Causal Networks Based Process Improvement

D. Günther1, R. Neumann2, K. Georgieva2 and R. R. Dumke2
1CERT, Volkswagen AG, Wolfsburg, Germany
2Dept. of Computer Science, University of Magdeburg, Magdeburg, Germany

Abstract - This paper includes a causal-based modelling of software process models in order to analyse the correct relationships between the different (key) process areas of these models. A first short description of causal network approaches shows the identified problems and possible benefits using these formal techniques in the software engineering area. The definition and extension of the causal modelling using causal networks helps to understand the relationships between the different software process artefacts and their causalities. Our causal network based process model (CNPM) concept describes the considered objects outside and inside the software processes or functions and their causalities expressed as roles. The description of first applications of the CNPM approach for the Capability Maturity Model Integration (CMMI) demonstrates the meaningfulness of this approach.

Keywords - Software process improvement, causal network, process analysis and evaluation, software quality

I. INTRODUCTION

Causal networks as a special kind of semantic networks are very expressive in order to see or analyze the relationships between process activities, areas and indicators in a logical manner. Typical results of such a modelling are

- The consequence of process activities to other ones involving different quality characteristics like correctness, completeness etc. (see [4] and [13])
- The repercussion of the chosen approaches for process evaluation and improvement (see [3] and [9])
- The overview about strong and weak process connections in order to keep quality improvements (see [2], [5] and [6])
- The application of (causal) model-based principles in order to reduce the process complexity and involvements (see [1], [7] and [10]).

In a general manner a causal network “is a directed acyclic graph arising from an evolution of a substitution system, and representing its history” [13]. The process evolution involves causal relationships between events, states, entities, objects, artefacts etc. which could be based on a special kind of empirical reasoning. In following we will characterize a causal network based approach that helps to identify incompleteness and mismatches of text-based process models in an explicit manner.

II. CAUSAL NETWORK-BASED PROCESS MODEL DESCRIPTION AND ANALYSIS

The causal network based process model (CNPM) concept is defined in the following four parts and components of this approach (see [2], [3] and [11]):

A. Causal Network Model Components

The causal network model $M^{\text{CNPM}}$ is based on the following software process ingredients and involvements:

$$M^{\text{CNPM}} = (U^{\text{CNPM}}, V^{\text{CNPM}}, F^{\text{CNPM}}),$$

where

- $U^{\text{CNPM}}$ is a set of background variables that is determined by objects $o_{u,i}^{\text{CNPM}}$ ($i \in \{1,2,...,m\}$) as software process artefacts outside the considered model
- $V^{\text{CNPM}}$ is a set $\{V_{1}^{\text{CNPM}}, V_{2}^{\text{CNPM}}, ..., V_{n}^{\text{CNPM}}\}$ of variables that are determined by objects $o_{v,i}^{\text{CNPM}}$ ($i \in \{1,2,...,n\}$) in the model – that is, variables or objects in $U^{\text{CNPM}} \cup V^{\text{CNPM}}$; and
- $F^{\text{CNPM}}$ is a set of functions $\{f_{1}^{\text{CNPM}}, f_{2}^{\text{CNPM}}, ..., f_{n}^{\text{CNPM}}\}$ such that each $f_{i}^{\text{CNPM}}$ ($i \in \{1,2,...,n\}$) is a mapping from (the respective domains of) $U^{\text{CNPM}} \cup V_{i}^{\text{CNPM}}$ to $V_{i}^{\text{CNPM}}$ and such that the entire set $F^{\text{CNPM}}$ forms a mapping from $U^{\text{CNPM}}$ to $V^{\text{CNPM}}$. In other words, each $f_{i}^{\text{CNPM}}$ tells us the value $V_{i}^{\text{CNPM}}$ given the values of all other variables in $U^{\text{CNPM}} \cup V_{i}^{\text{CNPM}}$ and the entire set $F^{\text{CNPM}}$ has a unique solution $V^{\text{CNPM}}(o)$. Symbolically, the set of equations $F^{\text{CNPM}}$ can be represented by writing

$$o_{v,i}^{\text{CNPM}} = f_{i}^{\text{CNPM}}(r_{i}^{\text{CNPM}}, o_{v,j}^{\text{CNPM}}, o_{u,j}^{\text{CNPM}}),$$

where $r_{i}^{\text{CNPM}}$ is any realization of the unique minimal set of variables as roles $R_{i}^{\text{CNPM}}$ in

---

1 Against the causality in natural science, software processes are based on activities of subjects. Therefore, we use a description of subjects as roles. Note that the roles define the
The full formal description you can find in [3] and [11].

Examples of this CNPM description are

1. \( M_{1}^{CNPM} = \langle U_{1}^{CNPM}, V_{1}^{CNPM}, F_{1}^{CNPM} \rangle \) with 
   \( U_{1}^{CNPM} = \{ \text{Object 1}' \}, V_{1}^{CNPM} = \{ \text{Object 2}' \}, \)
   \( F_{1}^{CNPM} = \{ o_{1}^{CNPM}, r_{1}^{CNPM} \}, o_{1}^{CNPM} = \{ \text{Function 1}' \} \) and \( r_{1}^{CNPM} = \{ \text{Role 1}' \} \)

2. \( M_{2}^{CNPM} = \langle U_{2}^{CNPM}, V_{2}^{CNPM}, F_{2}^{CNPM} \rangle \) with 
   \( U_{2}^{CNPM} = \{ \text{Object 2}' \}, V_{2}^{CNPM} = \{ \text{Object 3}' \}, \)
   \( F_{2}^{CNPM} = \{ o_{2}^{CNPM}, r_{2}^{CNPM} \}, o_{2}^{CNPM} = \{ \text{Function 2}' \} \) and \( r_{2}^{CNPM} = \{ \text{Role 1}' \} \)

Considering the different levels of causality as dependencies and improvements leads to different kinds of analysis and interpretations.

B. Causal Network Model Operations

The \( M_{CNPM} \) can be modified in the following manner considering the typical causal relationships between software process artefacts:

- **Union or summarizing** of CNPM models consists of the union of the different model parts. A unified CNPM model \( M_{3}^{CNPM} \) could be built as \( f_{\text{join}}(M_{1}^{CNPM}, M_{2}^{CNPM}) \) with 
  \( U_{3}^{CNPM} = \{ \text{Object 1}' \}, V_{3}^{CNPM} = \{ \text{Object 2}', \text{Object 3}' \}, F_{3}^{CNPM} = \{ o_{3}^{CNPM}, r_{3}^{CNPM} \}, o_{3}^{CNPM} = \{ \text{Function 1}', \text{Function 2}' \} \) and \( r_{3}^{CNPM} = \{ \text{Role 1}' \} \)

- **Partitioning** of a CNPM model consists of building sub models and special parts of models.

- **Restructuring** of a CNPM model is reasonable in different practical situations and consists of addition and extraction of any model parts.

C. Causal Network Model Analysis

The \( M_{CNPM} \) can be analyzed considering the typical causal relationships between software process artefacts in the following manner. The CNPM model could be considered as a directed graph where every node has some predecessors and any successors. Hence, it is possible to analyze or count these elements for a first level of CNPM analysis and evaluation. For instance, we obtain the number of all roles in the CNPM, the number of derived objects etc. Based on this idea, we can define the following function \( f_{\text{extract,input}}^{CNPM} \) of analysis as

\[
f_{\text{extract,input}}^{CNPM} : M_{x}^{CNPM} \times f_{i}^{CNPM} \rightarrow U_{i}^{CNPM}
\]

where \( M_{x}^{CNPM} = <U_{x}^{CNPM}, V_{x}^{CNPM}, F_{x}^{CNPM}> \), \( f_{i}^{CNPM} \in F_{x}^{CNPM}, U_{i}^{CNPM} \subseteq U_{x}^{CNPM} \) and

\( U_{i}^{CNPM} = \{ u_{i}^{CNPM} : u_{i}^{CNPM} = \text{predecessor}(f_{i}^{CNPM}) \} \).

Applying these functions to our described examples of CNPM models we can derive the following characteristics:

- \( \text{predecessor}(f_{1}^{CNPM,M2}) = \{ \text{Role 1}', \text{Object 2}' \} \)
- \( \text{predecessor}(f_{1}^{CNPM,M3}) = \{ \text{Role 1}', \text{Object 1}' \} \)
- \( \text{successor}(f_{2}^{CNPM,M3}) = \{ \text{Object 3}' \} \)

D. Causal Network Model Exploration

The \( M_{CNPM} \) can be evaluated in the following manner considering the typical causal relationships between software process artefacts. The CNPM model could be characterized as empirical evaluation that requires the identification of the empirical aspects explicitly. Such empirical characteristic for objects could be process artefact level, artefact quality or process artefact performance. From this point of view, the CNPM model evaluation could be performed as following:

- **causal coverage analysis** of the fulfilled requirements from a special software process point of view,
- **causal trace analysis** of the successful consideration of process flow based requirements,
- **causal achievement analysis** of the derived results and outputs in different parts on the CNPM model.

In order to explain some of these kinds of analysis we will consider the CPNM model \( M_{y}^{CNPM} \) describing the empirical-based process aspects mainly and the CPNM model \( M_{y}^{CNPM} \) describing the causal basics in general. On that we characterize a simple causal coverage analysis as

\[
\text{coverage}_{M_{y}}^{CNPM} = \sum(|F_{y}^{CNPM} | | U_{y}^{CNPM} | | V_{y}^{CNPM} | /
\sum(|F_{y}^{CNPM} | | U_{y}^{CNPM} | | V_{y}^{CNPM} |)
\]

where \( F_{x}^{CNPM} \subseteq F_{y}^{CNPM}, U_{x}^{CNPM} \subseteq U_{y}^{CNPM}, V_{x}^{CNPM} \subseteq V_{y}^{CNPM} \).

causal heuristics addressed to the considered/presented function in the set of the software process artefacts.

\footnote{The full formal description you can find in [3] and [11].}
Furthermore, in the case of coverage lower 1 we have the situation of any missing objects. That could be characterized in the following manner.

\[ F_{\text{missing function}}^{\text{CNPM}} = \{ F_x^{\text{CNPM}} \setminus F_y^{\text{CNPM}} : F_y^{\text{CNPM}} \subseteq F_x^{\text{CNPM}} \} \]

\[ F_{\text{missing input}}^{\text{CNPM}} = \{ U_x^{\text{CNPM}} \setminus U_y^{\text{CNPM}} : U_y^{\text{CNPM}} \subseteq U_x^{\text{CNPM}} \} \]

\[ F_{\text{missing output}}^{\text{CNPM}} = \{ V_x^{\text{CNPM}} \setminus V_y^{\text{CNPM}} : V_y^{\text{CNPM}} \subseteq V_x^{\text{CNPM}} \} \]

For the causal trace analysis and achievement analysis the existing graph algorithm and methods of evaluation can be used that would not be considered here.

III. CNPM APPROACH APPLICATION FOR CMMI ANALYSIS

One of the possible uses for the CNPM model is the mapping of process standards. This shall be described by example of the key process area „Organizational Training“ (OT) of the CMMI (see [8] and [12]). Also it will be considered that specific practices of this model give a hint for the implementation of a CMMI conformant process environment. The specific practice (SP) 1.1 will be used as an example for the implementation of a first part of an CNPM network.

A. CNPM-Based Analysis of SP 1.1

The CMMI practice SP 1.1 as “Establish the Strategic Training Needs” contains the following sub practices:

- Analyze the organization’s strategic business objectives and process improvement plan to identify potential future training needs.
- Document the strategic training needs of the organization.
- Determine the roles and skills needed to perform the organization’s set of standard processes.
- Document the training needed to perform the roles in the organization’s set of standard processes.
- Document the training needed to maintain the safe, secure and continued operation of the business.
- Revise the organization’s strategic needs and required training as necessary.

To create a network it is necessary to split the text into tasks, objects and roles. This decomposition leads to the following elements:

**Objects:** Strategic business objectives, Process improvement plan, Set of standard processes, Training needs for roles and skills, Training needs for business, Needed roles, Needed skills

**Functions:** Analyse, Document strategic training needs, Determine roles and skills, Document training needs to perform standard processes, Document training needs for safe, secure, continued business, Revise if necessary

**Roles:** The text of the CMMI contains no detailed information about the role executing the task. But it gives the general definition, that the management is responsible for all quality activities. So for the following networks the management will be used as executing instance of this task.

The resulting network is shown in the following figure 1.

![Organizational Training-SP 1.1 – first approach](image1)

A deeper analysis of the objects contained in this network shows, that there is no task, creating the objects „training needs for roles and skills“ and „training needs for business“. This shows the incompleteness of the CMMI in some detailed views. The inserted processes are the following:

- Determine training needs for roles and skills
- Determine training needs for business

Furthermore, it can be seen, that the network contains two functions for documenting two different types of training needs. Giving credit to the fact that the documentation of training needs doesn’t depend on the type of the training need that is to be documented, both functions can be combined to a single one.

- Document training needs

The network constructed by these changes is shown in the figure 2. Using the methods described above, the derived networks about the other SP 1 components (as SP 1.2, 1.3, 1.4, 2.1 and 2.2) can be combined to show the complete picture of the tasks fulfilling the requirements of key process area “organizational training”.
B. CNPM-Based Analysis of CMMI Process Descriptions

Further results of CMMI key process (KP) analysis can be characterized as the following chosen situations:

- KP “Causal analysis and resolution”: created but not used practice 2.1 the “Products”,
- KP “Configuration management”: created but not used practice 1.2 the “Change request database”; double (not unique) definition of practice 3.1 as “Configuration documentation”
- KP “Decision analysis and resolution”: created but not used practice 1.1 the “Organization standard processes”
- KP “Organizational training”: missing management component (see above)
- KP “Project monitoring and control”: created but not used practice 1.3 the “Risks documentation” and practice 1.6 the “Changes documentation”
- KP “Project planning”: created but not used practice 2.5 the “Knowledge management”
- KP “Process and product quality assurance”: created but not used practice 1.2 the “Evaluation criteria” and practice 1.1 “Quality requirements”
- KP “Requirements development”: created but not used practice 1.23 the “Requirements history” and practice 1.5 the “Review results”

Note, that the analysis of the KP’s “Integrated project management”, “Measurement and analysis”, “Organizational innovation and deployment”, “Organizational process focus” and “Organizational process performance” could be identified with correct (causal-based) semantics.

IV. CONCLUSIONS

The presented CNPM-based approach was applied in practice in order to transform the textual CMMI standard in a causal network based form. This implies the chance of explicit description of the CMMI process evaluation from an implicit one. Furthermore it allows to consider other causalities and empirical relationships in the software process area depending on concrete industrial situations and methodologies.

In our further research we interpret any improvements in order to keep causal-based correctness in the CMMI. These investigation led to any improvement documented in our next papers.

V. REFERENCES