Currently, every organisation tries to shape its processes to optimally suit the market and offer the best service to the customer. This is generally called Business Process Reengineering (BPR). When an organisation is analysed for the purpose of identifying possibilities for optimising their routines and procedures, three basic principles are outlined:

- A task or activity describes what should be done.
- An organisation describes who should do something.
- An organisation model describes which information is needed to perform an activity.

When looking at current tools that help users to define business process models, there is still a big gap between what organisations require and what is offered.

- Tools only allow to define the surface of processes (how their activities connect to each other and who should perform them), but they do not allow a deeper representation of the reality of the organisation (e.g. its standards, how they relate to their processes, how the behaviour of an activity varies depending on the different people that can perform it).

- Also, given that the number of processes and their variants is potentially huge, there is little help on automatic design and optimisation of processes. This is specially important in these days in which markets change very rapidly, and organisations have to adapt their processes adequately to those changes.

In order to address these issues, AI techniques, and, in particular, planning can help greatly. On one hand AI forces one to explicitly (declaratively) represent the knowledge that one has about a domain, in order to use it later. On the other hand, AI techniques deal very easily with optimisation problems, as well as with automatically searching in huge problem space (process and organisation alternative models in this case).

### BPR and AIP&S

Here we present a first step towards the use of planning techniques in the BPR domain. Some of these ideas have emerged during the meetings of the Technical Co-ordination Unit on Workflow Management of PLANET and preliminary work has been published elsewhere [P.Kearney and D. Borrajo, 2000]. The first task to be done is to understand the analogies between the two domains: BPR and AI planning and scheduling (AIP&S). Then, computational tools able to handle those
have to be defined. Therefore, a formal
description of the entities is needed,
with their properties and relationship
between them. In the business process
management we can identify the
following phases in relation to planning
technology [P. Kearney and D. Borrajo,
2000]:

- **Business process modelling** consists
  on the design, simulation, and
  possibly optimisation of the
  organisation’s processes. From the
  AI perspective, it could be seen as a
  plan, although it is generally
  generated by humans through
  graphical software tools instead of
  being automatically generated by a
  planning system.

- **Business process planning** is in
  charge of the allocation of resources
  and time to each task in a process,
  which relates to AI scheduling.

- **Enactment** refers to the activities
  execution by humans in the
  organisation. There is usually a
  monitoring task that informs of
  problems in the process enactment
  and tries to anticipate events to
  avoid future problems. Relevant
  subfields of AIP&S are plan repair
  or re-planning.

**SHAMASH. An AI-based modelling
tool**

In order to provide a solution to the
integration of planning and BPR, we are
using a business process modelling tool
with AI capabilities, SHAMASH [A.
Sierra-Alonso *et al.*1999, D. Camacho
*et al.*1999], with an AI planner, Prodigy
[M. Veloso *et al.*1995]. SHAMASH is a
process modelling tool carried out in the
course of the R+D project funded by the
Esprit Programme of the European
Commission.

SHAMASH provides a formal language
to represent the processes within an
organisation based on C++ objects and
rules. The first step that the user has to
follow is to define the activities and
entities that compose the process. Each
activity allows to introduce a number of
rules that define its behaviour. For
instance, one can say that the time of
performing a given task depends on the
particular person that carries it out. The
next step is to generate a graphical
model of how activities are linked to
each other to accomplish a given
process, such as the accounting process
of a company. Other capabilities that
the tool offers are the capability of
defining knowledge on standards
(normative constrains of the
organisations with respect to their
processes), behaviour-based simulation
and automatic optimisation of the
process (in order to detect and solve
bottlenecks, inconsistencies, badly used
resources, etc.), text generation (to
automatically create a web-based
version of the processes), and workflow
interface, (that automatically translates
the defined process models into the
input of a workflow engine that is able
to enact the process).

Shamash has been built as a KBS where
its KB is composed of:

- Objects codified as C++ classes and
  instances.

- Rules, defined with a given
  language and a syntax-based rule
  editor that helps the user on their
  creation. The pre-conditions of rules
  that represent the activities
  behaviour define the conditions that
  have to be true to execute the
  activity, and the post-conditions
  define the results of the execution of
  the activity.

- Inference engine that uses a
  modified RETE net matching
  algorithm.
Automating the process modelling task. SHAMASH and Prodigy

Among the most time consuming steps in a modelling task are the definition of the activities with the rules that govern its behaviour and the connection of all the activities, adding conditional branches. Currently, there is no commercial tool that is able to handle these two steps automatically. In our approach, we have first concentrated on the second step [MD. R-Moreno et al.2000], the automatic generation of the process model given a set of activities defined in terms of their behaviour (rules that define their pre- and post-conditions). In order to do so, our approach consists on letting the user define the organisation activities using an object-oriented and rule-based approach through SHAMASH, automatically translate the activities specifications to Prodigy input language, letting Prodigy generate a plan (process model using the activities defined in SHAMASH), and then translating back to SHAMASH the plan.

The translation process has been performed having in mind the relationship among the different representation elements of both fields.

• Predicates: given that the factual knowledge in SHAMASH is represented as C++ objects, we had to translate this information into predicates in order to be used by Prodigy. Therefore, we used a straightforward translation mechanism: for each class C with attribute A, we defined a predicate A with two arguments; the first one has as value the identifier of each instance of that class, and the second has the value of that attribute for that instance. If the value is a list of values Vi, a ground predicate is created for each one: A(I,Vi).

• Types: Prodigy requires that each variable in the operators belongs to a user-defined type. Thus, we had to translate every SHAMASH class into a Prodigy type.

• Operators: given that the pre-conditions of activities in SHAMASH have the same meaning as pre-conditions of planning operators, we translated every pre-condition (represented in SHAMASH as rules) into operators preconditions. To do so, we had in mind the previous conversion step of objects descriptions into predicates. For instance, if a pre-condition rule said “If there is an instance I of a class C such that value of attribute A is V” (where the values of A are of type T), it was translated into a pre-condition of the corresponding operator saying “A(I,V)” where I is of type C and V is of type T. Correspondingly, post-conditions rules were translated into effects of operators.

• Problem: the initial state has been taken from the description of the organisation and its resources that are represented in SHAMASH also as classes, and instances. The goal in this first prototype had to be explicitly given by the user as the post-condition rule of the end activity of the process that is modelling.

• Plan: the planner generates a sequence of instantiated operators which represent in SHAMASH terminology the activities that have to be performed in order to carry on the process which will lead to the specified goal. In fact, it is an instantiated process in which the specific resources of the organisation are given as arguments of the instantiated operators.
Other advantages of this approach

Until now, we have outlined the advantages of using AI planning for automatically generating the processes models. However, we believe that using a BRP tool can also be very helpful for AI planning. SHAMASH tool description language is one that is closer to final users and programmers, overcoming one of the strongest problems for marketing planning techniques which is how to input the domain theory (see roadmap of Knowledge Engineering TCU of PLANET). The concepts used by SHAMASH are the same ones that organisations apply to their processes. Also, from a tool-based approach, rules (that correspond to the same knowledge as operators pre- and post-conditions) are entered through a syntax-based rules editor in SHAMASH, allowing an automatic verification of the syntax, and guiding the user with the pre-defined classes, instances and attributes.

Bibliography


