FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

Migration of Processes from Shared to Dedicated Systems

Masterstudium:
Software Engineering & Internet Computing

Johannes Binder

Technische Universität Wien
Information & Software Engineering Group
Arbeitsbereich: Software Technology and Interactive Systems
Betreuer: ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber

Motivation

Processes that are deployed on a shared system are difficult to:

Maintain Updating dependencies of individual processes may also affect other processes. In retrospect it is cumbersome to document the environment of a process.

Relocate Processes can not easily be transferred to another physical host, to improve performance or reliability.

Share It is difficult to share processes with other stakeholders if the process environment is difficult to set up or relies on specific versions of its dependencies.

Preserve Creating a snapshot of the process should not include resources of other processes.

Problem Statement

Following aspects need to be considered when migrating existing process to dedicated virtual machines:

Identification of dependencies From all files and packages deployed on the source system, determine which ones are necessary to execute a specific process.

Documentation of the process environment Represent the dependencies in an open machine-interpretable format (process model).

Determination of tool alternatives Provide the possibility to replace tools that should be replaced by suitable alternatives.

Automated building of dedicated systems Prepare a new virtual system (target system) by installing all dependencies of the process and redeploy the process on it.

Verification of the model and the target system The model needs to contain all dependencies of the process. All dependencies are expected to be deployed on the target system.

Methodology

- Theoretical design of the framework based on current best practices of migration and virtualization
- Implementation of a software prototype implementation based on the results of the theoretical design
- Evaluation of the approach on different scenarios focusing on scripted processes and workflow engines

Migration Process

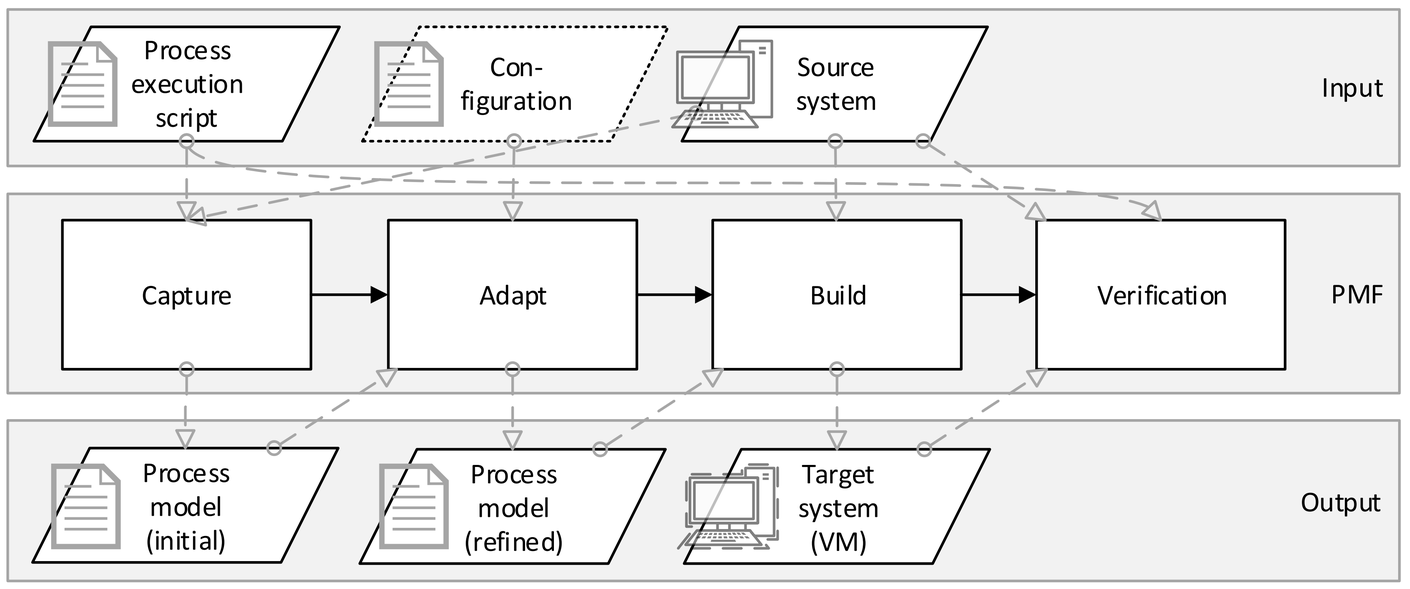
The process migration framework (PMF) is executed on the source system. It runs a prepared script that executes the process (process execution script) and observes the runtime behavior to detect the dependencies (e.g. software components, data, libraries, web services) of the process. The main steps of the PMF are described briefly in the following.

Capture The process environment is identified. The source system is analyzed, and the identified information about the process environment is represented as model.

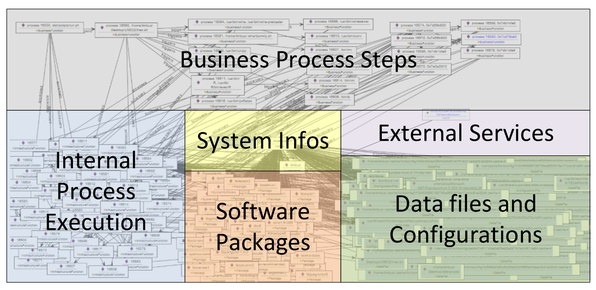
Adapt The model is refined. The model created in the previous step can be adapted by e.g. replacing software etc. This step is optional, but adds the flexibility to handle changing requirements.

Build The target system is built. A virtual machine that corresponds to the refined model and that is able to execute the process is created.

Verification The model and the target system are verified. It is verified if the process on the target system shows the same behavior as the process on the source system. Also the model is verified for correctness and completeness.



High-level representation of the process migration framework (PMF)



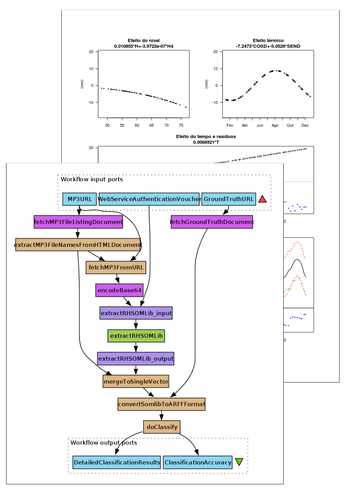
An example for a resulting process model which shows the dependencies of the process, like packages, files, and external resources such as web services, but also the steps of the process

Evaluation

Scenarios The framework is evaluated on an eScience experiment and on a civil engineering business process.

Resources The evaluated processes include software packages, data files, and web services.

Results The processes could be migrated successfully to new virtual systems.



Conclusion

The PMF has several advantages compared to existing migration solutions:

- The migration scope is on individual processes, not the source system as a whole. Only resources that the process accesses are considered and transferred to the target system.
- A documentation of the process environment is generated by the framework. The framework does not rely on existing documentation, which may be out of date or not accurate.
- The framework operates on high level concepts, which improves the maintainability of model and target system (i.e. packages vs. bitlevel).
- Dependencies are deployed natively on the target. There is no additional layer of virtualization between operating system and process introduced by the PMF.

http://www.ifs.tuwien.ac.at/dp/process/projects/pmf.html

Kontakt: e0727990@student.tuwien.ac.at