



Evaluation of WfMC Awards for Case Management: Features, Knowledge Workers, Systems

Johannes Tenschert^(✉) and Richard Lenz

Institute of Computer Science 6 (Data Management),
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
{Johannes.Tenschert,Richard.Lenz}@fau.de

Abstract. In recent years, many production case management (PCM) and adaptive case management (ACM) systems have been introduced into the daily workflow of knowledge workers. In many research papers and case studies, the claims about the nature and requirements of knowledge work in general seem to vary. While choosing or creating a case management (CM) solution, typically one has the target knowledge workers and their domain-specific requirements in mind. But knowledge work shows a huge variety of modes of operation, complexity, and collaboration. We want to increase transparency on which features are covered by well-known and award-winning systems for different types of knowledge workers and different classes of systems. This may not unveil gaps between requirements and offered solutions, but it can uncover differences in solutions for varying user bases. We performed a literature review of 48 winners of the WfMC Awards for Excellence in Case Management from 2011 to 2016 and analyzed case studies in regard to targeted knowledge workers, advertised features, and type of system. Different types of knowledge workers showed a different bias on certain system types and features in regard to collaboration and variability of processes.

Keywords: Adaptive case management ·
Production case management · Knowledge-intensive business process ·
Knowledge work

1 Introduction

In recent years, many PCM and ACM systems have been introduced with the goal of improving the efficiency and quality of knowledge work. Typically, one may find sentences like “Most knowledge workers spend their time in business applications like Salesforce.” [1], “Due to [...] and the high degree of interactivity between knowledge workers [...]” [2], “If knowledge workers can rely on [...] they can provide simplified automated process fragments without worrying about all possible exceptions” [3]. Each of these articles may be right with their assumptions for target users. But is this really true for knowledge work in general?

According to Davenport [4], knowledge work can be classified into four groups by complexity and level of interdependence (i.e. required collaboration). He distinguishes the transaction, integration, expert, and collaboration model [4, p. 27]. The transaction model (\downarrow collaboration, \downarrow complexity) covers work reliant on formal procedures and training, i.e. routine work. The expert model (\downarrow collab., \uparrow compl.) is judgment-oriented and highly reliant on individual expertise and experience. The integration model (\uparrow collab., \downarrow compl.) covers repeatable work reliant on formal processes but also dependent on integration across functional boundaries. The collaboration model (\uparrow collab., \uparrow compl.) covers complex improvisational work and relies on deep expertise across functions. This already suggests that different types of knowledge workers may have different requirements in regard to collaboration, variability in processes, and management of case data.

There are many ways to classify CM systems, e.g. by area of operations: CRM, ERP, ECM, issue tracker. We apply a condensed categorization of [5] that introduces seven categories ranging from predictable, repeatable to variable and unique processes. We distinguish BPM, PCM and ACM. BPM systems cover predictable and repeatable work. PCM systems are more flexible and tailored to a particular domain. ACM systems support unexpected workflows and unstructured data, and they are useful not only in one domain. We reviewed winners of the WfMC Awards for Excellence in Case Management that were published as case studies in regard to their classes of targeted knowledge workers, advertised features, and type of system. This analysis is intended to increase transparency on which features are covered by well-known and award-winning systems. It is not intended to unveil gaps between requirements and offered solutions.

In the following sections, we briefly introduce related work outlining features and requirements in CM, and the methods for our analysis. In Sect. 4, we introduce the extracted features, and Sect. 5 presents our findings with a complete matrix and resulting analyses. Afterwards, we discuss our approach and results, and in Sect. 7 we conclude the paper.

2 Related Work

Palmer et al. [6] compared characteristics of ACM, ECM, CRM, and BPM based on typical requirements in ACM and hint which type of system to use based on business problems addressed and expected workflow. However, they do not directly address PCM systems and do not cover the interdependence of different types of knowledge workers. Matthias [7] outlines requirements of ACM, e.g. in regard to decisions, database organization, variability, and access control. Our literature review focuses on the features available in award-winning CM systems to approximate different requirements of different types of knowledge workers.

Motahari-Nezhad and Swenson [8] outline the state of the art in CM with six groups for systems and examples. We also compare classes of systems, although on a more granular level, to find similarities and differences of systems tailored to different types of knowledge workers. Hauder et al. [9] derive ten requirements for ACM based on a literature review. But in the original sources, these requirements

were probably formulated with a specific knowledge worker in mind. Hauder et al. did not find one reference implementation fulfilling all extracted requirements, which suggests that requirements of knowledge workers also differ within ACM.

3 Methods

We analyzed 48 winners of the WfMC Awards for Excellence in Case Management from 2011 to 2016 [10–15]. There were 51 award winners, but the case studies Kirtland AFB (2014), Remfry and Sagar (2015), and Texas Office of the Attorney (2015) have not been published and were therefore omitted.

We performed this analysis by evaluating each case study and extracting features elaborated in the text and figures. For each case study, we compiled a list of features with a reference to their source in the text or figure. The case studies did not always show a consistent vocabulary for the names of features and many were similar. Hence, we used more general terms for similar features, e.g. “reporting” and “dashboard” yielded “BI (analytics/reporting)”. The approach was iterative. Obviously, this method yields only features that are elaborated and deemed important by the authors, and actual systems may have a different focus. They ranged from support for (semi-)structured processes, (un-)structured artifacts, and collaboration support to non-functional features like “system of record”. Due to space constraints, some features that were not characteristic for system types and type of knowledge work are omitted. For the categorization of a case study to ACM, PCM, and BPM, we apply a condensed categorization derived from Swenson [5]. There were no clear differentiators, so many systems fall in more than one category. This also impacts aggregated statistics: Percentages are provided for *all* systems of a group.

The categorization of knowledge work is based on Davenport’s classification of knowledge-intensive processes [4]. Knowledge work is divided into four groups by complexity of work and level of interdependence (collaboration). Systems can target multiple stakeholders with a varying degree of complexity and interdependence. These dimensions were classified based on available user descriptions and if necessary on assumptions (e.g. work of lawyers and physicians typically yields the expert model). For each case study, the extracted knowledge workers, features, and type(s) of system were inserted into Table 1. We used RapidMiner¹ to find correlations between types of system and types of knowledge work, between types of knowledge work and features, between system types and features, and conditional probabilities between all attributes (knowledge work, systems, features). Due to space constraints, some analyses had to be omitted here.

4 Extracted Features

We extracted the following features by evaluating all case studies, i.e. their text and figures, for descriptions, portrayals, and occurrences of features. Structured

¹ <https://rapidminer.com>.

(aspect, complete) and semi-structured (tasks, stages) processes originated from “(semi-)structured processes” that were detected in nearly every case study.

Ad-hoc Activities. Tasks or goals that were not predefined in a process model or (task) repository and are nonetheless captured in the system as part of a case. Ad-hoc activities should cover tasks that were not anticipated at design-time.

Artifact Templates. The system supports templates for artifacts that are not only generated at case initiation. Tasks and documents are excluded from this definition. Instead, they are covered in the features *document generation* and *task templates*.

Automatic Notifications. The system provides automated messages, e.g. for warnings, alerts, or reminders.

BI (Analytics/Reporting). The system provides some dashboard or reporting for KPIs of the case. Subsumed in one feature due to similarity and intention.

Business Rules. The system enables either predefined business rules, e.g. creating a case if a specific event is detected, or user-defined rules.

Case Conversations. The system provides threaded or flat discussions for cases.

Case History. The system provides a visible activity stream of a case.

Case State. Different states of a case are emphasized and more sophisticated than *open* and *closed*. Varying from three predefined states to completely user-defined states assigned in an ad-hoc fashion.

Case Tags. The system allows to annotate cases with keywords to facilitate search.

Case Templates. In the system, cases and their contents are initiated with a template containing predefined artifacts or relations. A template may be initiated using some existing artifact, e.g. an event for a new customer or incident. Examples for predefined artifacts are a set of related tasks for a certain type of case, and a copy of an existing case with generated case-specific contents.

Checklists. The system provides a list of tasks that are typically performed or a list of attributes to typically ensure. The lists may be part of a case template/type or specific to a certain state of a case.

Collaboration within a Case. The case study emphasizes case-specific collaboration, e.g. collaborative access, collaborative workflow, chat. May differ from *notes on artifacts* in emphasizing on the collaboration aspect. A user-to-user chat that is only visible to the chatting participants can be *collaboration within a case*, but does not qualify as *case conversations* visible to all stakeholders with access rights (e.g. [16]).

Configurability. The case study emphasizes that the solution is easily configurable to end users or domains.

Deadlines. The system displays for each case pending calendar entries or deadlines for tasks or milestones.

Declarative Process Modeling. The system enables modeling (parts of) processes by stating the dependencies between tasks.

Document Generation. The system provides automatic generation of letters (e.g. MS Word, email) based on templates. This feature covers correspondence between stakeholders of a case, but not system messages indicating certain events or reminders.

Documents. The system covers document management, i.e. documents are stored and related to a case.

Domain-specific Information. The system is tailored to a specific domain, e.g. by considering certain attributes and providing a domain-specific user interface.

Guard Rails. The system prevents users from or actively reminds them to performing certain actions based on regulatory and organizational rules.

Integration with External Services. The system provides integration with existing systems in order to streamline processes and reduce redundant data entry.

Interactions of a Case. The system provides an overview of case-specific correspondence.

Meetings of a Case. The system tracks case-specific meetings and may cover meeting minutes as well.

Table 2. System type by supported knowledge worker

Given ↓ / Then →	ACM	PCM	BPM	Other	Σ
Collaboration model	100.0%	25.0%	0.0%	0.0%	4
Expert model	36.8%	84.2%	21.1%	10.5%	19
Integration model	52.9%	70.6%	5.9%	0.0%	17
Transaction model	18.2%	93.9%	27.3%	0.0%	33

Table 3. Supported knowledge workers by type of system

Then → Given ↓	Collaboration model	Expert model	Integration model	Transaction model	Σ
ACM	28.6%	50.0%	64.3%	42.9%	14
PCM	2.5%	40.0%	30.0%	77.5%	40
BPM	0.0%	36.4%	9.1%	81.8%	11
Other	0.0%	100.0%	0.0%	0.0%	2

Mobile Devices. The system provides a mobile-enabled website or an “App” for a mobile platform that connects to the system.

Notes on Artifacts. The system enables users to annotate cases or artifacts.

Queries/Buckets of Cases. The system provides predefined queries or views to display a selection of cases based on status, type, tags, assigned users or teams, stakeholders, or resource constraints.

Queries/Worklists Across Cases. The system provides predefined queries or views to display a selection of tasks based on status, assigned users or teams, priority and deadlines, or skill-based. These views are provided across cases.

Relations Between Cases. The system enables to link related cases either with or without a parent-child-relationship.

Relations Between Interactions. The system relates interactions of a case to other interactions, e.g. linking two requests. One occurrence in [2].

Resource Allocation. The system provides (automated) resource allocation, resource planning, or documentation of resource allocation.

Roles. The system allows to assign roles. If roles are case-specific, this feature overlaps with *stakeholders of a case*.

Search. The system provides some sort of search to find cases or artifacts.

Semi-structured Process (Predefined Tasks). The system provides a task repository covering typical process instances. Users may decide in which order and what tasks of this repository are performed.

Semi-structured Process (Stages). Depending on the case state (stage), the system suggests potential activities.

Stakeholders of a Case. In the system, stakeholders (e.g. clients, assigned personnel) are captured for each case. For example, specific stakeholders are assigned to tasks and artifacts, or cases contain a list of (annotated) stakeholders.

Structured Process (Aspect). The system enables completely structured and optionally automated subprocesses, e.g. predefined and related forms and tasks in a BPMN model of one aspect of the case. One aspect does not cover the whole process, i.e. it is a subgoal of the case.

Structured Process (Complete). In the system, the workflow of the case is modeled a priori and performed as a standardized process. The system might allow certain deviations from the expected workflow, but typically cases adhere to the model.

Suggestions of Activities. There are multiple courses of action to proceed in a case. The system actively suggests certain activities, interactions, or escalations, but ultimately the knowledge worker decides whether to apply the suggestion. The displayed options may be based on prioritized outcomes or restrictions.

System of Record. The article emphasizes a system of record, i.e. the system is used as and users are aware it is the authoritative data source for cases and information.

Tasks. The systems monitors case-related tasks that are either predefined or ad hoc.

Task State. The system enables different states of a task that are more sophisticated than *open* or *closed*.

Task Templates. The system provides a task library with entries that are either plain or adaptable to the context, or reuse of previous tasks as a template for the task at hand. This differs from *semi-structured process (predefined tasks)* in not modeling the tasks of the process. Obviously there is some overlap between these features and many award winners provide both.

Typed Interactions. The system distinguishes different types of requests, classifies correspondence by intention (e.g. comment, complaint, proposal) or provides typed response options to correspondence (e.g. agree, decline, counter-offer).

User Access Control. The system provides access control to cases or certain artifacts, e.g. role-based or by sharing.

Views for Multiple Stakeholders/Roles. The system offers differing views to users depending on their role in the system or case.

5 Findings

Table 1 shows for each case study which system types, types of knowledge work, and features were extracted. We generated Tables 2, 3, 4 and 5 based on this matrix. First, the evaluation confirms that different types of knowledge workers use different system classes (Table 2), and that a certain type of system may indicate the type of knowledge work (Table 3). Since there are overlaps in the user base and systems often cannot be completely attributed to one category, rows in all tables do not add to 100%. The most dominant system class is PCM. There was a high overlap between PCM and BPM systems: Only one system classified as BPMS was not also classified as PCMS. Seven ACMSs were categorized as ACMS only, the other seven also overlap with PCMS. Two case studies (Other) could not be categorized into ACMS, BPMS, or PCMS at all [17, 18], but they overlap with PCMS. Obviously, these overlaps have influence on Table 4.

Knowledge workers of the collaboration model are only covered in four case studies, so their results should be taken with a grain of salt. Nonetheless, the low number indicates that they may have been largely ignored, even though they need exactly what ACM aims to offer: Great flexibility and collaboration in one system of record. One environment is usually shared by different types of users with different requirements, but 25 of 48 of the case studies seem to be tailored to only one type of knowledge worker. The transaction model alone is represented

by 16 case studies, and 10 systems are classified for users of both the transaction and the integration model. Due to these overlaps, Table 5 cannot completely discern the features different types of knowledge workers seem to require.

The analyzed case studies in the collaboration model all provided document management (*documents*), *case conversations*, a *case state*, an activity stream of a case (*case history*), and three out of four provided support for mobile devices. However, due to the low number of case studies categorized in the collaboration model, those could be outliers. *Documents* seem to be important in every type of knowledge work. Unsurprisingly, *case conversations* are rare in the expert and transaction model. Both *case history* and *case state* appear in half of the case studies classified as expert or integration model. Unlike the feature *case state*, no case study allowed user-defined *task states*. In the integration model, there is a high emphasis on *case templates* and *collaboration within a case*. The expert model shows the highest share of systems that integrate with external services.

Most case studies provide some support for predefined structured or semi-structured processes with a complete process model, a process model of aspects of a process, predefined tasks, or task support based on the current stage of a case. All systems classified as BPMS either model the complete process or aspects of it. Obviously, ACMSs focus less on structure. Of those systems, 35.7% provide predefined tasks, and 28.6% model aspects of a process. PCMSs also model aspects of a process (47.5%) and offer predefined tasks (32.5%). For case studies classified as ACMS only, no instance had support for structured aspects.

In ACMSs, the most prevalent features seem to be *documents*, *tasks*, *collaboration within a case*, and surprisingly *roles*. Table 4 suggests that *BI (analytics/reporting)*, would be important for ACMSs as well. However, for the seven case studies classified as ACM-only, only two seem to provide this feature. The high share of ACMSs with reporting capabilities stems from ACM/PCM hybrids. Moreover, providing a *case history*, *case state*, *case templates*, notes on certain artifacts, *access control*, and *ad-hoc activities* seems to be characteristic for ACMSs. All system types have a high emphasis on being a system of record.

Ad-hoc activities are available both in ACMS (50%) and PCMS (31%). Unsurprisingly, support for document and task management seems to be important across all types of system and knowledge work. Many systems offer different views for different stakeholders (around 30% per type), but usually every stakeholder has the same view. Support for mobile devices is aligned with the type of the system: 36% of ACMSs, 18% of PCMSs and 9% of BPMSs have some support for mobile devices. Moreover, the support is highest in the collaboration model (75%) and integration model (35%), i.e. the types of knowledge work emphasizing on collaboration. For the collaboration features *case conversations*, *stakeholders of a case*, *notes on artifacts*, and *collaboration within a case*, 71% of all ACMSs had support for at least two of these features. For PCMSs, only 33% had at least two of these features. All collaboration features have the highest probability of being present in an ACMS rather than in other system types, but due to the distribution of system types, they usually imply PCMS. Unsurprisingly, they are usually present in the integration and collaboration model.

Table 4. System types and features

Feature	Feature implies system class					System class implies feature			
	ACM	PCM	BPM	Other	Σ	ACM	PCM	BPM	Other
Ad-hoc activities	44%	75%	0%	13%	16	50%	30%	0%	100%
Artifact templates	100%	0%	0%	0%	1	7%	0%	0%	0%
Automatic notifications	28%	80%	20%	8%	25	50%	50%	45%	100%
BI (analytics / reporting)	24%	91%	24%	6%	34	57%	78%	73%	100%
Business rules	24%	95%	29%	5%	21	36%	50%	55%	50%
Case conversations	64%	55%	9%	0%	11	50%	15%	9%	0%
Case history	44%	72%	22%	6%	18	57%	33%	36%	50%
Case state	47%	71%	12%	0%	17	57%	30%	18%	0%
Case tags	67%	33%	0%	0%	3	14%	3%	0%	0%
Case templates	42%	74%	16%	0%	19	57%	35%	27%	0%
Checklists	60%	40%	0%	0%	5	21%	5%	0%	0%
Collaboration within a case	47%	79%	16%	0%	19	64%	38%	27%	0%
Configurability	100%	67%	0%	0%	3	21%	5%	0%	0%
Deadlines	25%	88%	31%	6%	16	29%	35%	45%	50%
Declarative process modeling	100%	100%	0%	0%	3	21%	8%	0%	0%
Document generation	43%	100%	14%	0%	14	43%	35%	18%	0%
Documents	31%	86%	22%	0%	36	79%	78%	73%	0%
Domain-specific information	19%	97%	32%	6%	31	43%	75%	91%	100%
Guard rails	27%	87%	20%	13%	15	29%	33%	27%	100%
Integration with external services	29%	90%	26%	6%	31	64%	70%	73%	100%
Interactions of a case	44%	78%	11%	0%	9	29%	18%	9%	0%
Meetings of a case	50%	100%	0%	0%	2	7%	5%	0%	0%
Mobile devices	50%	70%	10%	0%	10	36%	18%	9%	0%
Notes on artifacts	44%	67%	17%	0%	18	57%	30%	27%	0%
Queries / buckets of cases	29%	71%	43%	0%	7	14%	13%	27%	0%
Queries / worklists across cases	43%	86%	7%	0%	14	43%	30%	9%	0%
Relations between cases	56%	67%	22%	0%	9	36%	15%	18%	0%
Relations between interactions	100%	0%	0%	0%	1	7%	0%	0%	0%
Resource allocation	14%	100%	29%	0%	7	7%	18%	18%	0%
Roles	29%	87%	23%	3%	31	64%	68%	64%	50%
Search	75%	100%	0%	0%	4	21%	10%	0%	0%
Semi-structured process (predefined tasks)	29%	76%	18%	12%	17	36%	33%	27%	100%
Semi-structured process (stages)	33%	67%	0%	0%	3	7%	5%	0%	0%
Stakeholders of a case	43%	79%	7%	14%	14	43%	28%	9%	100%
Structured process (aspect)	20%	95%	30%	0%	20	29%	48%	55%	0%
Structured process (complete)	0%	100%	78%	0%	9	0%	23%	64%	0%
Suggestions of activities	20%	80%	20%	40%	5	7%	10%	9%	100%
System of record	26%	87%	29%	3%	38	71%	83%	100%	50%
Task state	67%	33%	0%	0%	3	14%	3%	0%	0%
Task templates	50%	90%	0%	10%	10	36%	23%	0%	50%
Tasks	30%	84%	19%	5%	37	79%	78%	64%	100%
Typed interactions	60%	80%	0%	0%	5	21%	10%	0%	0%
User access control	42%	84%	16%	0%	19	57%	40%	27%	0%
Views for multiple stakeholders / roles	36%	93%	29%	7%	14	36%	33%	36%	50%
Σ						14	40	11	2

Finally, a large share (65%) of the analyzed case studies describes some sort of templating: *case templates*, *document generation*, *task templates*, and *artifact templates*. The feature *artifact templates* appears only once [19] as a generic way for content templates in a wiki. All systems providing document generation are classified as PCMS. For *document generation*, the case studies show a clear emphasis on the transaction and expert model. Except for the systems in [16, 20], all document templates seem to be predefined. In the future, this could be improved by introducing it for user-specific correspondence as well. *Case templates* are characteristic for ACMs (57%), but they are provided in PCMSs as well.

Table 5. Knowledge workers and features

Feature	Feature implies types of knowledge work					Type of knowledge work implies feature			
	Collab.	Expert	Integr.	Trans.	Σ	Collab.	Expert	Integr.	Trans.
Ad-hoc activities	13%	31%	38%	63%	16	50%	26%	35%	30%
Artifact templates	0%	0%	100%	0%	1	0%	0%	6%	0%
Automatic notifications	4%	36%	40%	68%	25	25%	47%	59%	52%
BI (analytics / reporting)	3%	47%	32%	68%	34	25%	84%	65%	70%
Business rules	5%	48%	29%	76%	21	25%	53%	35%	48%
Case conversations	36%	27%	64%	45%	11	100%	16%	41%	15%
Case history	22%	56%	44%	44%	18	100%	53%	47%	24%
Case state	24%	53%	47%	53%	17	100%	47%	47%	27%
Case tags	0%	33%	67%	33%	3	0%	5%	12%	3%
Case templates	5%	32%	63%	74%	19	25%	32%	71%	42%
Checklists	20%	20%	80%	60%	5	25%	5%	24%	9%
Collaboration within a case	11%	42%	58%	68%	19	50%	42%	65%	39%
Configurability	0%	33%	100%	100%	3	0%	5%	18%	9%
Deadlines	0%	56%	31%	56%	16	0%	47%	29%	27%
Declarative process modeling	0%	67%	33%	67%	3	0%	11%	6%	6%
Document generation	7%	64%	36%	64%	14	25%	47%	29%	27%
Documents	11%	39%	36%	72%	36	100%	74%	76%	79%
Domain-specific information	3%	55%	26%	68%	31	25%	89%	47%	64%
Guard rails	13%	33%	47%	67%	15	50%	26%	41%	30%
Integration with external services	6%	48%	23%	68%	31	50%	79%	41%	64%
Interactions of a case	11%	56%	67%	67%	9	25%	26%	35%	18%
Meetings of a case	0%	50%	100%	50%	2	0%	5%	12%	3%
Mobile devices	30%	30%	60%	70%	10	75%	16%	35%	21%
Notes on artifacts	17%	33%	50%	67%	18	75%	32%	53%	36%
Queries / buckets of cases	0%	43%	29%	86%	7	0%	16%	12%	18%
Queries / worklists across cases	14%	36%	50%	64%	14	50%	26%	41%	27%
Relations between cases	11%	67%	56%	56%	9	25%	32%	29%	15%
Relations between interactions	100%	0%	100%	0%	1	25%	0%	6%	0%
Resource allocation	0%	29%	29%	71%	7	0%	11%	12%	15%
Roles	10%	35%	29%	81%	31	75%	58%	53%	76%
Search	0%	50%	50%	75%	4	0%	11%	12%	9%
Semi-structured process (predefined tasks)	0%	47%	35%	65%	17	0%	42%	35%	33%
Semi-structured process (stages)	33%	0%	33%	67%	3	25%	0%	6%	6%
Stakeholders of a case	14%	50%	57%	43%	14	50%	37%	47%	18%
Structured process (aspect)	0%	55%	35%	70%	20	0%	58%	41%	42%
Structured process (complete)	0%	22%	11%	100%	9	0%	11%	6%	27%
Suggestions of activities	20%	60%	20%	20%	5	25%	16%	6%	3%
System of record	5%	39%	32%	74%	38	50%	79%	71%	85%
Task state	33%	67%	67%	33%	3	25%	11%	12%	3%
Task templates	10%	60%	40%	50%	10	25%	32%	24%	15%
Tasks	5%	49%	35%	65%	37	50%	95%	76%	73%
Typed interactions	40%	20%	60%	60%	5	50%	5%	18%	9%
User access control	11%	37%	32%	74%	19	50%	37%	35%	42%
Views for multiple stakeholders / roles	14%	43%	21%	71%	14	50%	32%	18%	30%
Σ						4	19	17	33

6 Discussion

Since the classifications we applied for knowledge work and system types lack clear differentiators, they obviously are subjective to a certain degree. Moreover, the features extracted depend on the authors of that text to actually write about a particular feature or to provide a comprehensive screenshot. Hence, some features of systems that are not described or displayed will be missing. Nonetheless, our analysis shows significant differences between features present in types of knowledge work and types of system. Moreover, knowledge workers of different type focus on different types of systems, i.e. ACMSs seem to be tailored

to or required by the collaboration model. These differences are sufficiently clear to not just stem from erroneous or subjective detection.

The features detected in our analysis can stem from users asking for certain features and the provider assuming requirements as well. Hence, they can only pose as an approximation to the requirements of certain types of knowledge worker. However, the results at least suggest, that a requirements analysis for new or existing ACMS and PCMS cannot be limited to a literature review on typical requirements of knowledge workers, but has to consider peculiarities of the target users at least on the granularity of Davenport's classification.

The feature *typed interactions* covers suggested responses like agree, decline, counter-offer or comment in negotiations [2], as well as approve and deny of document templates for change management [21]. Even though the authors did not comment on speech act theory [22, 23] and probably are not aware of supporting their processes by emphasizing on the pragmatic intention of the user's interactions, they are providing such a support. Finally, the feature *system of record* was not extracted from all case studies. Nonetheless, if the investments are made to support knowledge workers with a CM system, one main motivation might have been to provide such a system. Hence, providers of these systems could put more emphasis on creating a system of record in order to make this goal visible.

7 Conclusion

In this analysis, we evaluated 48 winners of the WfMC Awards for Excellence in Case Management in regard to their classes of targeted knowledge workers, features, and type of system. We confirmed that different types of knowledge workers use different system classes and that the type of system also indicates the type of knowledge work supported. Different types of knowledge work showed a different emphasis in the provided features. Knowledge workers in the collaboration model seem to be underrepresented, i.e. either in the awards or in the target users of CM systems. A large share of award-winning case studies was classified as PCMS. ACMSs focus on semi-structured processes and ad-hoc activities. Unsurprisingly, ACM systems have a higher emphasis on collaboration than PCMSs and BPMSs. Support for document management seems to be important for CM regardless of system type or type of knowledge work.

Of course, this analysis only covers features of award-winning case studies, not requirements of the systems and knowledge workers. Nonetheless, these features were elaborated by the authors of the case studies and most likely deemed important by them. They can hint at what sort of features is most likely asked for by certain users. In order to gain the actual requirements of different types of knowledge workers, interviews and further analyses are still necessary.

References

1. Franks, P.C.: Integrated ECM solutions: where records managers, knowledge workers converge. *Inf. Manag.* **50**(4), 18–22, 47 (2016)
2. 4Spire: Fleet one, USA. In: Fischer, L. (ed.) *Empowering Knowledge Workers: New Ways to Leverage Case Management*, pp. 145–150. Future Strategies Inc. (2014)
3. Tenschert, J., Michelson, G., Lenz, R.: Towards speech-act-based compliance. In: 2016 IEEE 18th Conference on Business Informatics (2016)
4. Davenport, T.H.: How knowledge workers differ, and the difference it makes. In: *Thinking for a Living: How to Get Better Performances and Results from Knowledge Workers*, pp. 25–38. Harvard Business Press (2005)
5. Swenson, K.D.: Innovative organizations act like systems, not machines. In: Fischer, L. (ed.) *Empowering Knowledge Workers: New Ways to Leverage Case Management*, pp. 31–42. Future Strategies Inc. (2014)
6. Palmer, N.G., Dugan, L.: Understanding and evaluating case management software. In: Fischer, L. (ed.) *Thriving On Adaptability: Best Practices for Knowledge Workers*, pp. 31–40. Future Strategies Inc. (2015)
7. Matthias, J.T.: Technology for case management. In: *Mastering the Unpredictable: How Adaptive Case Management will Revolutionize the Way that Knowledge Workers Get Things Done*, pp. 63–88. Meghan-Kiffer Press (2010)
8. Motahari-Nezhad, H.R., Swenson, K.D.: Adaptive case management: overview and research challenges. In: 2013 IEEE 15th Conference on Business Informatics (CBI), pp. 264–269. IEEE (2013)
9. Hauder, M., Münch, D., Michel, F., Utz, A., Matthes, F.: Examining adaptive case management to support processes for enterprise architecture management. In: *IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations*, pp. 23–32, September 2014
10. Fischer, L. (ed.): *Taming the Unpredictable: Real-World Adaptive Case Management*. Future Strategies Inc. (2011)
11. Fischer, L. (ed.): *How Knowledge Workers Get Things Done: Real-World Adaptive Case Management*. Future Strategies Inc. (2012)
12. Fischer, L. (ed.): *Empowering Knowledge Workers: New Ways to Leverage Case Management*. Future Strategies Inc. (2014)
13. Fischer, L. (ed.): *Thriving On Adaptability: Best Practices for Knowledge Workers*. Future Strategies Inc. (2015)
14. Fischer, L. (ed.): *Best Practices for Knowledge Workers: Innovation in Adaptive Case Management*. Future Strategies Inc. (2016)
15. Fischer, L. (ed.): *Intelligent Adaptability*. Future Strategies Inc. (2017)
16. ISIS Papyrus: “Paneon GmbH, Austria”. In: Fischer, L. (ed.) *How Knowledge Workers Get Things Done: Real-World Adaptive Case Management*, pp. 155–164. Future Strategies Inc. (2012)
17. Castillo, L.: Cognocare, an ACM-based system for oncology. In: Fischer, L. (ed.) *How Knowledge Workers Get Things Done: Real-World Adaptive Case Management*, pp. 119–128. Future Strategies Inc. (2012)
18. Castillo, L.: Cognocare, an ACM-based system for oncology. In: Fischer, L. (ed.) *Thriving On Adaptability: Best Practices for Knowledge Workers*, pp. 119–126. Future Strategies Inc. (2015)
19. Callewaert, F.: The antwerp port authority. In: Fischer, L. (ed.) *Thriving On Adaptability: Best Practices for Knowledge Workers*, pp. 213–230. Future Strategies Inc. (2015)

20. Hyland “Grinnell mutual, USA”. In: Fischer, L. (ed.) *Intelligent Adaptability*, pp. 99–106. Future Strategies Inc. (2017)
21. ISIS Papyrus Europe AG “Leading european bank - banking correspondence management system”. In: Fischer, L. (ed.) *Intelligent Adaptability*, pp. 107–121. Future Strategies Inc. (2017)
22. Austin, J.L.: *How to do Things with Words*. Oxford University Press, Oxford (1975)
23. Searle, J.R.: *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, New York (1969)