

Mock Exam

Despair

1 Provide an overview of the Big Data Analytics-as-a-Service methodology, focusing on the role of procedural models

MBDAaaS is model driven approach towards BDA, their are applied in a specific order, according to their methodology, Declarative, Procedural (output of model selection) and Deployment (output of the workflow compiler). The model Driven approach reach "Usability, productivity, accountability, reproducibility, verifiability and technology neutrality", this way a large basin of costumer is reachable and can operate efficiently analytics, multiple solution can be compared and reused, modularity is enforce, consistency is respected and multiple platform are supported. Two methodologies, code and service based. Service require no code (given by the MBDAaaS), thus is not for expert and for simpler operation, while code based is require a certain degree of skills and is therefore a product of the adv.user. Tuning and refinements is code based, while Design and bootstrap is service based. **Declarative Models** is organized in 5 areas, Representation, Preparation, Analytics, Processing, Visualization and Reporting. Each of these specification can be structured in three levels, Goal (indicator, objective and Constraint) and Feature (type, sub type and sub sub type). BDA web-based GUI is built composing existing services and is based on model transformation. Since basic users will not require coding Analytics services are provided by the target platform. Consistency check is enforce in the declarative model. Building block used for the 3 models, model selection and workflow are 4. **Declarative specification** allow to define declarative models shaping a BDA and retrieving a set of compatible services. **Service Catalog** specifies the set of abstract services (e.g., algorithms, mechanisms, or components) that are available to Big Data customers and consultants for building their BDA. **Service Composition Repository** permits to specify the procedural model defining how services can be composed to carry out the Big Data analytics. **Deployment Configurations** define the platform-dependent version of a procedural model, as a workflow that is ready to be executed on the target Big Data platform. **Service Selection** is operated before procedural model. Service selection receive a list of declarative models, extract areas and categories from declarative models, identify services compatible with the extracted areas, ad return a list for compatible services for each area. **The procedure aim to identify specification that are effective discriminating services. The procedural models** are platform independent models that unambiguously describe how analytics should be configured, following statements (goal and constraints) given by the **declarative models**. The output of a procedural model is a workflow in the form of services orchestration (Sequence,

Choice, If-then, Do-while, Split-join). The user can compose these services to address the scenario objective. Composition have to be connected since the output of one is a requirement of the other. User creates the flow based on the list of returned services. Services enriched with ad hoc parameters. The flow is submitted to the service which translates it into **service composition**. During service composition all internals are explicit, there is clear specification of the services and reuse and modularity is enforced. Is possible to manage multiple declarations, since a task may require another task as said before. Is possible to arrange a solution through connectors between tasks or saving a workflow as a new service in the catalog. The workflow compiler is composed of two sub process. Structure generation and service configuration. The former identify the process operators composing the procedural model. The latter identifies and insert into the deployment model each service in the procedural model. The workflow compiler also support transformations to any orchestration engine available as a service. The output of the workflow compiler is the executable workflow. While the input is the Service composition (from model procedural) and infos on the target platform. Deployment models depends on platform, rive analytics execution in real scenario and move the procedural models towards a target platform through configuration. **Workflow compiler** transforms the procedural model in a **deployment model** that can be directly executed on the target platform. The same way model selection move us from Model declarative to model procedural.

1.1 Extras???

Batch and stream computations:

Our methodology guide the user in selecting consistent set of services for both batch and stream computations.

Multiple platforms:

Our methodology implements a smart compiler supporting the deployment of interconnected computations residing on different platforms.

End-to-end verifiability:

Our methodology provides an end-to-end procedure for checking the consistency of model specifications.

Model reuse and refinement:

Our methodology supports model reuse and refinement

Declarative, procedural and deployment models can be stored in templates to replicate or extend designed computations.