A Survey on Volunteer Management Systems

J. Schönböck, M. Raab and J. Altmann
Upper Austrian University of Applied Sciences, Hagenberg, Austria
Email: {firstname.lastname}@fh-hagenberg.at

E. Kapsammer, A. Kusel, B. Pröll, W. Retschitzegger, and W. Schwinger
Johannes Kepler University Linz, Austria
Email: {firstname.lastname}@jku.at

***Abstract*—Voluntary work is indispensable in nearly every area of today's society, e.g., service activities in health care or emergencies. Not least because of this omnipresence of volunteering, already a plethora of volunteer management systems (VMS) has emerged, trying to support diverse volunteering processes and to deal with the broad spectrum and peculiarities of voluntary work. Thus, an in-depth understanding of functional commonalities and differences of VMS is urgently needed. The goal of this paper is therefore to provide an in-depth survey of existing VMS. For this, first, an initial attempt towards a reference model (RM) for VMS is presented, capturing their basic functional ingredients and interrelationships in terms of UML class diagrams. Second, the RM is operationalized by means of a set of evaluation criteria used to compare seven carefully selected VMS, thereby discussing their peculiarities and shortcomings. Third, lessons learned are provided together with research directions for future VMS.**

I. INTRODUCTION

Importance of voluntary work. Voluntary work adds an important contribution to society comprising primarily *service and charitable activities* in areas like health care or emergency management but also *expressive activities* in areas like sports, art and culture. According to an international study [36], volunteers account for up to 10% of the workforce world-wide, and in the US the volunteer rate even surmounted 25%, meaning that 62.8 million citizens volunteered through or for a *volunteer involving organization (VIO)* [40]. The spectrum ranges from just a few minutes with no ongoing *commitment* (e.g., "micro-volunteering" [3] or "crowdworking" if the tasks are fulfilled online [27]), to expert roles that require permanent commitment and certain skills [10]. No matter if voluntary work is conducted on- or offline, onsite or remote, occasionally (e.g., in case of disaster relief) or long-term (e.g., a community service), it is often done in *teams* to *collaboratively* or *corporately* accomplish tasks. Thereby, team-size range from small, informal groups at a local level, to large, globally acting organizations such as the Red Cross [39].

Volunteer Management Systems. According to the UN Worlds Volunteerism Report [39], information and communications technology (ICT) is a major enabler of volunteering processes. In general, *volunteer management systems (VMS)* bring together potential volunteers with volunteering opportunities, allowing the scheduling, allocation and execution of tasks, providing communication and coordination mechanisms for collaboration and cooperation and finally, facilitating assessment and motivation strategies. There already exist a vast number of VMS [11], most of them commercial, each pursuing different goals and exhibiting different functionality to support the phases of a volunteering process. Thus, VMS also tackle interdisciplinary research areas of computer-supported cooperative work (*CSCW*), e.g., *human resource management* (specifically *user modeling* [22], [26] and *eRecruitment* [5], [17]), *workflows* [23], *crowdworking* [27], and *motivation theory* [8].

Contribution and Structure of this paper. Considering the diversity of existing VMS and the tackled research areas, a more in-depth understanding of functional commonalities and differences of this kind of systems is urgently needed. Although first comparisons exist, they provide a high-level overview of different VMS along with emphasizing VMS usage [11], [24]. The goal of this paper is to provide an in-depth survey of functional features in existing VMS, comprising three main contributions. Firstly, based on challenges identified for VMS and on requirements gained from VIOs in our research project (e.g., the Austrian Red Cross and the Waldorf school), (cf. Sec. II), a *reference model (RM)* is presented (cf. Sec. III), capturing functional components of a VMS in UML class diagrams. Second, the RM is operationalized to *compare seven VMS* and to demonstrate the applicability of our RM, thereby discussing their peculiarities and shortcomings by means of *lessons learned* together with *future research directions* (cf. Sec. IV-V).

II. VOLUNTEER MANAGEMENT SYSTEMS' CHALLENGES

Considering the volunteering process, several challenges on appropriate system support arise.

Inhomogeneous Conglomerate of Resources and Tasks. Volunteering entails dealing with a highly inhomogeneous conglomerate of *resources* and *tasks* as stated by our project partners. Concerning *human resources* (HR), the challenge is to consider diverse competencies, interests as well as personalities of volunteers, being the prerequisite for appropriate task allocation, which may lead to increased intrinsic motivation at the very end [8], [27]. Furthermore, since VIOs often do not exhibit any pre-defined hierarchical structures, a VMS should support *dynamic team building*

[41] for conducting more complex and possibly ad-hoc tasks at hand. Besides HR, also *non-human resources* (NHR) which are needed to achieve a certain task (e.g., a specific equipment), either provided by the VIO or by the volunteers themselves need to be considered. Additionally, volunteering work needs to be broken down into *tasks* of an appropriate granularity and described in terms of the required resources and the expected outcome. Thus, tasks may require some *sophisticated workflow execution* mechanisms at runtime with *structural relations* in-between, e.g., composition of tasks, and *behavioral relations*, e.g., task control flow. Another challenge is that voluntary work can of course not always be fully pre-planned, requiring concepts to adapt workflow definitions at runtime or even to introduce ad-hoc tasks and workflows dynamically, not only in a central way by the VIO but also decentralized by the volunteer [28], to be able to cope with a much more heterogeneous group of users than in traditional workflows [27].

Flexible Allocation Allowing Brokerage and Negotiation. In this peculiar kind of setting where work is *unpaid*, keeping volunteers motivated plays the key role, not only to be able to inspire volunteers for certain tasks, but even to achieve a sustainable commitment of volunteers over time. Thus, volunteers should be allocated to tasks that match their personal profile (e.g., interests or competencies). Thereby, the challenge is that the required profile of a task may not necessarily be equal to the heterogeneous profiles of volunteers. Simple mechanisms, e.g., in terms of role-based assignments, may not be sufficient [25]. Instead a VMS needs to enable a kind of *brokerage* between tasks and volunteers. In order to yield the best match between the two and thus, to maintain and even increase the volunteers' motivation, different *matching strategies* may be beneficial. This also includes that volunteers themselves may propose tasks to other volunteers in order to achieve the overall goal of a *well-balanced effort distribution*. Because of the very nature of volunteer work, task allocation may not necessarily follow a simple *push strategy* only, i.e., a task is assigned by the VIO [14] but may follow a *pull principle*, i.e., the volunteers may select tasks from available ones, instead. However, the final decision of undertaking certain tasks resides by the volunteers themselves and may even require some kind of *negotiation*, e.g., concerning the actual contribution of the volunteer to the task.

Adaptation- and Motivation-Oriented Assessment. As the required outcome of a voluntary task is not always clear in advance and since a VIO has to deal with inhomogeneous resources and tasks, appropriate assessment mechanisms along the whole volunteering process are crucial in order to identify potential *improvements* in terms of *adaptations* as well as suitable *motivation* strategies. In this respect, a VMS should allow assessment by the different stakeholders involved in voluntary work, most important the volunteers themselves, the VIO and the beneficiaries of volunteer work. Furthermore, different aspects of the volunteering process should be the target of assessment [1], comprising volunteers (e.g., their engagement), tasks (e.g., their complexity or outcome), the appropriateness of task allocation and long-term social aspects like satisfaction or gain of expertise.

Continuous Evolution. For the same reasons as assessment is indispensable for voluntary work, a VMS is required to consider evolution in two different dimensions. First, a continuous improvement of the voluntary process is necessary, thus resembling a "learning by doing" approach by appropriate adaption of various aspects of the voluntary process. Besides adapting resources (e.g., by incorporating achievements of a volunteer), also the task perspective is concerned (e.g., extending, merging or decomposing a task into subtasks) and the allocation of tasks or the building of teams (if, e.g., a volunteer is over strained by a task or disappointed in a certain team). Second, and above all, suitable motivation strategies [8] have to be provided, e.g., awards, rankings or some gamification elements [37], [43].

III. A Reference Model for VMS

Based on the challenges discussed in the previous section, a reference model (RM) which outlines the core artifacts of VMS and their interrelationships in terms of UML class diagrams will be presented. Similar to RMs we proposed for other domains (e.g., aspect oriented modeling [42], user profiles [26] or dynamic spatial systems [32]), the design rationale behind was to build a RM on the one hand in a *top-down manner*, i.e., incorporating related work from the diverse research areas relevant for the RM (e.g., existing (meta-)models for CSCW [12], competencies [20] or workflows [23]) and on the other hand in a *bottom-up manner*, i.e., including concepts from existing VMS, as well as on requirements identified from the VIOs involved in our research project. Thus, our RM serves as a framework, interrelating already existing core building blocks and eventually adapting and extending them to the domain at hand. Fig. 1 provides an overview of the key packages which are used to modularize our RM. These packages are described in the following distinguishing among the (structural) *components* of a VMS, its *process* and the vital orthogonal aspect of a VMS' *evolution*.

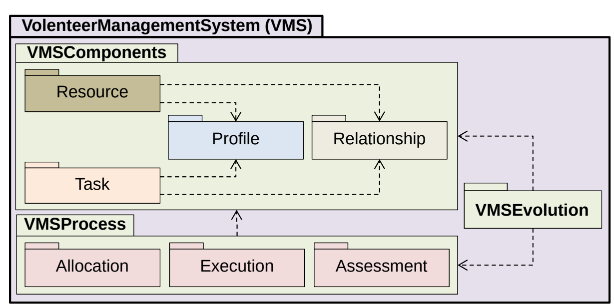

Figure 1. Reference Model for VMS

A. VMS Components

The package VMSComponents contains sub-packages of according concepts, capturing Resources as well as Tasks, being the central artifacts within a VMS, along with dedicated packages modularizing Profiles and Relationship aspects as utilized by the former two. Following, these sub-packages are discussed consecutively, except Relationships, which are described when needed.

Resources. As already extensively discussed in related areas like workflow management [45], a *resource* is a source or supply from which benefit is produced. Thus, the conceptualization of resource should be as domain-independent and generic as possible [45] to include volunteers, but also some equipment or non-tangible resources needed to perform a volunteering task. Consequently, the class Resource is the generalized concept that represents all resources that should be managed by the system (cf. Fig. 2). Thus, at the top level, a resource naturally may be divided into humans (cf. HumanResource) and non-humans (cf. NonHumanResource). HumanResources are further divided into Volunteers, i.e., people who are members of a VIO, and NonVolunteers, i.e., paid employees of the VIO. In contrast, a NonHumanResource may be distinguished into PhysicalResources, e.g., tools or vehicles, and VirtualResources, e.g., guidelines, software and the like. Resources may further be described by Profiles and according Features – imported from the package Profile (cf. Fig. 2) – as well as related to each other in various ways as explicated by the relationship class ResourceRelationship – imported from the package Relationship (cf. Fig. 3). The latter allows to represent *structural relationships*, e.g., teams of volunteers, their membership in organizations and their ability to bring along their own equipment when needed.

Tasks. A large body of literature focuses on the conceptualization of *tasks* [19] and their interdependencies, e.g., in terms of *workflows* [23], [45]. Within our RM, we base on that work, but first we classify tasks along the criteria most relevant for volunteering work [10] (cf. Fig. 2). The criteria *humanity* splits tasks that require manual intervention by humans (HumanTask) from tasks which can be executed automatically (NonHumanTask), whereas the criteria *virtuality* splits online tasks which produce machine readable output (VirtualTask) from tasks which do not (PhysicalTask). *Predictability* divides carefully designed tasks (PlannedTask) from tasks that arose ad-hoc, e.g., during some other tasks (AdHocTask) [28]. The criteria *commonality* divides CollaborativeTasks, i.e., tasks which are executed by several volunteers to achieve a common goal, and CooperativeTasks, i.e., tasks executed together to achieve each volunteer's goal. Finally, *locality* separates location-independent (RemoteTask) tasks from tasks that require the volunteer to be at a certain location to fulfill the task (OnsiteTask).

Since volunteer work often consists of complex, interdependent tasks that need to be coordinated across volunteers with different expertise and capabilities [27], [30], our RM adheres to the proposal of [23], where a meta-meta model for seven business process modeling languages has been suggested. Some of the workflow concepts shown in the forthcoming packages (e.g., Relationships and Profiles) act as *extension points* allowing to plug in further concepts provided by existing business process languages, thereby emphasizing the framework character of our RM. One of these extension points is the class TaskRelationship (cf. package Relationship in Fig. 3). The subclass StructuralTaskRelationship allows the *composition* of tasks, i.e., to represent dependent subtasks, *versions* of a task, and reuse of tasks by *instantiating* a new task based on an existing one. BehavioralTaskRelationship expresses typical behavioral dependencies, e.g., *sequential* or *parallel* task executions. To be able to execute a task, Resources are required as input and might be produced as output. These relationships are annotated by means of the association class AllocationRelationship (cf. Fig. 3), allowing to annotate further semantics, e.g., to express that a certain resource is *required* or *optional*.

Profiles. Not least for an appropriate assignment of volunteers and to foster intrinsic motivation of volunteers (cf. Sec. II), it is central in VMS to allow expressing specific characteristics of volunteers like for any other Resource and describing the characteristics that comprise a certain Task, ideally at a rather fine-grained level. Since both resources and tasks comprise overlapping characteristics, Profiles explicate those descriptions in terms of a set of Features (cf. Fig. 2) which may themselves be composed by other features. The semantics of this composition is further expressed by means of the association class ProfileRelationship, imported from the package Relationship (cf. Fig. 3). Thereby, the composition of a Profile by its Features may be specialized according to the role a feature plays in describing a profile, allowing to capture that e.g., a resource *may* or *may not* exhibit a certain feature (e.g., a volunteer which is *not* interested in doing online tasks).

The subclasses of Feature represent the actual profile information. The rationale behind this categorization stems from a RM we developed for social user profiles [26]. These subclasses constitute again extension points for existing meta-models and taxonomies, as discussed in the following. Besides BasicData (e.g., name, address or technical information on resources), the subclasses depicted for representing interests (cf. Interest in Fig. 2) the *emotional state* and *personality* (cf. Personality in Fig. 2) can be further extended using the General User Modeling Ontology (GUMO) [22], which provides a comprehensive taxonomy for these issues. Incorporating these concepts and

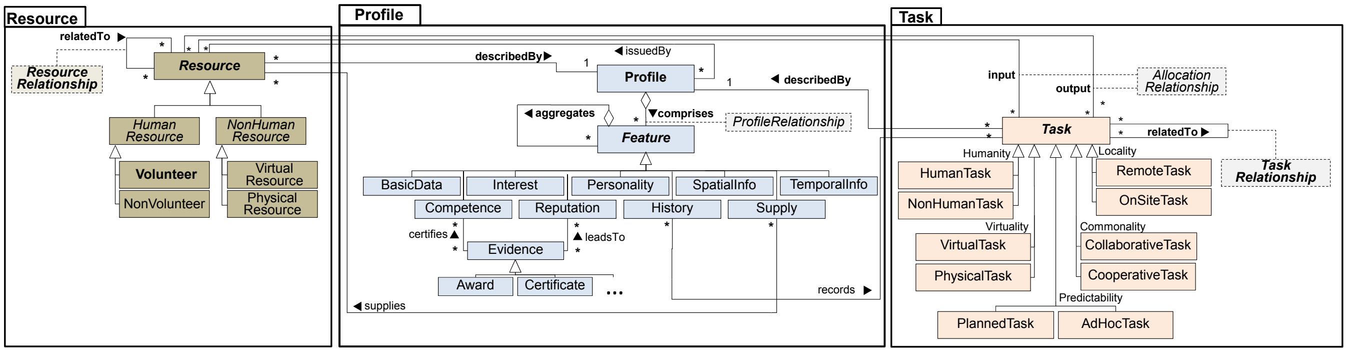

Figure 2. Resource, Profile and Task Package

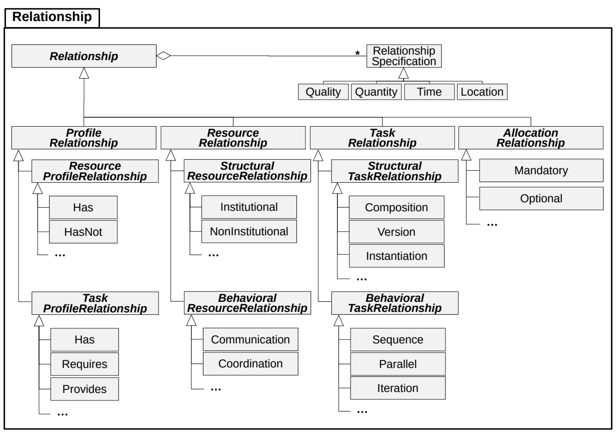

Figure 3. Relationship Package

properties at a fine-grained level is essential in VMS, since teams consisting of volunteers with similar interests and personality offer a better social integration, task performance and satisfaction [7], leading to better motivated teams.

Since appropriately assigning tasks to volunteers is essential, an adequate consideration of available and required *competencies* to successfully perform a task is crucial (cf. `Competence` and [34] for overview on competency models). For further categorizing and characterizing competencies, we adhere to [17] which provide an in-depth investigation of literature on *human resource management* and *eRecruitement* coming up with a sound conceptual ontology for digital resumes, thus further possibly specializing *competencies* into *education*, *skills* and *work experience* and according attributes. Furthermore, in order to provide a generic representation of relationships between different competencies, together with according *competency levels* (e.g., being either binary or rankable), we base on the so-called *InLoc (Integrating Learning Outcome)* specification [20], developed in the course of EU ICT standardization work for representing learning outcome domain-independently. In order to avoid the "cold start problem", meaning that all possible competencies have to be defined when first using a VMS, the DISCO project[1] provides a dictionary with more than 100.000 skill and competency definitions. Finally, competencies, unlike, e.g., interests, may call for an underpinning which is provided by *Evidence*. For example, volunteering in a medical VIO may require a certain certified competence in health care. In turn, certain evidence, e.g., an award being granted for successful volunteer work, may in parallel lead to an increase in reputation (cf. `Reputation`) being relevant for a volunteer's motivation [8].

[1]http://disco-tools.eu

Besides the above mentioned features, resources exhibit *spatial* and *temporal* availabilities and restrictions (cf. `SpatialInfo` and `TemporalInfo` in Fig. 2). For this, one can base on established concepts and ontologies as we proposed, e.g., in [32]. Additionally, for specifying core temporal data, formats like iCalendar are of relevance, allowing for a platform independent representation of calendar entries. To express temporal relationships one can base on the concepts presented in [2]. A profile should furthermore represent that a resource or even a task is supplying a resource (cf. `Supply`), as for example volunteers might contribute own resources during their volunteer work. Finally, as volunteering progresses, it should be possible to keep track of fulfilled tasks (cf. `History`).

B. VMS Process

After having described the core components of a VMS, this subsection focuses on the actual `VMSProcess` (cf. Fig. 4) spanning over `Allocation`, `Execution` and `Assessment` that should be supported by a VMS.

The rationale behind designing the `VMSProcess` package was firstly, to provide for a reification of the different *states* of a VMS process using a generic base class `State`, thereby providing a pivotal artifact for all phases, allowing also to represent the order of actual states during runtime (cf. the recursive association). Secondly, it has been decided to specialize this common base class within the different phases to various subclasses (as common in the area of workflows, cf. e.g., the WS-Human Task standard [21]) in order to reason about the actual state of the process at a fine-grained level. Thirdly, to provide for extensive communication possibilities necessary for volunteering work, the class `Communication` allows not only to announce the actual state of the volunteering process within a certain phase, but also to announce available resources (e.g., volunteers seeking for new volunteering opportunities) and available tasks. Additionally, the actual state can be used in order to determine the "context" for a certain communication, meaning that the addressees for announcing a certain state, being them External or Internal with respect to the VIO, are dynamically computed (e.g., announcing assigned tasks to responsible volunteers) and contacted via different media, e.g., EMail or SMS. Finally, CommunicationRules can be defined in order to configure preferences with respect to certain communication channels (e.g., preferred channel is SMS, but after 10pm use email instead).

Allocation and Execution. Within allocation and execution phases, volunteers should be assigned to tasks which are further on executed, whereby different sub-states are represented, basically based on [21] but extended for capturing the peculiarities of our volunteering domain. Please note that the arrangement of sub-states in Fig. 4 does not necessarily impose a specific execution order.

The state Announced indicates the availability of tasks and resources, again by means of the communication facility described above. Further on, matching the required profiles of tasks with the provided profiles of volunteers is conducted (cf. Matched) by means of different MatchingStrategies being either Manual or Automatic. For the latter, intelligent mechanisms are required that provide some kind of semantic match based on similarity measures to reduce effort in finding appropriate tasks and taking into consideration communication differences of individuals in taking up recommendations given by automatic systems, which has also been studied in different research areas (cf. e.g., [44]). For example, in workflow research, algorithms are used for the assignment of individuals to a set of tasks, cf., e.g., [9] and [38]. In the domain of online recruiting, matching algorithms are used to automatically find the most appropriate job seekers for a certain job announcement (cf., e.g., [16], [18], [29], [33]).

The actual assignment of resources to tasks out of the matches identified is dependent on the employed AssignmentMethod being either Central or Decentral in case volunteers themselves select from the matched tasks. In any case, due to the nature of volunteering, assignments may be Rejected by volunteers or Delegated to others. After a task is assigned to a certain volunteer or a group thereof, execution starts depending on temporal constraints of tasks and volunteers (i.e., the schedule) leading to the state InProgress. Thereupon, in case of some exceptions, i.e., a situation happens which has not been foreseen when planning the workflow, the task can be Suspended or Adapted in some way (e.g., splitted, merged or new ad-hoc tasks may be introduced) when needed [28]. Finally, if the execution of a task is Finished or Canceled, an according TaskResult is produced, that builds the basis for a later task assessment.

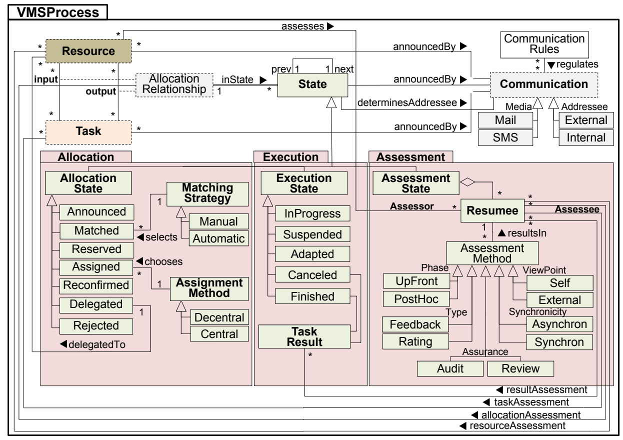

Figure 4. VMSProcess Package

Assessment. As already stated, the inhomogeneous conglomerate of resources and tasks together with non-enforceable execution generate an extensive demand for an assessment of the volunteering work represented by AssessmentState in Fig. 4. The assessment results in a set of Resumes, which is put forward by an assessor (cf. role Assessor of the class Resource), being e.g., a volunteer, a group of volunteers or an assessment algorithm. The Resumee in turn contains some evaluative statement on various different aspects of the volunteering work (cf. role Assessee). These include assessments about *(i) resources*, e.g., the availability of non-human resources or the engagement of a volunteer, *(ii) tasks*, e.g, their complexity or attractiveness, *(iii) allocation*, evaluating, e.g., the appropriateness of resources for a task or the suitability of a task with respect to a volunteer's competencies, and finally about the *(iv) task result*, computing, e.g., the (monetary) impact generated through a task conducted by a volunteer.

Depending on a VIO's needs, different AssessmentMethods may be required, which may be concurrently classified along various categories (cf. work in the area of eLearning and crowdworking, e.g., [1], [15], [27]). Although, assessment may primarily be scheduled after task execution, i.e., PostHoc, it may likewise be seen to be needed UpFront (cf., [1] and [27]). Thereby, upfront approaches focus on task preparation, e.g., to split work into fault-tolerant subtasks [4] or volunteer allocation, e.g., reputation-based approaches, which limit contribution to a task to volunteers with a pre-specified reputation. Post-hoc methods focus on a summarizing assessment, i.e., evaluation is conducted after task finishing. In the research area of quality assessment, traditionally a distinction is made between Reviews, e.g., of the process results and Audits, e.g., of rules and regulations during conducting the process. Depending on the viewpoint an assessment can be distinguished into Self assessment, i.e., volunteers assess themselves, and External assessment, i.e., persons outside of the VIO, e.g., the beneficiaries of volunteer work conduct an assessment [27]. With respect to the point in time of the assessment, feedback may either be given when volunteers are still engaged in a set of tasks, i.e., `synchron` methods like shepherding [15], or when (part of) a task is completed, i.e., `asynchron`. Concerning the outcome of the assessment, different result types may be provided, e.g., verbal `feedback` or quantitative `rating`.

C. Evolution

The `Evolution` package subsumes all activities that are required to continuously improve the volunteering work and therefore heavily bases on the assessments available as `Resumees`. Thereby, we included not only a set of adaptation methods into our `Evolution` package (cf. Fig. 5), responsible for the advancement of the VMS data in the course of diverse volunteering processes, but also a set of motivation methods that intend to increase the intrinsic motivation of a volunteer. The rationale behind was that preventing the de-motivation of volunteers and ideally increasing the motivation has to be seen as a continuous necessity in VIOs, just like adapting the VMS' data.

Concerning *motivation*, a broad spectrum of methods can be applied ranging from well known psychology and sociology approaches [8], to gamification, i.e., the use of game thinking and game mechanics in non-game contexts to engage users in solving problems [43], and needs to take into consideration influencing factors to maintain and increase intrinsic motivation (cf. [13]). In our RM we explicate some exemplary methods, including `Appraisal` like a simple "thank you-letter", `Awarding` like a "I volunteered" badge, `Ranking` like "best volunteer of the week", `IncentiveGranting` like discounts granted by sponsors of the VIO or professional training courses, `Challenging` like raising the level of difficulty of tasks to address the playful nature of humans (including also mechanisms like "job enrichment" or "job enlargement"), and finally, `Recruiting` to broaden the basis of volunteers and thus to further increase the possibilities of social interaction, being another motivating factor in volunteering.

Regarding *adaptation*, mechanisms should be provided to evolve the diverse profile features of resources and tasks (cf. `AdaptResource` and `AdaptTask` in Fig. 5). In this respect it should be for example possible to adapt a competence profile of a volunteer, as new competencies are gained when conducting a task, which entails a "learning by doing" approach. Likewise, the matching strategy and assignment methods employed (cf. `AdaptMatching` and `AdaptAllocation`) may be adjusted such that they better "understand" the human nature of volunteering [27].

IV. Comparison and Lessons Learned

This section applies the RM to compare seven selected VMS, thereby discussing their peculiarities and shortcomings and deriving lessons learned from the evaluation.

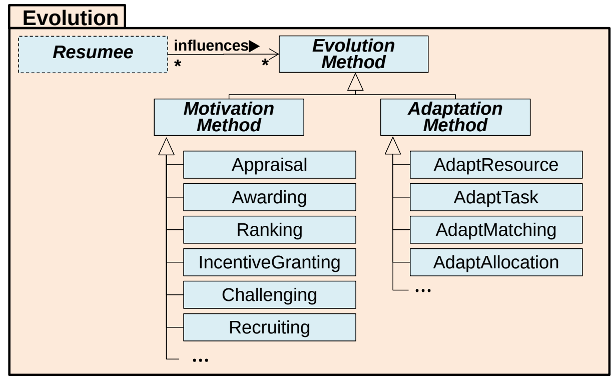

Figure 5. Evolution Package

A. Selected Systems

As already mentioned, interestingly, nearly all existing systems supporting more or less the whole volunteering process are commercial ones [11], [24]. Out of these, we selected 5 VMS which have been empirically evaluated as most used systems and which are web-based and offer at least an according possibility to manage human resources (volunteers and organizational structures). Additionally, two VMS have been included that provide at least partly complementary functionality, resulting in seven selected systems which are briefly characterized in the following.

First of all, the system *YourVolunteers* sticks out by providing so-called "shifts" i.e., tasks that occur regularly and which can be selected by volunteers, which should reduce management effort. *Volunteer Impact* has been selected due to its configurable communication support. The system *VolunteerMatters* allows for interesting mechanisms for the coordination of volunteers and the recruitment of new ones. *Volgistics* allows management of volunteers throughout the entire volunteer cycle, ranging from the online registration process to the calculation of awards, being above all, customizable in numerous respects (e.g., application forms, reports, schedule). *Samaritan* is one of the most comprehensive and customizable VMS, providing a dedicated system component ('eRecruiter') accessible by volunteers for, e.g., registering and editing their personal profiles and another component ('eCoordinator') serving as back-end for the management tasks of volunteer coordinators. Additionally, we included *movements.org* as a two-sided "brokering" platform, similar to eBay that enables the exchange of skills and resources. Users can post short briefs describing their projects, and providers can post offers of assistance describing their skills and ways in which they can help, emphasizing the brokerage character of the system, which is different to the other systems described above. With *GiveGab* we finally integrated an app which can be considered as a "social volunteering network", allowing to find local volunteering opportunities, also by connecting to friends, see where others are volunteering, and share volunteering experiences and impact with others.

Table I
COMPARISON OF VOLUNTEER MANAGEMENT SYSTEMS

		YourVolunteers (yourvolunteers.com)			Volunteer Impact (betterimpact.com)			VolunteerMatters (volunteermatters.com)			Volgistics (volgistics.com)			Samaritan (samaritan.com)			Movements (movements.org)			GiveGab (givegab.com)		
		HR	NHR	Task	HR	NHR	Task	HR	NHR	Task	HR	NHR	Task	HR	NHR	Task	HR	NHR	Task	HR	NHR	Task
Task + Resource	Kind	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓
	Profile Feature																					
	Basic Data	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓
	Competence	×	×	×	✓	×	✓	✓	×	×	✓	×	✓	✓	×	✓	~	×	~	~	×	~
	Evidence	×	×	×	×	×	×	×	×	×	×	×	×	~	×	×	×	×	×	×	×	×
	Interest	✓	×	✓	✓	×	×	✓	×	×	✓	×	✓	✓	×	✓	~	×	~	~	×	~
	Reputation	×	×	×	×	×	×	×	×	×	×	×	×	~	×	×	×	×	×	×	×	×
	Personality	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	SpatialInfo	×	×	×	×	×	×	×	×	×	✓	×	✓	×	×	✓	~	×	~	×	×	✓
	TemporalInfo	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓	×	×	×	×	×	✓
	History	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	✓	✓	×	×	✓	×	×
	Supply	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	Customization	~	×	×	✓	×	×	✓	×	×	✓	×	×	✓	×	✓	×	×	×	×	×	×
	Relationship																					
	Structural	groups (admin)	n/a	instance	groups (admin)	n/a	instance category	groups	n/a	category	groups (categories)	n/a	site, place, cluster	groups, corporation, partners	n/a	instance	×	n/a	predefined category	groups	n/a	composition
	Behavioral	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	Data Management																					
	Central | Decentral	✓ | ✓	n/a	✓ | ×	✓ | ✓	n/a	✓ | ×	✓ | ✓	n/a	✓ | ×	✓ | ✓	n/a	✓ | ~	✓ | ✓	n/a	✓ | ~	× | ✓	n/a	× | ✓	× | ✓	n/a	✓ | ×
	Import	×	n/a	×	manufacturer	n/a	×	✓	n/a	×	✓	n/a	×	✓	n/a	×	×	n/a	×	✓	n/a	×
	Synchronisation	×	n/a	×	×	n/a	×	×	n/a	×	×	n/a	×	×	n/a	×	×	n/a	×	×	n/a	×
Allocation + Execution	**Communication**																					
	Addressor	supervisor			supervisor, volunteer			supervisor			supervisor			supervisor			volunteer			supervisor, volunteer		
	Addressee	volunteers			volunteer(s), supervisor			volunteer(s)			volunteer(s)			volunteer(s)			volunteer			volunteer(s), supervisor		
	Media	email			email, text message			email, letter			email, text message			(scheduled) email			email			email		
	Distribution	1 : n			1 : 1, 1 : n, n : m (within group)			1 : 1, 1 : n			1 : 1, 1 : n			1 : 1, 1 : n			1 : 1, 1 : n (comments)			1 : 1, 1 : n, n : m (social network)		
	Communication Rules	×			~			×			✓			×			~			×		
	Task Announcement																					
	Internal	✓			✓			✓			✓			✓			✓			✓		
	External	~			✓			✓			×			✓			×			✓		
	Cross-Instance	×			×			×			×			✓			×			~		
	Allocation																					
	Matching Strategy	semi-automatic			semi-automatic			semi-automatic			semi-automatic			automatic			×			×		
	Assignment Method	central, decentral			central, decentral			central, decentral			central, decentral			central, decentral			central, decentral			central, decentral		
	Delegation (volunteer)	×			×			×			×			×			~			×		
	Rejection	✓			✓			✓			✓			✓			×			✓		
Assess-ment	Assessment Method	×			×			×			×			~ feedback			×			~ feedback		
	Assessee													×	×	✓				×	×	✓
Evolution	Motivation	~ appraisal			~ appraisal			~ appraisal			✓ appraisal, awarding			✓ appraisal, awarding			~ comments			✓ appraisal, awarding, challenging		
	Adaptation	×			×			×			×			×			×			×		
Platform	Operating system	web-based			web-based			web-based			web-based			web-based			web-based			web-based		
	Mobile access	✓			✓			~			~			✓			✓			✓		

B. *Evaluation and Lessons Learned*

After having presented the selected systems, this subsection provides the evaluation of functional features depicted in Table I and discusses lessons learned derived thereof.

Limited Profile Coverage. Concerning profiles the core functionality of managing human resources and tasks is achieved by all approaches but none of them provides support to comprehensibly manage non-human resources (cf. Table I). Regarding volunteers and tasks, the approaches rely on basic data, competencies, interests, history data, and temporal information, but do hardly consider data on personality, reputation, resources that one could supply, or spatial information that shares the location of task fulfillment or the volunteers possible operational range. However, most systems provide means to extend the existing features that may be stored in the profile, e.g., by allowing the specification of custom fields (cf. Table I). *VolunteerMatters* even allows usage of these fields during matching. Nevertheless, the evaluated approaches do not base on standards or publicly available ontologies, hindering interoperability and extensibility. Thus, current approaches should put special emphasis on integrating existing ontologies to provide a more fine-grained representation of volunteering data.

Walled Volunteering Data Islands. Concerning data administration, most approaches allow volunteers and administrators alike to manage data, i.e., central and decentral data management for user profiles is provided. However, the evaluated VMS fall short in interacting and incorporating external data and services (cf. *Data Management* in Table I). They hardly support import of external data to incorporate data available in social networks, like Facebook or LinkedIn, to fill profiles and thus to relax the cold-start problem of properly describing resources. Thus, a huge source of valuable data remains unexploited. Likewise, synchronization with external data sources and social networks is neglected. For example, synchronisation of contacts, tasks and appointments with a local database, e.g. through a smartphone, is not supported which could open up interesting motivational aspects since the user does not have to interact with the system directly. To overcome these limitations, VMS should provide APIs, based on open standards, e.g., REST, to seamlessly integrate data from external data sources.

Limited Support for Complex Tasks and Social Relationships. Available systems are able to deal with simple, often online, tasks known as human intelligence tasks (HITs) [27], which may be executed by a single person but are deficient in dealing with more complex tasks that require the consideration of special competencies, resources and dependencies between resources or tasks. That is, they provide less means to manage relationships between resources and task (cf. *Relationship* in Table I). Furthermore, there is a deficiency with respect to managing structural relationships (e.g., composing or decomposing of tasks) as well as behavioral relationships (e.g., dependencies). Concretely, almost all approaches allow in some way a grouping of tasks or specifying task instantiations from templates, but none of the evaluated approaches provide means to specify behavioural relationships, which would be the prerequisite for comprehensive workflow support. Finally, with respect to volunteers, existing systems only support simple groups, but there is insufficient support, e.g., to deal with social relationships like knowledge of other volunteers or friendship, which could be exploited during the allocation phase. Therefore, VMS should take advantage from the existing body of literature available for years in the area of workflow management [23] and for representing more fine-grained social relationships [26] as also foreseen in our RM.

Constrained Communication. Communication support is crucial in VMS, not only with respect to motivation but also for the actual support of the volunteering process. However, communication support is most often constrained to one-to-many communication from a supervisor (i.e., a person that created the task) to the volunteers. Only *GiveGab* and *Movements* allow communication among all volunteers, as it is implemented as a social volunteering network or brokerage system. Both allow communication through public comments as well as private messages to individuals. *VolunteerImpact* gives the administrator the right to set up special groups, called committees, within communication may or may not be allowed. All other VMS offer mostly great support in one-to-many communication but do not support individual communication among volunteers, e.g. for managing collaborative tasks. In relation to communication rules, only three approaches out of seven allow individually configuring communication. However, whereas in *VolunteerImpact* it is only possible to enable or disable communication within a specified group, and in *Movements* the volunteer can only specify the regularity of notifications, *Volgistics* allows a more flexible approach, e.g., the volunteer can opt-in or opt-out text message reminders, set up a time frame for incoming messages and even choose the category of messages, e.g. team leader update or recruitment appeals, that should be forwarded. Concerning the announcement of a task, all systems allow announcements within the system, but not all support the announcement of tasks across system borders. Thereby, most systems support publishing a tasks to e.g., social media, but only *Samaritan* supports sharing of tasks among different instances of the VMS. With respect to communication, in general VMS provide quite comprehensive support. However, specific tools often miss a valuable feature present in some other. Thus, a unified communication layer integrating all the existing communication features of the evaluated tools would be beneficial.

Suboptimal Effectiveness and Efficiency of Allocation. Despite being crucial in VIOs that volunteers accomplish tasks that are in their own best interests to stay motivated, current system support falls short in terms of effectiveness and efficiency. The effectiveness of resource and task allocation is hampered by incomplete profile descriptions, as already mentioned above, that would be needed for matching the most appropriate task. Additionally, current systems mainly support some kind of semi-automatic task assignment allowing a supervisor to filter tasks based on the resources' attributes, e.g., the volunteer's competencies or temporal availability, but demand knowledge of the tasks' requirements by the supervisor. To improve current VMS for use in large-scale VIOs, assistance in terms of automatic assignment that exceeds the current state in that it takes into account possible comprehensive descriptions of resources and tasks with respect to competencies but also interests, personality and reputation would be required. For this, intelligent matching strategies could benefit from semantic descriptions thereby addressing gaps in partly matching resource and task specifications, as e.g., discussed in [9].

Lack of Assessment Integration. Although VIO state that assessment is extremely important when working with volunteers, this is currently hardly considered at all by existing systems. Only Samaritan and GiveGab provide basic support by allowing a supervisor conducting surveys on the actual task they have fulfilled. In this respect assessment is seen as a step after task fulfillment only. None of the systems stretches over the full spectrum of assessment options as sketched in our RM. A proper assessment phase, however, can be seen as a feedback to the volunteer, to the supervisor and last but not least to the system. Thus, assessment results can not be immediately utilized for further improvements within the system to allow benefiting from the assessment during VMS evolution in terms of motivation and adaptation. In this respect a VMS should try to integrate according means for assessment, thereby basing on methods from crowdworking as e.g., presented in [6].

Sparse Support for Motivation. In [35] it has been discussed that volunteers demand for more entertaining, meaningful, and/or trendy issues. Therefore, it is of utmost importance that a VMS fosters motivation. However, as can be seen in Table I only sparse support is provided by current systems. Most of the systems provide means for appraisal by presenting statistics to volunteers allowing to check how many tasks have been completed within a certain period of time or how many hours have been spent for voluntary work. Only the VMS *Volgistics*, *Samaritan* and *GiveGab* provide additional means by awards, e.g., *Volgistics* based on hours or years of service. Thereby, the VMS keeps track and automatically suggests volunteers who are eligible for an award. *GiveGab* gives the volunteer the chance to set personal goals, join challenges, earn badges and stickers, compare volunteer hours and show the volunteer the impact of his activity. To improve means for motivation, gamification approaches, i.e. as seen in *GiveGab*, should be included within a VMS, whereby different game mechanisms are capable of motivating different types of "human desires" as discussed in [37]. For instance, points may be used to motivate audiences with a strong desire for rewards. The fact that it costs at least five times more to recruit a new volunteer than to cultivate greater relationship with existing ones [31], intensifies the requirement for gamification approaches.

Missing Support for Evolution. None of the evaluated tools provide support for evolution of profiles, since no according assessment mechanisms are provided (except Samaritan and GiveGab) which could be used as a basis. Thus, existing VMS miss the opportunity of turning static profiles into dynamically adapted profiles. Consequently, results stemming from task execution or from an assessment process should be used to accordingly adapt profiles, tasks, and allocation mechanisms to bear up motivation.

V. Future Work

The focus of the paper was on covering functional requirements of existing VMS. Although parts of the requirements stem from an analysis with VIOs which are partners (Austrian Red Cross and the Waldorf school) in our research project, a broader investigation would be needed. In this respect, an extensive end user study is considered as future work, which may be used to identify shortcomings of existing VMS and of the RM from a user perspective, considering also non-functional requirements like user experience and providing a deeper insight on motivational aspects in existing systems. Since motivational aspects are hardly provided by current systems, an additional point for future work would be building components for motivation that might be included into existing VMS and to put emphasis on how motivation-oriented system components actually affect motivation.

Acknowledgments

This work has been funded by the FFG COIN 845947. The authors would like to thank the following for their valuable contributions: Gerhard Funk, University of Art and Design Linz; Cäcilia Innreiter-Moser, Johannes Kepler University Linz; Kurt Feßl, Nikolaus Dürk, Roland Lehner and Katharina Kloiber, X-Net Services GmbH; Philippe Brandner and Christoph Haselberger, blp GeoServices GmbH; Andreas Riepl, GTN solutions; Christian Lill-Rastern, ideenweberei OG; Johannes Leitner, Waldorf school; Sandra Wolkerstorfer, Red Cross.

References

[1] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, E. Bertino, and S. Dustdar, "Quality Control in Crowdsourcing Systems: Issues & Directions," *IEEE Internet Comp.*, vol. 17, no. 2, pp. 76–81, 2013.

[2] J. F. Allen, "Maintaining Knowledge About Temporal Intervals," *Commun. ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[3] M. Bernstein, E. Cutrell, S. Dow, E. Gerber, A. Jain, and A. Kulkarni, "Micro-volunteering: helping the helpers in development," in *Proc of 13th CSCW*. ACM, 2013, pp. 85–88.

[4] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich, "Soylent: a word processor with a crowd inside," in *Proc. of the 23nd ACM symposium on UI software and technology*, 2010, pp. 313–322.

[5] C. Buttinger, B. Pröll, J. Palkoska, W. Retschitzegger, M. Schauer, and R. M. Immler, "JobOlize - Headhunting by Information Extraction in the era of Web 2.0," in *Proc. of the 7th IWWOST*, 2008.

[6] C. Callison-Burch, "Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazon's Mechanical Turk," in *Proc of EMNLP*, 2009, pp. 286–295.

[7] W.-C. Chen, Y.-M. Cheng, F. E. Sandnes, and C.-L. Lee, "Finding Suitable Candidates: The Design of a Mobile Volunteering Matching System," in *Human-Computer Interaction. Towards Mobile and Intelligent Interaction Environments*, 2011, pp. 21–29.

[8] E. G. Clary and M. Snyder, "The Motivations to Volunteer: Theoretical and Practical Considerations," *Current Directions in Psychological Science*, vol. 8, no. 5, pp. 156–159, 1999.

[9] S. Colucci, T. D. Noia, E. D. Sciascio, F. M. Donini, and A. Ragone, "Semantic-based Skill Management for Automated Task Assignment and Courseware Composition," *j-jucs*, vol. 13, no. 9, pp. 1184–1212, 2007.

[10] J. Cravens, "Internet-mediated Volunteering in the EU: Its history, prevalence, and approaches and how it relates to employability and social inclusion," *JRC-IPTS*, 2014.

[11] J. Cravens and R. Jackson, "Survey of software tools used to track and manage volunteer information," Tech. Rep., 2012. [Online]. Available: http://www.coyotecommunications.com/tech/volmanage.html

[12] C. R. G. de Farias, L. F. Pires, and M. van Sinderen, "A conceptual model for the development of cscw systems," in *Proc. of 5th Int. Conf. on the Design of Cooperative Systems COOP*, 2000, pp. 189–204.

[13] T. de Vreede, C. D. Nguyen, G. de Vreede, I. Boughzala, O. Oh, and R. Reiter-Palmon, "A Theoretical Model of User Engagement in Crowdsourcing," in *Proc of CRIWG*, 2013.

[14] C. Dorn, R. N. Taylor, and S. Dustdar, "Flexible Social Workflows: Collaborations as Human Architecture," *IEEE Internet Computing*, no. 2, pp. 72–77, 2012.

[15] S. Dow, A. Kulkarni, S. Klemmer, and B. Hartmann, "Shepherding the crowd yields better work," in *Proc. of CSCW*, 2012, pp. 1013–1022.

[16] M. Fazel-Zarandi and M. S. Fox, "Semantic Matchmaking for Job Recruitment: An Ontology-based hybrid Approach," in *Proc. of 8th Int. Semantic Web Conf.*, 2009.

[17] E. Furtmueller, "Using technology for global recruitment: why HR/OB scholars need US knowledge?" Ph.D. dissertation, Univ. Twente, 2012.

[18] I. Ghani, F. M. Hassan, M. Faheem, M. Faheem, and A. Ali Hajji, "Ontology matching approaches for e-recruitment," *Int. Journal of Computer Applications*, vol. 51, no. 2, pp. 39–45, 2012.

[19] S. Goschnick, L. Sonenberg, and S. Balbo, "A Composite Task Meta-model as a Reference Model," in *Human-Computer Interaction*, ser. IFIP Advances in ICT, 2010, vol. 332, pp. 26–38.

[20] S. Grant and C. Sgouropoulou, "What is a level of competence?" in *Proc. of 5th COME-HR*, 2011.

[21] O. S. Group, "Web Services - Human Task (WS-HumanTask) Version 1.1," 2010. [Online]. Available: http://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-cs-01.html

[22] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilamowitz-Moellendorff, "Gumo - The General User Model Ontology," in *User Modeling 2005*, 2005, pp. 428–432.

[23] F. Heidari, P. Loucopoulos, F. Brazier, and J. Barjis, "A Meta-Meta-Model for Seven Business Process Modeling Languages," in *Proc. of 15th CBI*, 2013, pp. 216–221.

[24] idealware, "A Consumers Guide to Software for Volunteer Management," 2011. [Online]. Available: http://www.idealware.org/volunteer_management

[25] G. Kappel, S. Rausch-Schott, and W. Retschitzegger, "Coordination in Workflow Management Systems - A rule-based approach," in *Coordination Technology for Collaborative Applications*, ser. Springer LNCS, 1998, vol. 1364, pp. 99–119.

[26] E. Kapsammer, S. Mitsch, B. Pröll, W. Retschitzegger, W. Schwinger, M. Wimmer, M. Wischenbart, and S. Lechner, "Towards a Reference Model for Social User Profiles: Concept & Implementation," in *Proc. of 5th Int. WS on Personalized Access, Profile Management, & Context Awareness in DB*, 2011.

[27] A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton, "The future of crowd work," in *Proc of 16th CSCW*. ACM, 2013, pp. 1301–1318.

[28] G. Kramler and W. Retschitzegger, "Towards intelligent support of workflows," *AMCIS 2000 Proceedings*, p. 409, 2000.

[29] H. Lv and B. Zhu, "Skill ontology-based semantic model and its matching algorithm," in *Proc of 7th Int. Conf on CAIDCD*. IEEE, 2006, pp. 1–4.

[30] T. W. Malone and K. Crowston, "The Interdisciplinary Study of Coordination," *ACM Comput. Surv.*, vol. 26, no. 1, pp. 87–119, 1994.

[31] M. A. Mitchell and S. Taylor, "Internal marketing: Key to successful volunteer programs," *Nonprofit World*, vol. 22, no. 1, pp. 25–26, 2004.

[32] S. Mitsch, A. Platzer, W. Retschitzegger, and W. Schwinger, "Logic-based Modeling Approaches for Qualitative and Hybrid Reasoning in Dynamic Spatial Systems," *ACM Comput. Surv.*, 2015.

[33] M. Mochol, R. Oldakowski, and R. Heese, "Ontology based recruitment process." in *GI Jahrestagung (2)*, 2004, pp. 198–202.

[34] K. Rezgui, H. Mhiri, and K. Ghedira, "Competency Models: A Review of Initiatives," in *Proc. of 12th ICALT*, 2012, pp. 141–142.

[35] D. Safrit and M. Merrill, "Management implications of contemporary trends in voluntarism in the united states and canada," Institute for Volunteering Research, Tech. Rep., 2000.

[36] L. M. Salamon, S. Sokowski, M. Haddock, and H. Tice, "The state of global civil society volunteering," Johns Hopkins University, Center for Civil Society Studies, 2013. [Online]. Available: http://ccss.jhu.edu/state-of-global-civil-society-volunteering/

[37] A. Shahri, M. Hosseini, R. Ali, and F. Dalpiaz, "Gamification for Volunteer Cloud Computing," in *Proc of 7th UCC*, 2014, pp. 616–617.

[38] M. Shen, G.-H. Tzeng, and D.-R. Liu, "Multi-criteria task assignment in wfms," in *Proc. of 36th HICSS*, 2003, pp. 9–19.

[39] UN Volunteers, "State of the World's Volunteerism Report," 2015. [Online]. Available: http://www.volunteeractioncounts.org

[40] US Department of Labor, "VOLUNTEERING IN THE UNITED STATES - 2014," 2014. [Online]. Available: http://www.bls.gov/news.release/pdf/volun.pdf

[41] X. Wang, H. Sun, M. Xu, and X. Zhou, "Dynamics in Hierarchical CSCW Systems," in *Advanced Parallel Processing Technologies*, ser. Springer LNCS, 2003, pp. 546–556.

[42] M. Wimmer, A. Schauerhuber, G. Kappel, W. Retschitzegger, W. Schwinger, and E. Kapsammer, "A survey on UML-based aspect-oriented design modeling," *ACM Computing Surveys*, vol. 43, no. 4, pp. 1–33, 2011.

[43] G. Zichermann and C. Cunningham, *Gamification by design: Implementing game mechanics in web & mobile apps*. O'Reilly Media, 2011.

[44] B. M. Zongrone, D. C. Derrick, and G. S. Ligon, "Individual differences that predict interactions in mixed-initiative teams," in *Proc. of HICSS, Kauai, Hawaii, USA, January 5-8, 2015*, 2015, pp. 610–618.

[45] M. zur Muehlen and M. Indulska, "Modeling languages for business processes and business rules: A representational analysis," *Information Systems*, vol. 35, no. 4, pp. 379–390, 2010.