Analyzing Knowledge Transfer Effectiveness – An Agent-Oriented Modeling Approach

Markus Strohmaier
University of Toronto and Know-Center Graz
mstrohm@cs.toronto.edu

Eric Yu
University of Toronto
yu@fis.utoronto.ca

Jennifer Horkoff
University of Toronto
jenhork@cs.toronto.edu

Jorge Aranda
University of Toronto
jaranda@cs.toronto.edu

Steve Easterbrook
University of Toronto
sme@cs.toronto.edu

Abstract

Facilitating the transfer of knowledge between knowledge workers represents one of the main challenges of knowledge management. Knowledge transfer instruments, such as the experience factory concept, represent means for facilitating knowledge transfer in organizations. As past research has shown, effectiveness of knowledge transfer instruments strongly depends on their situational context, on the stakeholders involved in knowledge transfer, and on their acceptance, motivation and goals. In this paper, we introduce an agent-oriented modeling approach for analyzing the effectiveness of knowledge transfer instruments in the light of (potentially conflicting) stakeholders' goals. We apply this intentional approach to the experience factory concept and analyze under which conditions it can fail, and how adaptations to the Experience Factory can be explored in a structured way.

1. Motivation

Facilitating the transfer of knowledge between knowledge workers represents one of the main challenges of knowledge management (KM) [1] [2]. As a consequence, past research on knowledge management has led to the emergence of a broad range of instruments, modeling languages and theories for knowledge transfer. *Knowledge transfer instruments* were proposed to facilitate knowledge transfer by means of *technological, organizational and sociological approaches.* On an organizational level, examples for knowledge transfer instruments include the Experience Factory concept [3] that aims to facilitate knowledge transfer between knowledge workers in a software engineering context. On a technological level, examples include knowledge management systems [4] and knowledge infrastructures [5]. Sociological approaches to knowledge transfer facilitation include, for example, communities of practice [6].

Many barriers are known that can influence and impair the effectiveness of knowledge transfer instruments. The contribution by [7], for example, reports on the difficulties of transferring knowledge between software development teams. By studying a deployed *lessons-to-learn* database for software engineering knowledge, they were able to identify obstacles concerning the knowledge transfer instrument such as *lack of awareness*, *low information quality, low usage* and *time-consuming maintenance*. From an organizational perspective, work in [8] reports on pitfalls of knowledge management initiatives including *failure to align KM with the organizations' strategic goals, creation of repositories without explicitly defining the intentions behind them, or failure to understand and connect KM to individuals' work activities*.

As a consequence, the *effectiveness* of knowledge transfer instruments appears to depend on the *stakeholders* that participate and share an interest in knowledge transfer (such as software developers or managers), and on their *acceptance*, *motivation* and *goals*. Therefore, analyses of knowledge transfer effectiveness need to take the context of knowledge transfer instruments, the stakeholders and their goals, into account. However, in practical settings goals may often be *unclear*, *unknown*, *implicit* and *conflicting* among stakeholders. While knowledge officers [4] want to collect experiences of past projects carefully, software developers typically must deliver software quickly - thereby creating a potential conflict of goals. As a result, analyzing the goals from different interdependent stakeholders is difficult, but represents a critical challenge for knowledge management (as for example claimed by ([9], page 13). In order to be able to analyze the effectiveness of knowledge transfer instruments, we need to analyze not only the goals of stakeholders, but how those stakeholders depend on each other to achieve them.

Based on these observations, this work introduces an *agent-oriented modeling approach* for *analyzing effectiveness of knowledge transfer instruments*. Agent orientation here refers to the *explicit focus on* and *modeling of actors* and their *goals*. While *modeling* approaches already exist for analyzing knowledge transfer (such as KODA [10] and B-KIDE [11]), modeling the *goals and intentions* of knowledge transfer participants and instruments has received little attention so far. However, the explicit modeling and analysis of goals and dependencies between different stakeholders has been investigated in the area of requirements engineering for more than a decade [12], [13]. Reports concerning the application of these ideas are available [13], leading one to the question if these approaches may be applicable in the domain of knowledge management as well. In parallel, advances in computational organization theory have inspired similar research efforts on agent-based modeling of knowledge flows from a different research perspective (such as [19]).

The overall contribution of this work is the introduction of an agent-oriented modeling approach for analyzing knowledge transfer effectiveness *in the light of stakeholders' goals*. After briefly reviewing related research and explaining the basic approach, we introduce a novel modeling method by extending the agent-oriented i* modeling framework [13]. We then apply the approach to a widely established knowledge transfer instrument: the *Experience Factory* concept [3]. Finally, we discuss our achievements and draw implications for future research.

2. Context of this work

Research in the domain of knowledge transfer focuses on the development and evaluation of *theories*, *modeling languages* and *instruments* for knowledge transfer.

Theories for knowledge transfer aim to satisfy the epistemological need for understanding and explaining the *nature of knowledge transfer* itself (such as [14] and [21]). In an organizational context, theories for knowledge transfer aid the understanding of the nature of knowledge relations in and across organizations on a conceptual level. *Knowledge flow theory* [14], [19] provides a classification of different types of knowledge transfer in organizations and a way for representing the transformations that are pursued by KM initiatives (through *vector representation*). In [21], knowledge transfer is theorized by distinguishing between situational, source, transfer, relational, recipient, utilization and organizational context. The contribution by [22] investigates factors that affect the success of R&D knowledge transfer. Further research has been done on conceptualizing the relationship between work and knowledge processes ([23] and [11]). **Modeling languages** aid the identification and visualization of concrete knowledge relations among specific organizational entities (such as specific individuals, roles, groups, departments, etc). Existing approaches include, for example, KODA [10] or B-KIDE [11]. **Instruments** for knowledge transfer aim to improve on and facilitate different aspects of knowledge relations. Technological instruments for knowledge transfer include synchronous and asynchronous communication tools such as wikis, discussion boards, or expert locators. Organizational instruments include for example mentoring, experience factories [3] and job rotation [17]. On a social level, communities of practice represent a prominent example for facilitating knowledge transfer among a group of people [6].

3. Overall approach

Not only do the participants of knowledge transfer have intentions, but knowledge management initiatives in general are driven by goals that are attributed by stakeholders. The *experience factory* can be regarded to represent a knowledge management initiative that pursues the goal of *facilitating the transfer of knowledge and experiences across software development teams and projects*. As a consequence, the instruments that are employed in organizations for knowledge management purposes can be regarded to represent *intentional actors*. They are *intentional* because they pursue assigned goals (assigned by actors such as management, business analysts or knowledge officers), and they represent *actors*, because KM instruments can exhibit *active* behavior to a certain extent (the Experience Factory concept for example exhibits active behavior by mediating and facilitating knowledge transfer between software developers, and pro-actively contacting them). On a technological level, examples for such *intentional actors* include workflow management systems, expert locators, recommender systems or computer supported collaborative work (CSCW) systems with notification functionalities.

But what are the concrete benefits of conceptualizing knowledge transfer instruments such as the Experience Factory *as intentional* actors*?* Relating the goals of knowledge transfer instruments (such as the experience factory) to the goals of other organizational actors (such as software developers, knowledge officers) promises to contribute to the analysis of knowledge transfer effectiveness in multiple ways: First, making intentions of instruments for knowledge transfer explicit aids in *reasoning and arguing* about them. A lack of understanding about goals is regarded to be a major reason for failure of KM initiatives ([9], page 13) and the creation of repositories without explicitly defining the intentions behind them is regarded to represent a common pitfall [8]. Therefore, explicating the goals of stakeholders *and* knowledge transfer instruments seems to be a necessary and sound starting point for analysis. Second, reasoning about them allows for *evaluating different degrees of goal satisfaction* among different stakeholders, thereby taking situational context of knowledge transfer into account. This can be expected to aid in analyzing the effectiveness of a set of alternative knowledge transfer instruments in light of potentially conflicting goals. Third, by making the relations between stakeholders' goals and knowledge transfer instruments explicit, it can be expected that the *how* and the *why* knowledge instruments *work* or *fail* can be made visible.

Having an agent-oriented modeling approach for analyzing effectiveness of knowledge transfer instruments available therefore can be expected to yield the following benefits:

1. Effectiveness of knowledge transfer instruments in specific situations can be *analyzed before deploying* them in their environments, thereby potentially reducing costs of errors or experimentation and facilitating the *process of design*.
2. Alternatives to available knowledge transfer instruments can be *formally explored* rather than randomly identified, thereby potentially facilitating a *process of improvement*.

Approaches for the development of models that reflect relationships between *goals*, between *agents*, and between *goals* and *tasks* exist (such as goal-interdependency graphs [12], and strategic dependency and strategic relationship diagrams [13]). In the domain of requirements engineering, these approaches have been applied, for example, to justify software designs, to explore and select among design alternatives, and to trace features back to higher level system goals. In the following section, we first introduce the agent-oriented modeling framework i*, and subsequently illustrate an approach to analyze the effectiveness of knowledge transfer instruments on three different levels of detail.

4. An agent oriented approach to analyzing knowledge transfer

In the following, we introduce selective aspects of the agent-oriented i* modeling framework that acts as a conceptual foundation for this work. For more comprehensive background information about the i* framework and about research on agents we make a reference to available literature [13].

4.1. The i* framework

The i* framework for modeling strategic relationships between actors was chosen as the fundament for this work because of its ability to 1) model both internal and external aspects of actors by means of *strategic dependency (SD)* and *strategic rationale (SR) models*, 2) model common concepts such as *softgoals*, *goals*, *tasks* and *resources* and 3) reason about modeled goals by means of *goal evaluation algorithms* [15].

The i* Framework does not explicitly provide elements for modeling aspects of knowledge transfers, but provides a series of useful concepts to build upon: In i*, *actors* are represented as *agents*, *roles or positions*. SD models depict *goal, task*, *resource and softgoal dependencies* between actors. SR models contain *goals*, *tasks*, *resources* and *softgoals* of specific actors that are related to each other through *task-decomposition* and *means-ends links*. Figure 1 introduces the elements of the i* framework and their corresponding graphical representations, which are relevant in the context of this work.

Keeping the need for simplicity in modeling constructs in mind, minimal extensions to i* are introduced in order to be able to identify and model knowledge dependencies and supportive means as a basis for an agent- and goal-oriented representation of knowledge transfer instruments. Extensions focus on both the modeling language (the way of modeling [16]) and the modeling method (the way of working [16]). While i* provides a comprehensive set of elements as part of its modeling language, it leaves the modeling method open [13].

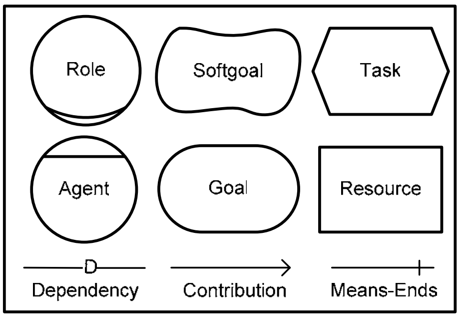

Figure 1 Selected elements of the i* framework

In the following sections, we introduce the basic *way of working* and *modeling* [16] for the development of agent-oriented models of knowledge transfer. We introduce the *Knowledge Transfer Agent (KTA) Modeling Method* which describes *how* to create models of knowledge transfer by utilizing *modeling extensions* and standard i* framework modeling elements.

4.2. The KTA modeling method

The KTA modeling method consists of three distinct levels, each one increasing the level of detail and analysis possibilities. We assume that in practical settings analysts may adopt different levels of analysis, according to the requirements of specific situations. Whereas quick, high-level analysis might only require level 1, more thorough and detailed analysis efforts might follow through all three proposed levels of the KTA modeling method. Because of sequential dependencies between the levels, level 2 analysis implies conducting level 1 analysis, and level 3 analysis implies conducting level 1 & 2 analyses.

Level 1 - Identification of knowledge dependencies: In this first level, strategic *knowledge dependencies* between different *actors* (such as software developers, management, customers) are identified. Questions for identifying *knowledge dependencies* include: *Who do actors turn to for advice and/or expert knowledge? Who turns to them for advice? What kind of knowledge is involved?* Integrating the answers to these questions into i* models results in a set of identified strategic *knowledge dependencies* between a set of *actors*, as depicted in Figure 2.

In a **knowledge dependency**, a *depender depends on a dependee to provide knowledge* (in for example the form of advise). In contrast to resources, **knowledge** *refers to skills, heuristics and experiences of intentional actors*. With a *knowledge dependency*, the *depender* has access to *knowledge* he would otherwise not have, but at the same time becomes vulnerable when the *dependee* fails to provide that knowledge. The *dependee* is assumed to be able to provide the kind of *knowledge* to the *depender*, while the *depender* is assumed to be able to apply it. In order to be able to take an explicit knowledge perspective on agent-oriented i* models, KTA introduces *knowledge* and *knowledge dependencies* as modeling extensions to the i* framework.

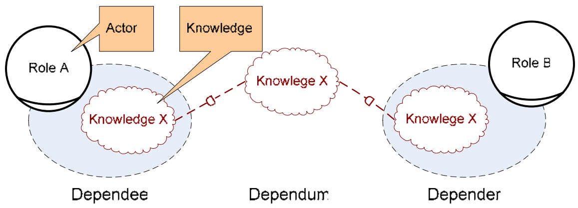

Figure 2: Level 1 - knowledge dependencies

In an Experience Factory context, having explicit information about knowledge and knowledge dependencies between actors allows for answering questions such as: *What kind of knowledge do software developers need in order to achieve their goals? What kinds of knowledge dependencies exist between software developers and other members of the organization? How does failure of knowledge dependencies impair the ability of, for example, software developers to achieve their goals?*

This kind of information can already be useful for organizations from a knowledge management perspective. It can help, for example, in identifying *knowledge risks* in organizations (knowledge dependencies that are likely to fail) or knowledge networks and communities (participants in a specific knowledge dependency). However, specific information about the applied knowledge transfer instruments involved in this knowledge dependency cannot be obtained at this stage. To further detail the notion of knowledge dependencies we therefore need to model more specific information about them, which is addressed in level 2:

Level 2 - Identification of supportive means per knowledge dependency: At the second level, the supportive means that are utilized by the participants of a specific knowledge dependency are identified. Questions for identifying these means include: *How does knowledge transfer take place? What kind of communication channels and storage objects are involved?* To model these aspects of knowledge transfer, we introduce *communication channels* (such as face-to-face communication, phone) and *storage objects* (such as documents, systems, binders) based on the existing modeling approach B-KIDE [11] from the domain of knowledge management (see Figure 3).

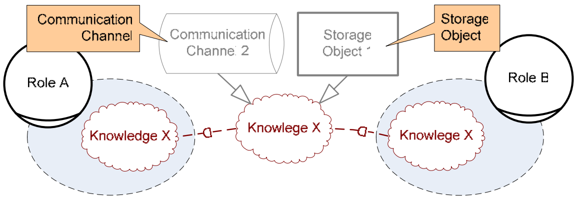

Figure 3: Level 2 - supportive means

A **communication channel** *represents an instrument that is used for the intentional transfer of knowledge from a sender to a receiver, where the receiver can be assumed to reliably receive it.* A **storage object** *represents an instrument that is used for storing information, where the stored information is available to others* [11]. The weakest form of knowledge transfer thus is when senders *store* information in a publicly available space where receivers can, but do not necessarily have to, access it (a network drive would be an example for such a storage object). The stronger notion of *communication channels* assumes the delivery of the information to the receiver, but does not assume that the receiver is able to apply or comprehend it (for example, a pager). It is only on a knowledge dependency level that it is assumed that the receiver is able to integrate this information in her body of knowledge, using available *storage objects* and *communication channels* to obtain it. *Storage objects* and *communication channels* are connected to *knowledge dependencies* through *means-ends links*, indicating that *different means* can be used to address the same *knowledge dependency* (as depicted in Figure 3 on a conceptual level). In that sense, means-ends relationships represent a starting point for analysts to *explore alternative means* that are capable of achieving the same ends.

In an Experience Factory context, having explicit information about available communication channels and storage objects of knowledge dependencies allows for answering questions such as: *How is knowledge being transferred between software developers, using which communication channels? How is experience being stored (or packaged)? What notion of knowledge transfer (strong vs. weak) is employed? What would be alternatives to existing knowledge transfer instruments?*

This kind of information can help in understanding *how* knowledge is being transferred between software developers, and gives an overview of the broad range of knowledge transfer instruments currently employed in organizations. Exploring new means for the same ends helps in exploring alternative knowledge transfer instruments. However, at this level knowledge transfer instruments are still modeled as non-intentional entities – having no goals themselves, while in practice these instruments typically *serve a purpose* (and thereby pursue assigned goals). In order to be able to analyze success or failure of the goals of these knowledge transfer instruments, we propose to re-conceptualize the identified knowledge dependencies and supportive means as intentional actors themselves, - thereby introducing new agents to the developed models.

Level 3 - Re-conceptualization of supportive means as agents: Now that supportive means are identified, the final level of the KTA modeling method suggests re-conceptualizing *knowledge dependencies* and corresponding *supportive means* as a distinct agent (a so-called *Knowledge Transfer Agent*), pursuing its own *goals* and having its own specific dependencies that reveal the circumstances *under which it can fail* (depicted in Figure 4).

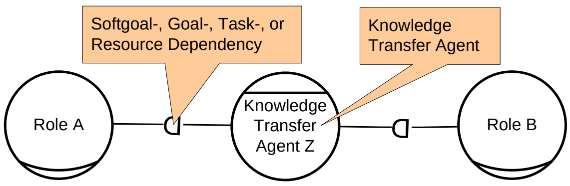

Figure 4: Level 3 - knowledge transfer agents

Questions for identifying goals and dependencies of the newly introduced agents include: *What are the (both explicitly and implicitly) assigned goals of knowledge transfer agents? Who assigned these goals? How do knowledge transfer agents achieve their goals? On what and whom do they depend on in order to work?* As a consequence, the identified and modeled knowledge dependencies are replaced by newly introduced *Knowledge Transfer Agents*. Now, the knowledge transfer instruments themselves can be analyzed in terms of goal achievement. A **knowledge transfer agent** *is an intentional human, organizational or technological actor that focuses on the facilitation of knowledge transfer between two or more other actors.* Note that on this level, the need for additional modeling elements (such as *knowledge dependencies* and *knowledge*) no longer exists. Decomposing the knowledge transfer agent into standard i* elements such as *goals, tasks and resources*, explains how the knowledge transfer agent is expected to facilitate knowledge transfer and to achieve its goals.

From level 3, three main implications follow: 1.) The introduced modeling extensions of level 1 and 2 constitute a necessary, *analytical and methodical instrument* for *taking a knowledge perspective* and *identifying* knowledge dependencies between actors 2.) Level 1 and 2 of the KTA modeling method therefore *guide* and *focus* potential subsequent modeling activities from a knowledge perspective and 3.) By modeling knowledge transfer agents based on i*'s standard notation, the goal evaluation algorithms that have been applied in the domain of requirements engineering can now be applied to questions of knowledge management as well.

In an Experience Factory context, re-conceptualizing knowledge transfer instruments as intentional actors (knowledge transfer agents) increases the ability of analysts to analyze effectiveness and reasons for success or failure of the experience factory concept, and analyze its impact on corresponding stakeholders (such as software developers and management) and their goals.

So far, we have only coarsely illustrated the application of the KTA modeling method in the context of software engineering. In the following, we will apply the method in greater detail, modeling the experience factory concept with the KTA modeling method and analyzing it in terms of its effectiveness and potential improvements.

5. The experience factory case

To investigate the viability of the introduced concepts, we will now apply all three levels to a widely discussed and established example from the domain of software engineering: The *Experience Factory* concept that focuses on the facilitation of knowledge transfer between software developers. We aim to demonstrate that by applying the KTA modeling method to the experience factory concept, we are able to conduct analyses of knowledge transfer instruments on three different levels. We aim to demonstrate that the KTA modeling method can address questions that are difficult to answer with more traditional approaches, which do not focus on the modeling of goals and intentions (such as B-KIDE [11] or KODA [10]). Questions we intend to address include: *Under which conditions can the experience factory concept fail, and how does failure affect the goals of corresponding stakeholders?* And *How can adaptations to the experience factory concept be explored in a structured way - taking the constraints and goals of different stakeholders into account?*

To answer these questions, we first model and represent the experience factory as a knowledge transfer agent and subsequently analyze this agent in terms of potential problems and vulnerabilities.

5.1. The experience factory as a knowledge transfer agent

Experience factories [3] emerged from the software engineering research community as a reaction to the increasing importance and role of experience in software development. Based on the observation that software development is a *creative, knowledge-intensive activity* rather than a *repetitive, manufacturing activity* ([3], page 2) and that its main capital is *intellectual* rather than *physical* [17], experience factories have been proposed as *separate organizational entities* designed to facilitate the *reuse and transfer of experiences* among software development teams. In that sense, the experience factory concept constitutes a *knowledge transfer agent* conforming to our definition, by representing an actor (the "separate organizational entity") that has goals ("increase knowledge reuse") and focuses on the facilitation of knowledge transfer between two or more other actors ("software development teams"). In a more formal way, the goals pursued by the experience factory concept can be described by utilizing *knowledge flow theory* [18], [19],: Let a = a_1e + a_2r + a_3l within the coordinate system e = expliciteness, r = reach and l = lifecycle then the goal of the experience factory can be expressed as the vector $\overrightarrow{AB}$ with *A* = (implicit, group, share) and *B* = (explicit, group, share).

In the following we will model key aspects of the experience factory concept by using the introduced KTA modeling method on all three levels, while highlighting some aspects of the experience factory concept that are relevant to knowledge transfer and excluding others (such as different techniques to package experiences).

5.2. Applying the KTA modeling method

According to the **first level** analysis of the proposed method, the dependencies between software developers can be modeled as the *knowledge dependency* Knowledge about Past Projects between the two *roles* Experience Consumer and Experience Provider, both representing software developers. As depicted in Figure 5, having *knowledge dependencies* modeled can aid in identifying knowledge risks and their impact on goal satisfaction. In this example, it can be observed that a lack of transfer regarding Knowledge about Past Projects would impair the Experience Consumer's ability to achieve his goal Develop and Maintain Software.

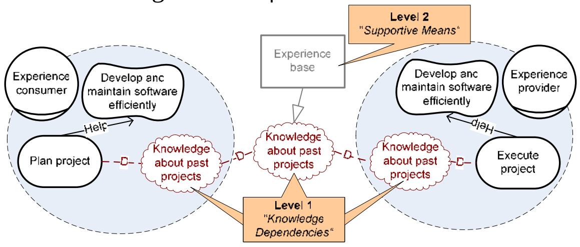

Figure 5 Applying level 1 and 2 of the KTA method to the experience factory example

At the **second level**, a *supportive means* (the *storage object* Experience Base) is identified and added to the model. By looking at Figure 5, analysts can identify that the *storage object* Experience Base represents one (but only one!) *means* to address the identified *knowledge dependency* Knowledge about Past Projects. Analysis might yield to the exploration of other *supportive means*, such as mentoring, coaching, project weblogs or communities of practice to address the identified *knowledge dependency* Knowledge about Past Projects. Transforming these competing and/or complementary knowledge transfer instruments into *agents* helps in analyzing their effectiveness and choosing alternatives that best suit the corresponding stakeholders' goals.

At the **third level**, the *knowledge dependency* Knowledge about Past Projects and its supportive means Experience Base are transformed into the *knowledge transfer agent* Experience Factory (depicted in Figure 6), by asking: *What are the (both explicitly and implicitly) assigned goals of the Experience Factory? How does it achieve its goals? On what and whom do they depend on in order to work?* From a strategic rationale modeling perspective (modeling the agent's internals), the Experience Factory can be considered to pursue one overall *goal*, execute three main *tasks*, and maintain one critical *resource* (simplified, based on [3]). The overall *goal* of the experience factory concept is to Facilitate Inter-Project Experience Transfer. The *task* Provide Project Support is concerned with learning about project settings and providing Experience Packages to Experience Consumers as a *resource*. Experience Packages are collected in an Experience Base that represents a container for all Experience Packages available. The second *task* Analyze Projects is concerned with collecting Data and Lessons Learned *resources* at the end of each project, and *packaging* them into experience packages (the third *task*). By having transformed the Experience Factory concept into a *knowledge transfer agent*, existing goal evaluation algorithms from the domain of requirements engineering can be applied to investigate the effectiveness of the experience factory concept as a knowledge transfer instrument. This will be demonstrated in the next section.

5.3. Performing goal evaluation

Now that the experience factory is conceptualized as an agent, strategic dependencies are made explicit and can be reasoned about. This aids in answering questions such as *"Under which conditions can the Experience Factory concept fail?"* As it is illustrated in Figure 6, the Experience Factory has *dependencies* that, if not satisfied, cause the Experience Factory to fail. In Figure 6, two such *dependencies* can be identified (on a strategic dependency level): **A)** The Experience Factory *depends* on Experience Consumers to provide information about Project Characteristics in order to Provide Project Support and **B)** the Experience Factory *depends* on Experience Providers to Provide Experiences. If either Experience Providers or Experience Consumers fail to perform these activities, the Experience Factory's ability to achieve its goals would be impaired.

Goal evaluation algorithms aid in formalizing such reasoning processes. The goal evaluation algorithm introduced by [15], for example, is based on the assignment of qualitative evaluation labels to elements of i* models and consists of 1) the assignment of initial evaluation values aimed towards asking an interesting domain question and 2) the propagation of these initial evaluation values through the network of actors using a combination of propagation guidelines and human judgement. Typically, judgement is performed by domain experts that have knowledge about typical failures or problems that may occur in the respective domain. More in-depth information on the applied goal evaluation algorithm can be found in [15]. The algorithm introduces qualitative evaluation labels on a six point scale that range from satisficed, partially satisficed, conflict, unkown, partially denied to denied. Applying such a goal evaluation algorithm to the experience factory concept would start with the initial assignment of one or more evaluation label.

When, for example, an Experience Provider is not willing to or does not Provide Experience, the goal evaluation algorithm would require analysts to assign the initial qualitative value *denied* to the corresponding task Provide Experience (depicted as a cross in a dashed circle in Figure 6).

This corresponds to formulating the question *"How would a deficiency in providing experience affect the goals of the Experience Factory and the goals of Experience Consumers and Providers?"*

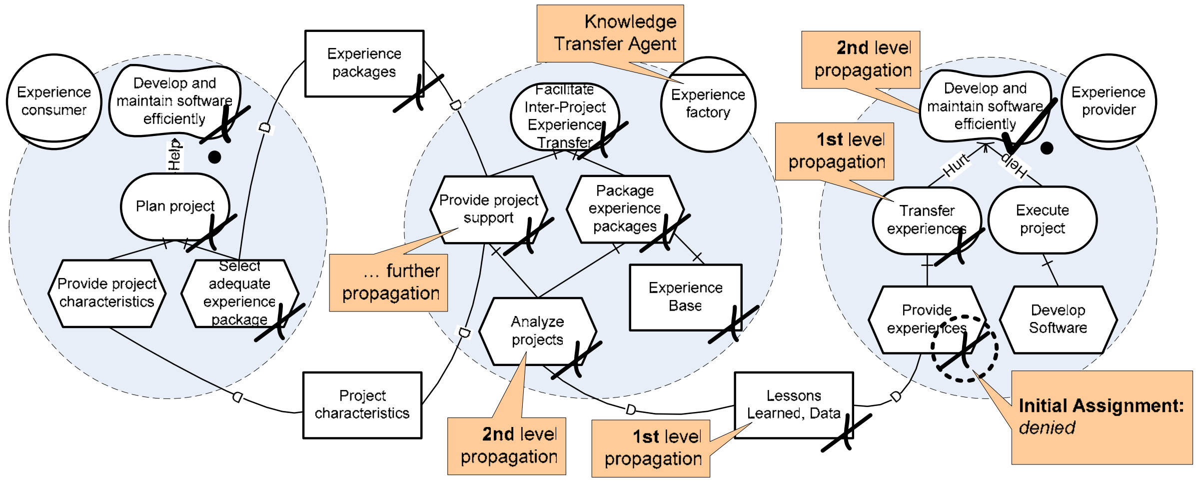

Figure 6: Level 3 - The experience factory concept as a knowledge transfer agent including evaluation

As illustrated in Figure 6, *first level propagation* of this initial assignment would result in denying the goal Transfer Experiences of the Experience Provider and denying the resource dependency Lessons Learned, Data. Further propagation of these evaluations reveals that a deficiency in providing experience significantly impairs the experience factory's ability to achieve its goal Facilitate Inter-Project Experience Transfer. As a consequence of further propagation, the Experience Consumer is impaired in his ability to Select Adequate Experience Projects, and as a further consequence, is partly impaired in his ability to Develop and Maintain Software efficiently (depicted in Figure 6 as a partly denied goal). However, the Experience Provider can potentially better achieve his goal Develop and Maintain Software, because of more time available (depicted in Figure 6 as a partly satisficed goal ✓). By conducting that kind of analysis, the effects of failure and criteria for success of knowledge transfer instruments can be explored and analyzed in detail.

5.4. Analysis and results

Having agent- and goal-based models of knowledge transfer instruments and their contexts aids reasoning about them in an explicit way. *But do the conclusions obtained from such reasoning processes yield reasonable results and conclusions?* To answer that question, we investigate research reports of failures, problems and obstacles with the experience factory concept that point to situations in which experience factories are impaired in achieving their assigned goals (which corresponds to ineffectiveness).

Current research (such as [7] and [17]) reports on the following problems: *"Project managers who feel they need to focus on completing their current project on time, and not on helping the next project manager succeed, often consider this* [providing and documenting experience] *a burden."* or *"Technology's fast pace often discourages software engineers from analyzing the knowledge they gained during the project, believing that sharing the knowledge in the future will not be useful."* ([17], page 36).

These issues can clearly be related to the previously identified dependency **B** (Provide experiences). The effects of these issues significantly impair the experience factory's ability to achieve its overall goal - as illustrated by the application of the KTA modelling method and the qualitative goal evaluation algorithm. However, hardly any problems are reported with the identified dependency **A** (Provide information about project characteristics). By investigating this dependency in the agent-oriented models in greater detail, it can be seen that experience providers have an immediate benefit of providing information about project characteristics (which is: receiving experience packages in return), while experience providers do not receive immediate payback for their investment. Having the introduced agent-oriented models of knowledge transfer instruments aids analyses of such circumstances and aids reasoning about improvement alternatives. One potential improvement option can be the introduction of incentive programs for experience providers, as suggested by [9]. Having incentive programs available would introduce a new goal to the experience providers' set of goals, and would make the provision of experiences more attractive for them (unless there are other conflicting goals). Reapplying the goal evaluation algorithm to a new version of the model which includes this improvement potential as a new goal would show that incentive programs are suitable to increase motivation for Experience Providers and therefore contribute to the goals of the Experience Factory (given that no other conflicting goals exist), but still may hurt the goal Develop and Maintain Software of Experience Providers, by taking up additional time. Another alternative would be using different techniques for packaging experiences [20] or changing the knowledge flow "vector" [18] of the experience factory concept. Selecting another packaging technique would have the potential to balance the quality of experience packages with the efforts necessary for creating them, thereby satisfying the goals of the experience factory agent, experience users and experience providers.

5.5. Limitations

By applying the KTA modeling method to the experience factory concept, we aimed to demonstrate its expressiveness and analytical power. However, the method has only been applied on a conceptual level (vs. a concrete implementation of the experience factory in a specific company), and only conclusions that are already known to the respective research community were drawn. Furthermore, this explorative contribution assumed validity of developed agent-oriented models and therefore did not elaborate on validation issues. While we could exemplarily demonstrate that general issues of the experience factory can be identified and analyzed with our method, its application in concrete settings might require more complex modeling efforts.

6. Conclusions

To a certain extent, agent oriented concepts have already been applied to knowledge management [24], [25]. However, only little research has been done on applying intentional concepts to the *modeling and analysis of knowledge transfer instruments*. The contributions of this work represent a first step towards a conceptualization of knowledge transfer instruments as *agents*, and thereby *opens up* the possibility of applying existing goal oriented evaluation algorithms from the domain of requirements engineering. As we have demonstrated, conducting goal evaluation allows for explicitly reasoning and arguing about effectiveness and vulnerabilities of knowledge transfer instruments. Current research in part focuses on empirically examining the "*relative performance of alternate designs* [for knowledge transfer]" [18]. By being able to model and reason about alternate knowledge transfer instruments, the KTA modeling method aids in selecting *effective* knowledge transfer instruments in the light of different stakeholders' goals. However, we expect that analysts applying our method will not always conduct analyses on all three levels, but will focus on the level of detail that is adequate for dealing with the questions that need to be addressed in a specific situation.

7. Contributions and future work

The KTA modeling method introduced in this work makes the following contributions: First and foremost, it contributes to analyzing effectiveness of knowledge transfer instruments in the light of (potentially conflicting) stakeholders' goals. By introducing different levels of detail, the KTA modeling method aims to satisfy the need for different levels of analysis – allowing for both quick, high-level analysis and also more thorough, in-depth investigations of knowledge transfer instruments. It allows for analyzing *how* knowledge transfer instruments work and achieve their goals, and *why* they can succeed or fail. The experience factory case has illustrated the usage of the KTA modeling method for that purpose. In addition, this work represents a contribution to the domain of requirements engineering by introducing a step-wise method that demonstrates how knowledge management problems (level 1 & 2) can be transformed into requirements problems (level 3). In that sense, the KTA modeling method can help requirements engineering efforts to explicitly take a knowledge perspective on knowledge intensive environments.

However, more research needs to be done. First, future work should focus on analyzing knowledge transfer instruments not only on a conceptual, but on a deployed level. This can include for example investigations of organizations that have a running Experience Factory in place. Furthermore, the implications of taking a knowledge perspective on requirements engineering problems were not a major issue in this work. In order to further investigate the contribution of the KTA modeling method to the requirements engineering community, it needs to be applied and investigated in further requirements engineering contexts.

Acknowledgements: The research of this contribution is funded in part by the Austrian Competence Center program Kplus, the FWF Austrian Science Fund (within an Erwin Schrödinger Fellowship Program) and the Natural Sciences and Engineering Research Council of Canada.

8. References

[1] M. Alavi and D. E. Leidner, "Knowledge management and knowledge management systems - conceptual foundations and research issues," MIS Quarterly, vol. 25, no. 1, 2001.

[2] K. Joshi, S. Sarker, and S. Sarker, "Knowledge transfer among face-to-face information systems development team members: Examining the role of knowledge, source, and relational context," in Proceedings of the 37th Hawaii International Conference on System Sciences, 2004.

[3] V. R. Basili, G. Caldiera, and D. Rombach, "Experience Factory", Encyclopedia of Software Engineering. Wiley & Sons, 1994.

[4] R. Maier, Knowledge Management Systems. Springer Verlag Berlin, 2002.

[5] Y. Sivan, "Nine keys to a knowledge infrastructure: A proposed analytic framework for organizational knowledge management," Research Paper, March 2001.

[6] E. Wenger, Communities of Practice: Learning, Meaning, and Identity. Cambridge University Press, 1999.

[7] S. Komi-Sirviö, A. Mäntyniemi, and V. Seppänen, "Toward a practical solution for capturing knowledge for software projects," IEEE Software, vol. 19, no. 3, May 2002.

[8] M. A. Fontaine and E. Lesser, "Challenges in managing organizational knowledge," IBM Institute for Knowledge Based Organizations, Tech. Rep., 2002.

[9] G. Lawton, "Knowledge management: Ready for prime time?" Computer, vol. 34, no. 2, pp. 12–14, 2001.

[10] A. Abecker, K. Hinkelmann, H. Maus, and H. Müller, Geschäftsprozessorientiertes Wissensmanagement. Springer, Berlin, 2002.

[11] M. Strohmaier, "B-KIDE: A framework and a tool for business process oriented knowledge infrastructure development," Ph.D. dissertation, Graz University of Technology, Austria, 2004.

[12] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, 2000.

[13] E. Yu, "Modelling strategic relationships for process reengineering," Ph.D. dissertation, Department of Computer Science, University of Toronto, 1995.

[14] M. E. Nissen, "An extended model of knowledge-flow dynamics," Communications of the Association for Information Systems, vol. 8, pp. 251– 266, 2002.

[15] J. Horkoff, "Using i* models for evaluation," Master's thesis, University of Toronto, 2006.

[16] B. Hommes and V. van Reijswoud, "Assessing the quality of business process modelling techniques," in Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000.

[17] I. Rus and M. Lindvall, "Knowledge management in software engineering," IEEE Software, vol. 19, no. 3, May 2002.

[18] R. Ibrahim and M. E. Nissen, "Developing a knowledge-based organizational performance model for discontinuous participatory enterprises," in Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.

[19] M. E. Nissen and R. E. Levitt, "Agent-based modeling of knowledge flows: Illustration from the domain of information systems design," in Proceedings of the 37th Hawaii International Conference on System Sciences, 2004.

[20] M. Lindvall, M. Frey, P. Costa, and R. Tesoriero, "Lessons learned about structuring and describing experience for three experience bases," in LSO '01: Proceedings of the Third International Workshop on Advances in Learning Software Organizations. London, UK: Springer-Verlag, 2001, pp. 106–119.

[21] R. Peinl, "A knowledge sharing model illustrated with the software development industry," in Proceedings of the Multikonferenz Wirtschaftsinformtik (MKWI 2006), Passau, Germany. GITO Verlag, 2006.

[22] J. L. Cummings and B.-S. Teng, "Transferring R&D knowledge: the key factors for affecting knowledge transfer success," Journal of Engineering and Technology Management, vol. 20, pp. 39–68, 2003.

[23] U. Remus, "Prozeßorientiertes Wissensmanagement – Konzepte und Modellierung," Dissertation, Wirtschaftswissenschaftliche Fakultät der Universität Regensburg, Regensburg, Deutschland, 2002.

[24] R. Guizzardi, "Agent oriented constructivist knowledge management," Ph.D. dissertation, University of Twente, 2006.

[25] X. Deng, "Intentional modeling for enterprise architecture," Master's thesis, University of Toronto, 2006.