 Constraint Optimization Problem Definition

We defined an appropriate grammar to represent COPs, based on the requirements of the project use cases, but also taking into account its general reapplicability. We relied on a parser generator for this task, and the choice was antlr4 (`http://www.antlr.org/`). This decision allows the definition of complex aggregates of QoS and environment variables instead of mere lists of objectives for simple QoS, extending the expressive capability with respect to the non-functional optimisation problem definition.

### 3.3 Process Service Plan

The computation of the service plan is presented in Algorithm 1, which uses four helper functions. The first one is **SIM** ($IOPE_A, IOPE_B$) computing the similarity between two IOPE annotations based on a selected measure. A second helper function is the **COPsolve** (Parameters) for computing the set of Pareto-optimal solutions of the COP. This is a simple compiler that transform our COP definition into a running instance of a JaCoP solver (see `http://jacop.osolpro.com/`), using the set of parameters given. **ComposeVariableBindings** (Solution) takes care of computing a possible set of variable bindings for the data flow. It is based on the checking of the semantic compatibility of the variables, to ensure a meaningful assignment, going further the simple type compatibility checking. This ensures the direct executability of the computed service plan. Eventually, **MergePMwithSolution** (PM,Plan) takes care of adding the full metadata section into the original process model to create an executable PSP.

**Functional Optimisation (Services selection)** The first step for creating a Process Service Plan is to select all the possible candidates functionally valid for each task. We rely on functionally equivalent *exact* or on *plug-in* matches [17] limited to direct sub class relationships, in order to have a PSP whose logical properties (in term of IOPE) are conserved with respect to the given PM. Every task existing in the process model is considered, as the selection of a valid combination of the task to be actually implemented in the returned process service plan is left for the non-functional optimisation, based on the COP solution.

**Non-Functional Optimisation (Optimal Services composition)** Amongst all the possible combinations of services of the candidate pools of the tasks, the best (or Pareto-optimal in case of a multi-objective problem) option is chosen as part of the overall solution. This implies solving the COP problem associated to the process model. For an introduction to the BPMN extensions defined in CREMA and used by ODERU, we refer the reader to [18].

To achieve its objectives, during the CREMA project a set of functions was designed and implemented into ODERU. The provided calls allow to ask for an integrated composition and optimisation (meaning, considering both the functional and non-functional requirements specified into the input BPMN) or separately, in case when (a) the user is interested only in a functionally valid plan or (b) when exists already a composed plan that requires to be optimised based on the non-functional QoS measures and the user-defined objective function(s). This is valid both at design time (input is a process model) and at run-time (input is an instance of the process model, together with the execution log, if available). For accountability, then a functions to allow the user approval of the computed PSP is provided, together with a set of utility operations, such as for retrieving the ordered list of services found to implement a task and for fetching previously computed PSP, in case when other options would be useful or interesting to be explored.

## 4 Applications

In this section, two applications of the proposed approach are showcased, to demonstrate its applicability and the capability to cover different requirements.

– ***Use Case A: Machine Maintenance*** This first use case refers to the maintenance of hydraulic metal presses, in particular the clutch-brake mechanism, that is its main active part in this scenario. To provide the necessary assistance to the press owner, the producer has some geographically distributed Technical Assistance Service (TAS) organised in teams, which can provide just-in-time on-site maintenance. The selected TAS Team usually requires also one or more replacement parts, in order to restore the full functionality of the press: these replacement parts are provided by some Spare Part (SP) Provider. This PM starts with a task for collecting (and remotely analysing) information about the signalled misbehaviour of the press, to decide if a maintenance is necessary and if the press has to be stopped instantly. Once the maintenance need is confirmed, customer requirements are collected: location, type and length of the warranty, maximum length of the press unavailability and maximum acceptable maintenance costs. Based on this information, the optimisation has to determine the best combination amongst all available TAS Teams and SP Providers, which respects the customer requirements and minimises the total costs. Based on the service selection for these tasks, the model continues computing the earliest possible date for the maintenance, and schedules the actual intervention by proposing and agreeing it with the customer. Eventually, after the maintenance has been executed, the model finishes by collecting customer feedback. At first the semantic service selection is applied, creating a ranked list of candidates for each task, as for Algorithm 1 and then optimisation happens by COP solving.

**Design time optimisation** is defined in this context as a simple reference, as the parameters used for the COP instantiation are some default values. Under this assumption, **run-time optimisation** means recomputing the actual costs and time based on the updates in the model, for example for considering the current scheduling of TAS Teams and the availability and offers for the required *spare part(s)* from the list of SP providers. As a result in both situations, a functionally equivalent process service plan was computed, with minimal value for the *objective function*, and the user partner confirmed that the proposed solution was effectively solving the given problem.

For the **Validation** phase, the user partner is expecting to obtain the following results, by the application of the optimisation to the process model: reduction of up to 60% of the unscheduled machine breakdown. At the same time, it is considering to achieve a reduction of up to 15% of the total machine breakdowns (increasing, consequently, up to about 18 % the machine availability for production operations). On the maintenance intervention, the expected benefit is a reduction of up to 50% in the intervention time and up to 25% in the costs.

– ***Use Case B: OEE for Automotive parts production*** In the second use case, a process model for the production of car exhaust filtering systems is designed. The process starts with the responsible operator selecting a *production task*. This action triggers a sub-process devoted to check availability of and fix

allocation of relevant resources and welding robots. Inside it, two tasks calculate the type and number of robots necessary and available in a loop, and computes for each of them the best parameter settings. At this point the production of the batch can start: the operator loads the required components, lets the welding operation execute, runs testing of the produced exhaust pieces and terminates, reporting any issue if present. Regarding the non-functional optimisation, after the selection step for sets of functionally equivalent services for each task and the following composition of a functionally optimal complete process service plan, in this scenario the COP is only relevant for a particular task, namely the "Allocate Robot". The general idea is to setup the welding robot such that it performs optimally with respect to the three main aspects of OEE (overall equipment effectiveness). OEE is composed of measurements for *availability*, *performance* and *quality*. Although these values are combined to give an overall indication for OEE, the three aspects are typically considered separately in order to provide more insight about the actual reasons for low effectiveness. To explore different (non-dominated) solutions and to better understand the actual behaviour with respect to availability, performance and quality, this optimisation problem is considered as multi-objective optimisation problem. So, the three aspects are separate objective functions with respect to the user-controlled parameters $x$, adjustable to explore the solution space. In this scenario, the Availability is defined as the Mean Time Before Failure (MTBF) of the robot cell: it depends only on the *welding current* **I**, and can be further decomposed in four elementary components. The Performance aspect only directly depends on the *torch speed* **S**, once divided by an ideal cycle time. Eventually, the Quality aspect is the most complex, as it relies on multiple independent dimensions that jointly determine the measure Defect Parts per Million (DPM). Besides the objective functions, a set of constraints is required to further characterise the problem.
**Design time optimisation** means here to compute the optimal parameter setting for the best robot cell existing in the pool of candidates. The settings for this optimisation are based on QoS measures computed from cumulated historical data of the robot cell. In this respect, this is the best possible configuration achievable, without considering constraints coming from robot unavailability or conflicting assignments of the same robot cell. **Run-time optimisation** here is considered for two different cases: (a) there are multiple batches of the same product to construct and after each one of them an analysis searching for better parameters setting can be performed, (b) there is a robot unavailability (e.g. hard failure) and consequently an alternative robot cell can (or should) be considered. As result, the computed process service plans maximize the three components of the *OEE* measure. As there is no natural automatic way of scalarizing the solutions on the Pareto-frontier, the user partner was in charge of selecting the preferred plan amongst the presented options. Its feedback indicated that the proposed solutions seems appropriate and can solve the given problem optimally. For the **Validation** phase, the user partner is expecting to achieve the following results, by optimising the robot cell selection: in a first scenario to increase the speed to allocate production schedule to the manufacturing assets (from the cur-

rent 6 hours to 1 hour), to reduce significantly the time for engaging additional manufacturing assets (from 6 months to 2 weeks) and, eventually, to increase the aggregated OEE measure from the current 60% to 70%. On another scenario, the plan is to increase OEE single components: "Quality" feature from the current 55% close to 75% and "Availability" ones from 60% to 70%.

## 5 Conclusions

In this work we presented our innovative flexible solution to optimal service composition of process models ODERU, which composes functionally correct plans and supports optimisation of non-functional aspects, in the form of a Constrained Optimisation Problem, using as measures generic QoS and supporting user-defined composed objective functions. To showcase the capabilities of the tool, we applied it for two scenarios in the manufacturing domain, with satisfactory results. ODERU will be publicly released at the end of the CREMA project under the Affero GPL v3.0 licence at `https://oderu.sourceforge.io`.

The main advantages of ODERU in respect of the existing approaches are manifold: the first improvement is the business process formulation: it allows a full integration of functional service selection and composition with non-functional optimisation based on user-defined QoS and objective functions arbitrarily complex in the COP. This is achieved through our BPMN extensions and thanks to the development of a grammar for the optimisation part. Secondly, the produced output is directly enactable by an execution environment, being a complete plan. This means that it is equipped with all the relevant information: service assignments, data flow (variable bindings) and optimal variable assignments for initialising the enactment environment. Eventually it, by encoding the computed PSP in an extended BPMN format, allows to maintain in a single place model and plan, together with the variables assignment and the optimality value achieved.

There are still open points we would like to tackle in the future. The most important ones affect (a) the internal ODERU work-flow and (b) the usage of data stream information for proactively directing and guiding the tool behaviour.

***Acknowledgment*** This work was partially financed by the European Commission within the CREMA project, agreement 637066; and by the German Federal Ministry of Education and Research (BMBF) within the project INVERSIV.

## References

1. McIlraith, S.A., Son, T.C., Zeng, H.: Semantic web services. IEEE intelligent systems **16**(2) (2001) 46–53
2. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic business process management: A vision towards using semantic web services for business process management. In: IEEE International Conference on e-Business Engineering (ICEBE) 2005, IEEE (2005) 535–540

3. Abramowicz, W., Filipowska, A., Kaczmarek, M., Kaczmarek, T.: Semantically enhanced business process modeling notation. In: Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications. IGI Global (2012) 259–275
4. Dimitrov, M., Simov, A., Stein, S., Konstantinov, M.: A BPMN based semantic business process modelling environment. In: Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM-2007). Volume 251. (2007) 1613–0073
5. Born, M., Hoffmann, J., Kaczmarek, T., Kowalkiewicz, M., Markovic, I., Scicluna, J., Weber, I., Zhou, X.: Semantic annotation and composition of business processes with maestro for BPMN. In: European Semantic Web Conference, Springer (2008) 772–776
6. Karastoyanova, D., van Lessen, T., Leymann, F., Ma, Z., Nitzche, J., Wetzstein, B.: Semantic business process management: Applying ontologies in BPM. In: Handbook of Research on Business Process Modeling. IGI Global (2009) 299–317
7. Weber, I., Hoffmann, J., Mendling, J.: Beyond soundness: on the verification of semantic business process models. Distributed and Parallel Databases **27**(3) (2010) 271–343
8. Klusch, M., Kapahnke, P., Schulte, S., Lecue, F., Bernstein, A.: Semantic web service search: a brief survey. KI-Künstliche Intelligenz **30**(2) (2016) 139–147
9. Pilioura, T., Tsalgatidou, A.: Unified publication and discovery of semantic web services. ACM Transactions on the Web (TWEB) **3**(3) (2009) 11
10. Zhang, Y., Huang, H., Yang, D., Zhang, H., Chao, H.C., Huang, Y.M.: Bring QoS to P2P-based semantic service discovery for the Universal Network. Personal and Ubiquitous Computing **13**(7) (2009) 471–477
11. Rodriguez-Mier, P., Pedrinaci, C., Lama, M., Mucientes, M.: An integrated semantic web service discovery and composition framework. IEEE Transactions on Services Computing **9**(4) (2016) 537–550
12. Klusch, M., Gerber, A.: Fast composition planning of OWL-S services and application. In: ECOWS'06. 4th European Conference on Web Services, 2006, IEEE (2006) 181–190
13. Strunk, A.: QoS-aware service composition: A survey. In: 2010 IEEE 8th European Conference on Web Services (ECOWS), IEEE (2010) 67–74
14. Zou, G., Lu, Q., Chen, Y., Huang, R., Xu, Y., Xiang, Y.: QoS-aware dynamic composition of web services using numerical temporal planning. IEEE Transactions on Services Computing **7**(1) (2014) 18–31
15. Mazzola, L., Kapahnke, P., Vujic, M., Klusch, M.: CDM-Core: A manufacturing domain ontology in OWL2 for production and maintenance. In: Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 2: KEOD,. (2016) 136–143
16. Burstein, M., Hobbs, J., Lassila, O., Mcdermott, D., Mcilraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., et al.: OWL-S: Semantic markup for web services. W3C Member Submission (2004)
17. Rodriguez-Mier, P., Pedrinaci, C., Lama, M., Mucientes, M.: An integrated semantic web service discovery and composition framework. IEEE Transactions on Services Computing **9**(4) (2016) 537–550
18. Mazzola, L., Kapahnke, P., Waibel, P., Hochreiner, C., Klusch, M.: FCE4BPMN: On-demand QoS-based Optimised Process Model Execution in the Cloud. In: Proceedings of the 23rd ICE/IEEE ITMC conference, IEEE (2017)