A concern-oriented sustainability approach

Isabel Sofia Brito
Instituto Politécnico de Beja
Campus do Instituto Politécnico
7800-295 Beja, Portugal
Email: isabel.sofia@ipbeja.pt

José M. Conejero
Quercus Software Engineering Group
Universidad de Extremadura
10003, Spain
Email: chemacm@unex.es

Ana Moreira, and João Araújo
NOVA LINCS, Universidade NOVA de Lisboa
2829-516 Caparica, Portugal
Email: {amm,joao.araujo}@fct.unl.pt

***Abstract*—Sustainability and sustainable development has become a concern worldwide, hence introduced in roadmaps and strategies of public and private organizations. This trend has not been neglected by the computer science community, who is increasingly considering sustainability as a first class entity in software development. To properly address sustainability, its various dimensions need to be reasoned about and their impact on each other and on other system concerns studied from the very early stages of software development. To this purpose, we present a concern-oriented requirements approach that allows both, modeling sustainability concepts and their relationships, and managing conflicting situations triggered by impacts among sustainability dimensions or between those and other system concerns. To tackle the complexity of conflict management, a rigorous trade-off analysis technique based on multi-criteria decision making methods is used to rank, stakeholders and effects between concerns' responsibilities. We use a real project to validate our proposal, discuss the results obtained and synthesize major points that require further research.**

***Index Terms*—sustainability, requirements, conflicts, trade-offs, metamodel**

I. Introduction

Sustainability has become a world's concern. According to the UN Brundtland Commission, sustainable development should meet "the needs of the present without compromising the ability of future generations to meet their own needs"[1]. At the same time, the pervasiveness of the Internet in our daily lives and the increasing need for software systems that must cooperate in environments with a multitude of heterogeneous systems and users, triggers new challenges for novel approaches and solutions. Hence, the next generation approaches should support multiple dimensions of sustainability covering a wide set of purposes, ranging from energy efficiency (environmental sustainability), to cost reduction (economic sustainability), to social capital maintenance (social sustainability). Such purposes are also of target interest for software, where memory and power efficiency, reduction of costs in software development and evolution, and its use for general improvement of people's lives are examples of the topics that must be tackled in software engineering [1].

Given the importance of these topics, as well as their complexity and the so often conflicting impact on each other, we must be able to develop approaches that support their specification and early reasoning for better informed decisions. Therefore, we need to embrace sustainability as an explicit concern from the early stages of software development, starting at the requirements engineering level. The requirements engineering process can contribute to the sustainability of software system development by understanding the organizational goals, identifying stakeholder needs, eliciting requirements, finding metrics to assess realization of the requirements, and evaluating the system's sustainability. During this process, the participation of the systems' stakeholders is fundamental.

Some approaches have emerged to model sustainability related concepts [1], [2], [3], mostly at the modeling level [4]. However, none of these approaches provide a fully-fledged mechanism to address conflict management, for example. In particular, in [2] a metamodel was introduced to model sustainability concepts from the very beginning of software development. However, this metamodel has several limitations regarding the management of conflicts, contribution relationships and trade-off analysis. Thus, the contribution of this paper is threefold:

1) A new metamodel has been introduced that allows modeling sustainability concepts and handling with conflicts. This metamodel is inspired by the Aspect-Oriented Requirements Analysis (AORA) approach [5] and its metamodel.
2) A multi-criteria approach [6], [7] has been adapted with the aim of handling with conflicts in a systematic and rigorous way.
3) A systematic and integrated requirements engineering process has been introduced to identify and define requirements with their relationships and responsibilities, including those related to sustainability concepts.

To achieve this, we will take advantage of the information (e.g., stakeholders, requirements and responsibilities, and priorities different stakeholders may declare on one or more requirements) collected during the identification process.

The metamodel will handle sustainability as a first-class element during the requirements engineering activities, will guarantee consistency and completeness of requirements specifications, and will detail how the various requirements (system and sustainability) may impact on each other. Based

[1]http://www.un-documents.net/our-common-future.pdf

on this metamodel, requirements, including sustainable, are aggregated into concerns. A *concern* refers to a property addressing a problem of interest to one or more stakeholders and which can be defined as a set of coherent requirements.

To handle the conflicts, we use rigorous concepts and techniques from multi-criteria decision systems [8], [9], [10]. We extend the definition of conflict to include situations where stakeholders have different interests on the same set of concerns, and study how the level of granularity and effect (positive or negative contribution) of/between concerns can influence the decision process. Notice that even concerns that affect each other positively may be involved in a conflict if they need to compete with others for their satisfaction, due to scarce resources (e.g., money), for example. So the goal is to rank concerns' effects according to the stakeholders' opinion. The advantage of treating conflicting situations during requirements analysis is to facilitate negotiation and decision-making among stakeholders.

The rest of this paper is organized as follows. Section II offers an overview on sustainability concepts, aspect-oriented analysis used as an inspiration for handling concerns, and summarizes HAM, a multi-criteria decision method for conflict management [6], [7]. Section III presents a generic approach to handle sustainability concerns, including a concern-driven metamodel for sustainability, as well as a stepwise approach to identify and specify concerns, and to resolve conflicts. Section IV shows the use of our approach in the AgroTech project, instantiating the whole metamodel to specify concerns and using a HAM-based technique for conflict handling. Section V discusses some general issues related to the approach and its application to the case study. Section VI presents related work, highlighting similarities and differences with other existing work. Finally, Section VII concludes and suggests directions for future work.

II. Background

The three backbone technologies used in this work are summarized next. We start introducing basic sustainability concepts that will be a fundamental part of our approach, then summarize the AORA (Aspect-Oriented Requirements Analysis) [11] method used as a structural inspiration for our work, and, finally, outline the HAM (Hybrid Assessment Method) [6], [7] framework used to manage conflicting concerns.

A. Sustainability Concepts in Software Engineering

The software engineering community is making a considerable effort to address sustainability as a first-class citizen in software development [12], [13], [14], [15], [16]. Therefore, it is not surprising that several definitions [17] of sustainability have been recently provided, where, for example, software sustainability is defined as a composite, non-functional requirement which is a measure of a system's extensibility, interoperability, maintainability, portability, reusability, scalability, and usability [18]. The Software Sustainability Institute[2] complements stating that the software you use today will be available — and continue to be improved and supported — in the future [17]. These definitions highlight the "relationship" between sustainability requirements and non-functional requirements (NFRs). Amsel et al. consider sustainable software engineering as a process aiming at creating reliable, long-lasting software that meets the needs of users while reducing environmental impacts [19]. Naumann et al. [20] distinguish between *sustainable software*, referring to the product, and *sustainable software development*, referring to the development process. Both these types of sustainability have an impact on the economy, society, human beings and environment. These impacts needs to be evaluated when developing software.

There are several categorizations of sustainability. For example, the United Nations defined a set of ten themes that are mainly categorized into social, economic and environmental dimensions of sustainability Goodlan provided a categorization for general sustainability that is also based on social, economic, environmental and individual dimensions [21]. Penzenstadler and Femmer propose to add technology to these four categories [2]. Our work considers these five dimensions: *individual sustainability* (private goods and individual human capital), *social sustainability* (societal communities, mainly based on solidarity), *economic sustainability* (assets, capital and, in general, added value achieved by the improvement of sustainability in a particular context), *environmental sustainability* (activities performed to improve human welfare by protecting natural resources), and *technical sustainability* (long-time usage of software systems and their adequate evolution over time).

Penzenstadler and Femmer introduced a metamodel for sustainability that allows its instantiation for specific company processes or products [2]. In our previous work [22] we identified limitations of that metamodel, i.e., missing elements to handle conflicts such as effect, stakeholder, and priority and now we address these limitations in an integrated and generic approach to identify and describe concerns, and handle conflicts that can emerge between the various types of con concerns. Handling these limitations are part of the contribution of the present paper.

B. Aspect-Oriented Requirements Analysis Approach

The Aspect-Oriented Requirements Analysis (AORA) approach identifies, modularizes, specifies and composes concerns, including crosscutting concerns. A "concern" is a property addressing a problem of interest to one or more stakeholders and which can be defined as a set of coherent requirements. The AORA metamodel defines AORA's properties and rules elements [5], and the corresponding method identifies and specifies crosscutting and non-crosscutting concerns, detects conflicts in compositions, offers trade-off analysis mechanisms to resolve conflicting situations, defines a schema for semantic data that needs to be

[2]http://www.un-documents.net/ocf-02.htm

stored, defines a language to support a particular methodology or process using the metamodel and another language to express additional semantics on existing information, and, finally, serves as the basis to design and implement better supporting tools.

In a nutshell, AORA is composed of three coarse-grained tasks: identify concerns, specify concerns, and compose concerns. Concerns are identified using typical techniques from Requirements Engineering (e.g., interviews, questionnaires, existing documents) and are specified using a unique template (see example in Table I) supporting a complete and consistent set of descriptions and representations of concerns, be them crosscutting or non-crosscutting. The composition task offers the ability to compose crosscutting and non-crosscutting concerns to clearly understand the system requirements and to identify and analyze critical trade-offs between concerns. The AORA templates and conflict detection and resolution techniques will be adapted and used by our sustainability approach.

TABLE I
CONCERN DESCRIPTION TEMPLATE, ADAPTED AND SIMPLIFIED FROM [11]

Concern name	Name of the concern.
Description	Description of the concern's intended behavior.
Stakeholders	Entities with an interest in a particular decision, including those affected by it.
Stakeholder priorities	Expresses the importance of the concern to a given stakeholder, taking the values: Very Low Important, Important, Medium, Low, Very Important, and Don't Care.
Classification	Sustainable, non-functional, functional.
Responsibilities	List of what the concern must perform, or the information it must maintain.
Contributions	List of concerns responsibilities contributing or affecting this concern and its responsibility. Contribution can be positive (+) or negative (-).

C. HAM: Hybrid Assessment Method

During software development, many decisions need to be made to guarantee satisfaction of the stakeholders' requirements and goals. However, the satisfaction of these requirements and goals may require decisions over conflicting human interests as well as technological alternatives, with an impact on the quality and cost of the final solution. HAM (Hybrid Assessment Method) [6], [7] gives its user the ability to perceive the influence different decisions may have on the final result. HAM is a simple and efficient hybrid Multi-Criteria Decision Making (MCDM) [8], [9], [10] method that combines one single pairwise comparison decision matrix (to determine the weights of criteria) with one classical weighted decision matrix (to prioritize the alternatives). To avoid consistency problems regarding the scale and the prioritization method, HAM uses a geometric scale for assessing the criteria and the geometric mean for determining the alternatives ratings.

HAM is used in two phases, with five steps in total. Each step, and the associated calculations, is explained in detail in [6], [7]. The first phase has three steps, corresponding to an automated determination of weights for the decision weighted matrix, and the second phase starts is composed of the final two steps:

Step 1: Elicit the criteria and the alternatives. A criterion represents a rule on which a judgement has to be made. The alternatives, which represent the options available to the decision maker, have to be classified using the judgments of each criterion.

Step 2: Elicit trade-offs among criteria using pairwise comparisons. We use a geometric scale to rate the relative importance/preference of one criteria over another, and then construct a matrix of the pairwise comparison ratings. This scale was used because, according to [9] and [23], it is the most appropriate for MCDM. The scale summary can be found on Table II.

TABLE II
HAM INTERPRETATION, SCALE AND VALUE

Interpretation	Scale	Value
Extremely High Importance	9	9,000
Very High importance	9/3	3,000
High importance	9/5	1,800
Medium High importance	9/7	1,286
Equal importance	9/9	1,000
Medium Low importance	7/9	0,778
Low importance	5/9	0,556
Very Low importance	3/9	0,333
Extremely Low importance	1/9	0,111

Step 3: Calculate the criteria priority vector, normalize the respective weights and calculate the consistency ratio. To calculate the priority vector for criteria, which will represent the weights for HAM's second phase, we normalize the pairwise comparison matrix by dividing each column cell by the sum of that column. Then, the weights/importance criteria are calculated by using the geometric mean of the normalized pairwise comparison matrix, for all criteria. Next we calculate the consistency of the pairwise matrix to reduce any logical error that might have been introduced during the judgment process. If the consistency ratio is under 10 percent, logical consistency is guaranteed, otherwise, he trade-off values in the pairwise matrix are re-checked and a new iteration is performed until consistency is reached. These 3 steps result in a vector with the importance (weights) for criteria. The next phase (steps 4-5) deals with the prioritization of the alternatives.

Step 4: Identify contributions of each alternative with respect to each criterion, using a decision matrix. We use the same geometric scale of step 2 to identify these contributions.

Step 5: The fifth and final step includes the aggregation of the criteria values, with the geometric average, per alternative, using as weights the values obtained in step 3. At the end of this step, HAM offers a raked list of alternatives (or the ratings for each alternative).

AORA uses HAM for the resolution of conflicts between concerns that contribute negatively to each other and have the same importance, and between concerns triggered by stakeholders with contradictory interests on a set of concerns [7], [24].

III. A CONCERN-ORIENTED REQUIREMENTS APPROACH FOR SUSTAINABILITY

This section starts by introducing the fundamental concepts of sustainability, describing them rigorously in a metamodel. It then follows suggesting a stepwise process model to identify and describe concerns, including the identification and resolution of potential conflicts. Finally, it finishes discussing a conflict management technique based on HAM.

A. A Metamodel for Sustainability

"A metamodel is a precise definition of the constructs and rules needed for creating semantic models"[3]. Metamodels can serve several purposes, particularly [25], [11]: to define a schema for semantic data that needs to be stored, to define a language that supports a particular methodology or process, to define a language to express additional semantics on existing information, to allow a language designer or methodologist to better capture, analyze and understand new approaches, and to serve as a basis to define an automatic supporting tool.

The metamodel for sustainable concerns described in Figure 1 considers those purposes, and is used to collect requirements in natural language (see Section 4). Hence, this metamodel defines sustainable concepts and their relationships at the requirements level addressing limitations of the existing ones [2], such as effects relationships and conflict management. To accomplish this, the metamodel will take advantage of the information collected by AORA (e.g., concerns, classification, stakeholders' priorities and responsibilities), and considers new ones, such as effects and indicators, as explained next. The meta-classes in gray form a general concern-driven metamodel that can be specialized to the various types of concerns (functional, non-functional, sustainable). Here we were concentrating on sustainability concern, defined by the remaining meta-classes in white.

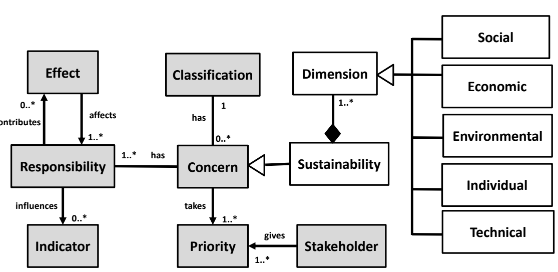

Fig. 1. A metamodel for sustainability concerns.

Here, a *Concern* refers to a matter of interest for the system, and can be classified (*Classification*) into three coherent "clusters": functional (composed of functional requirements), non-functional (equivalent to non-functional requirements), and sustainable (those we are addressing). Sustainable concerns are identified based on the five dimensions of sustainability, as described in Section II-A.

On the other hand, a *Stakeholder* is an entity (person or organization) with an interest or influence on a particular decision, as well as those affected by it. Stakeholders are fundamental to identify system concerns and to help solving conflicting situations during concerns integration, or composition. Sustainability concerns often require engagement and cooperation of multiple stakeholders to ensure agreement and commitment to the sustainability requirements. A stakeholder interest on a concern is expressed by the concerns' *Priority* (given by the scale in Table II). This information is relevant for conflict solving (see Effect).

While a *Responsibility* refers to "an obligation to perform a task, or know certain information" [26], an *Indicator* denotes a qualitative or quantitative metric to express the degree of fulfillment of a particular value (i.e., responsibility) [2]. A good indicator alerts to a problem before it gets too bad and helps recognizing what needs to be done to fix it.

Finally, according to Becker et al. [4], software engineers need to understand the effects by which software system design decisions can enable or undermine the sustainability of socioeconomic and natural systems over time. The Karlskrona Manifesto[4] defines *effects* as impacts or opportunities created by the software system. In this proposal, an *Effect* is a relationship between two concerns through their responsibilities, defining the way in which one concern affects the other. The effect relationship can be positive (or collaborative, helping the affected concern to be achieved, and represented by a "+" sign), or negative (or damaging, obstructing the affected concern, and represented by a "-" sign). It is described as a list of concerns and its responsibilities. For example, conservation of biodiversity in planet Earth, positively influences social sustainability but negatively influences economic sustainability (since it has a cost). Of course, these effects can be indicated for any kind of concern, independently of being sustainable, functional or non-functional. Based on this information conflicts can be handled using HAM. Notice that even concerns that affect each other positively may be involved in a conflict if they need to compete with others for their satisfaction, due to scarce resources (e.g., money).

As a way to evaluate this proposal, our metamodel was instantiated to model the impact of UAVs (Unmanned Aerial Vehicles) in sustainability [22], offering managers of the organization to study the impact of incorporating these technologies and their degree of benefits with respect to the economic, social, individual, environmental and technical sustainability dimensions, and also see the limitations due to strict aerial regulations, for example.

B. Concern-driven sustainability process model

Our approach is composed of three major tasks: identify concerns, specify these concerns in a template that is

[3]Unified Modeling Language Specification in Object Management Group, http://www.omg.org

[4]http://www.sustainabilitydesign.org/

conformant with our metamodel, and, finally, identify and resolve conflicts among concerns.

Identify concerns. As mentioned previously, concerns can be of type functional, non-functional or sustainable. Functional concerns are descriptions of how the system should behave in particular situations. Different types of methods are used to identify these requirements. Here we chose a use case driven approach [27]. Non-functional requirements, on the other hand, are global properties (assumptions, constraints or stakeholders goals) that can influence part or the whole system [28] and can be identified using several approaches, such i* [29] or KAOS [30] or the NFR Framework [28]. Also, we propose the use of existing catalogs, such as the ones developed by Chung [28], to identify non-functional requirements and promote reusability. Notice that some technical concerns (for example, maintainability, reusability, adaptability) are also non-functional concerns. Finally, we use existing catalogs, such as the sustainability indicator (SI) catalog[5], where sustainability concerns and general indicators and extensions (with official measurement values) can be found. In the SI catalog, the authors relate software quality with sustainability and use quality characteristics as indicators. Thus, stakeholders, SI catalogs and frameworks could be used to identify indicators for each sustainable concern and its responsibilities. Because an indicator denotes a metric of a property or a task, the indicator is related to the concerns' responsibility element. Finally, some effects may be found in catalogs such as those in [28], others might be difficult to identify and require experts from several domains (e.g., IoT systems).

Specify concerns. Each concern is described using an extended form of the AORA pattern-like template to include the concept elements of the metamodel. The differences are marked in **bold** in Table III.

TABLE III
CONCERN DESCRIPTION, ACCORDING TO THE METAMODEL

Concern name	Name of the concern.
Description	Concern's behaviour description.
Stakeholders	Stakeholders of the domain.
Stakeholder priorities	Expresses the importance a stakeholder gives to a concern, **using the scale in Table II**.
Classification	Sustainable, non-functional, functional.
Dimension	**Social, Economic, Environmental, Individual, Technical.**
Responsibilities	List of the concern's responsibilities
Indicators	**List of qualitative or quantitative metric of a property or a task. They are related to the concern's responsibility element.**
Effects	**List of concerns responsibilities affecting this concern and its responsibility. Contribution can be positive (+) or negative (-).**

Identify conflicts. Positive and negative contributions between concerns and responsibilities must be identified with the stakeholders' help. A conflict occurs anytime two or more concerns and its responsibilities affect each other negatively (see *Effects* relationship in template of Table III). In this situation, stakeholders are requested to allocate priorities to the involved concerns. For cases where two concerns have the same priority, a trade-off must be negotiated between the stakeholders. Other type of conflicting situation is when a concern affects negatively and positively the same concern (as the concerns' responsibilities affect positively and negatively other concerns — see Table III). For example, conservation of biodiversity positively influences social sustainability but negatively influences economical (since it has a cost). Notice that even concerns that affect each other positively may be involved in a conflict if they need to compete with others for their satisfaction, due to scarce resources (e.g., founds).

Resolve conflicts. Whenever a conflicting situation is identified, HAM is applied and all the stakeholders are involved, not only those with contradictory interests. This is to anticipate a possible situation where a negotiation leads to new conflicts. Of course, this would be the ideal situation, but most times not realistic. So, priorities can also be given by the domain expert, that will validate them with the appropriate stakeholders. The result of applying HAM is a ranking of concerns' effects (alternatives), i. e. concerns/responsibilities, according to the stakeholders' opinions (criteria), where the one with the highest value is the most important. The advantage of treating conflicting situations during requirements analysis is to facilitate negotiation and decision-making among stakeholders. As an example previously mentioned, conservation of biodiversity positively influences social sustainability but negatively influences the economic concern (since it has a cost). From the Government point of view, the positive contribution (social) could be more important than the negative (economic), however, from an organization or company perspective, the contribution to the economic concern could be of utmost importance.

Notice, that we adapted HAM to considerall types of concerns. Furthermore, instead of only ranking concerns, we obtain the ranking of the effects to help decision makers resolve conflicts.

IV. AGROTECH SUSTAINABILITY REQUIREMENTS SPECIFICATION

The AgroTech case study is used to drive the application of the concern-driven sustainability process presented in the previous section. The result is a specification built as an "instantiation" of the metamodel included in that process. The AgroTech case study is derived from a real project developed in Extremadura, Spain, one of the regions with lower industrialization degree in the country. By contrast, its economical model is mainly based on the exploitation of natural resources, namely cattle industry, agriculture and rural tourism. The government of the region encourages the use of new technologies to improve the performance of

[5] http://www.sustainablemeasures.com/node/89

natural resources, and to preserve them. This project's priority was the use of UAVs for sustainability purposes (e.g., early detection of fires, early detection of plagues, distribution of pesticides, detection of robberies in farms, measurement of different kind of environmental indexes (pollution, pollen, acoustic, illumination), animal behavior control (e.g., bird routes, births), and water irrigation decisions. The stakeholders of this project include farmers, the organization, UAV pilots, IT people or the Government. Considering the stakeholders' goals and the catalogs described before [28], regulation, maintainability and availability need to be considered in the AgroTech project. Note that, for instance, UAVs are strongly limited by the strict aerial regulations that they must fulfill. So, an analysis of these applications regulations should be performed to ensure their legal utilization.

A. Concerns identification

As previously described, the first step of the process is to identify the system's concerns. This has been achieved in [22], which reports an initial exploration of the limitations of existing metamodels. Examples of those concerns are: Environmental, Economic and Technical (sustainable) concerns; Regulation (non-functional) concern, Early detection of fires concern, and Plagues management (functional) concern.

B. Concerns specification

The metamodel is reflected in the structure of our concern template, where each meta-class has a corresponding element. For space reasons, the following subsections present a selected sample of concern templates, which includes the three types of concerns.

1) UAVs Environmental Sustainability Concern: Table IV shows the specification for the environmental concern. As pointed out before, the effects between concerns are analyzed based on the stakeholders opinions and NFR catalogs to determine positive and negative impact. For example, based on Table IV, stakeholders may observe the impact of considering the environmental concern onto the economic concern. This impact could be simultaneously positive (+) between reducing natural resources consumption (environmental) and saving money (economic), and negative (-) between conservation of biodiversity (environmental) and saving money (economic). Such a conflict requires a trade-off to be negotiated with the stakeholders.

2) UAVs Economic Sustainability Concern: The resulting specification for the economic concern is in Table V. This specification shows the impact of using drones in the processes but, in this case, related to financial issues. Note that some entities may appear in several dimensions since, for instance, they may influence the environment but also provide financial benefits, e.g., the "water bill" and "energy bill" indicators. As it may be observed, there are several activities performed by drones that influence some economic indicators. For example, the activity "Reduce natural resources consumption" influences the "Water bill" due to the water that may be saved when irrigation is dynamically adapted. This saving in water is translated into a corresponding saving in money. However, it is worth mentioning that the activities performed by drones also influence indicators that imply an increase in the costs for the organization. This is the case of "Cost of UAVs licenses and pilots" and "Cost of UAVs and maintenance" indicators. These indicators reflect new costs that the organization must afford and they help to analyze the effects of the "Evaluate cost of new processes" responsibility.

TABLE IV
TEMPLATE FOR ENVIRONMENTAL ISSUES OF USING UAVS

Concern name: UAVs Environmental Sustainability
Description: Seeks to improve human welfare by protecting natural capital. The dimension includes ecosystems, raw resources, climate change, food production, water, pollution, waste, etc. In [4]: covers the use and stewardship of natural resources. It includes questions ranging from immediate waste production and energy consumption to the balance of local ecosystems and climate change concerns.
Stakeholders UAV pilot, organization
Stakeholder priorities: Very important (organization), important (UAV pilot)
Classification: Sustainable
Dimension: Environmental
Responsibilities:
- Reduce fire areas
- Reduce natural resources consumption
- Reduce energy consumption
- Reduce pollution
- Conservation of biodiversity
- Plague management

Indicators:
- Reduce fire areas -> Burnt forest hectares
- Reduce natural resources consumption -> Water consumption
- Reduce energy consumption -> Energy bill
- Reduce pollution ->Acoustic index, CO2 index, light index

Effects
- Reduce natural resources consumption (water) (+) Economic sustainability/saving money
- Reduce energy consumption (+) Technological sustainability/<ALL>
- Conservation of biodiversity (–) Economic sustainability/Saving money
- Conservation of biodiversity (+) Social Sustainability/Improve welfare

Note that some "Responsibilities" cut across the Environmental and Economic concerns. "Reduce natural resources consumption" in Table IV and V is needed in both concerns. Finally, as said before, the effects between concerns are analyzed based on the stakeholders' opinions and NFR catalogs to determine positive or negative influences. Table V shows that the economical concern helps the environmental concern. So, while in general, we see that the environmental dimension hurts the economic dimension (it is more expensive to comply with sustainability regulations than not, for example), at a lower granularity level, responsibilities of these dimensions can in fact help each other.

3) UAVs Technical Sustainability: The technical dimension of sustainability fosters technology that may be easily adapted to future changes, guaranteeing long-term use. From a software engineering perspective, achieving this goal is clearly related to well-known quality indicators that influence the long-term use of a software product, e.g., adaptability, flexibility, maintainability and reusability. Table VI shows the

TABLE V
TEMPLATE FOR ECONOMIC ISSUES OF USING UAVS

Concern name: UAVs Economic Sustainability
Description: The impact of using drones based on financial issues. Aims at maintaining assets in terms of capital and added value
Stakeholders UAV pilot, organization, IT people, farmer
Stakeholder priorities: Very important (organization, IT people, UAV pilot and farmer
Classification: Sustainable
Dimension: Economic
Responsibilities:
- Reduce cost of reforesting
- Reduce natural resources consumption
- Reduce energy consumption
- Reduce robberies at farms
- Evaluate cost of new processes
- Reduce costs caused by pests

Indicators:
- Reduce natural resources consumption -> Water bill
- Reduce energy consumption -> Energy bill
- Evaluate cost of new processes -> Cost of UAV, including licenses, maintenances and pilots; Water bill; Pesticides bill
- Reduce costs caused by pests -> Pesticides bill

Effects
- Reduce natural resources consumption (+) Environmental/Reduce natural resources consumption
- Reduce energy consumption (+) Environmental/Reduce energy consumption
- Reduce costs caused by pests (+) Plague management

TABLE VI
TEMPLATE FOR TECHNICAL ISSUES OF USING UAVS

Concern name: UAVs Technical Sustainability
Description: Enhancing UAVs' quality characteristics in the software products built to control the devices and, also, in the construction of the devices (e.g. reusing pieces or generating low cost ones). Also, quality indicators that influence this longterm use of a software product, e.g. adaptability, flexibility, maintainability and reusability.
Stakeholders UAV pilot, organization, IT stakeholder
Stakeholder priorities: Very important (IT stakeholder and UAV pilot), important (organization)
Classification: Sustainable
Dimension: Technical
Responsibilities:
- Reuse hardware and software components
- Preserve hardware components
- Adapt software to regulation changes
- Optimize the flight routes

Indicators:
- Reuse hardware and software components -> Costs of building and programming a new UAV
- Preserve hardware components -> Costs of maintain UAVs
- Adapt software to regulation changes -> Costs of maintaining an application
- Optimize the flight routes -> Number of routes, battery usage

Effects
- Reuse and preserve hardware and software components (+) Economic/Costs of building a new UAV
- Reuse and preserve hardware and software components (+) Environmental/reduce natural consumption
- Optimize the flight routes (+) Economic/Reduce energy consumption
- Optimize the flight routes (+) Environmental/reduce natural consumption
- Adapt software to regulation changes (-) Economic/ Costs of building a new UAV

technical sustainability concern for our motivating example.

For example, it is worth remarking that an important activity to ensure a long-term usage of software is the implementation of adaptive software (also claimed in [1] [31]). Adaptability is also a well-known NFR, so non-functional requirements techniques could be applied to address this concern. Considering the "Effects" information, Technical concern affects negatively (-) and positively (+) the Economic concern, causing a conflict and hence requiring a trade-off to be negotiated with the stakeholders (see Figure 2). For instance, in Table VI, a positive (+) effect is indicated because reuse and preserve hardware and software components would help the Economic/Costs of building a new UAV, and negative (-) effect because adapting the software to regulate changes increases the Economic/Costs of building a new UAV.

4) UAVs non-functional Regulation Concern: The AgroTech specification also shows some regulations that may hinder the achievement of the system requirements. We used the "Regulation" concern to represent these limitations. Table VII summarizes the specification. Taking into accounts the effects shown, we can see that "Avoid populated areas" contributes negatively (-) to "Reduce natural resources consumption" since UAVs routes may not be optimized because of the aerial space restrictions. Similarly, we can see how the responsibility "Fly lower than 120m in populated areas" has a negative (-) contribution to the "Reduce pollution" Environmental responsibility since noise pollution may be generated by these flights. Also, "Regulation" affects negatively (-) the "Early detection of fire" functional concern since the detection of fires may imply flights that are not

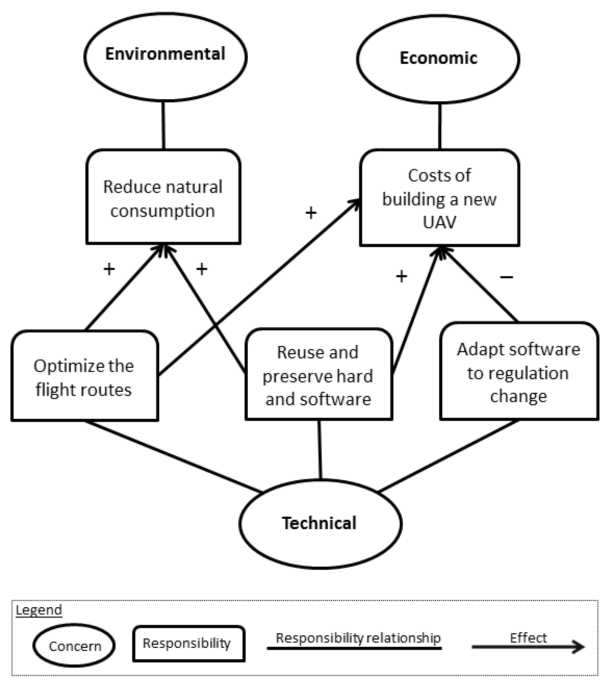

Fig. 2. Technical concern effects

compliant with the regulations (e.g. flying autonomously over a populated area). By contrast, we can see that all these regulations are positively contributing to ”Adapt software to regulation changes” Technical responsibility because of the need of building software that deals with all the regulations and their changes (e.g. just for complying with regulations from different countries).

TABLE VII
TEMPLATE FOR REGULATION CONCERN

Concern name: UAVs Regulation **Description:** Regulation applied to UAV **Stakeholders** Government, UAV pilot and population **Stakeholder priorities:** Very important (Government, UAV pilot and population) **Classification:** Non-functional **Responsibilities:** - Avoid populated areas - The UAV Flight must be supervised by a certified pilot - Fly lower than 120m in populated areas - Fly closer than 100m from the pilot - National Aerial restrictions **Indicators:** - <none> **Effects** - Avoid populated areas (-) Environmental/ Reduce natural resources consumption - Fly lower than 120m in populated areas (-) Environmental/Reduce pollution - Fly closer than 100m from the pilot (-) Environmental/Reduce pollution - Avoid populated areas (-) Plague management - Avoid populated areas (-) Early detection of fires - <ALL responsibilities> (+) Technical/Adapt software to regulation changes

5) UAVs Functional Concerns: The steps to specify functional concerns are similar. Basically, the concern template is used to reflect the metamodel entities. In our case study, the functional concerns considered are ”Early detection of fires”, and ”Plagues management”. Due to space limitations, we only illustrate ”Early detection of fires” (see Table VIII). For example, “Fly region with thermal cameras” responsibility will be carried out by drones equipped with those kind of cameras that will perform programmed flights in fire-risk regions of the forests.

This concern has many similarities with the “Plagues management” concern, since UAVs are also used to achieve their responsibilities. For example, “the UAV should fly with thermal cameras” responsibility and “The UAV should fly with thermal cameras (+) Environmental/Reduce fire areas” effect, among others, appear in both concerns.

C. Conflict identification and management

HAM is used to manage concerns conflicts, and is illustrated here with the “Technical concern”. Based on the “Effect” row, we identified conflicts triggered by negative contributions, such as ”Adapt software to regulation changes (-) Economic/ Costs of building a new UAV” (see Figure 2).

TABLE VIII
TEMPLATE FOR EARLY DETECTION OF FIRES CONCERN

Concern name: Early detection of fires **Description:** Using UAVs for early detection of fires. The UAVs are characterized by being able to take off and on from a static position. **Stakeholders** Fireman, UAV pilot **Stakeholder priorities:** Very important (Fireman), important (UAV pilot) **Classification:** Functional **Responsibilities:** - The UAV should fly with thermal cameras and detect smoke/fire - The UAV should detect dry zones and humidity - The UAV should fly around regions with sensor and detect smoke/fire - The UAV should preserve privacy during flights **Indicators:** - Number of hectares burnt per month - Average time in extinguishing a fire **Effects** -The UAV should fly with thermal cameras (+) Environmental/Reduce fire areas - Detect dry zones and humidity (+) Environmental/ Reduce natural resources consumption - The UAV should fly around regions with sensor (+) Environmental/ Reduce pollution - The UAV should fly with thermal cameras (+) Economic/Reduce cost of reforesting - Detect dry zones and humidity (+) Economic/Reduce natural resources consumption - The UAV should preserve privacy during flights (+) Social/preserve privacy

Step 1: Identify criteria and alternatives: Given that we also want to consider situations where different stakeholders may have contradictory interests in those concerns, the conflict analysis will consider all the concerns’ stakeholders involved in a conflict and also the stakeholder priorities. Considering that even concerns that affect each other positively may be involved in a conflict (when they compete for the same resource, for example), the list of concerns responsibilities contributing or affecting the Technical concern and its responsibilities correspond to the following HAM alternatives: Reuse and preserve hardware and software components (+) Economic/Costs of building a new UAV; Reuse and preserve hardware and software components (+) Environmental/reduce natural consumption; Optimize the flight routes (+) Economic/cost; Optimize the flight routes (+) Environmental/reduce natural consumption; Adapt software to regulation changes (-) Economic/ Costs of building a new UAV.

The criteria are the concerns’ stakeholders: UAV pilot, Organization, IT people, Farmer and Government.

Step 2: Elicit trade-offs among criteria: As mentioned, HAM uses a pairwise comparison matrix to determine the ratings for the stakeholders. The values of this matrix are calculated based on the scale from Table II. The pairwise matrix for the Technical concern is shown in Table IX. Because the case study documentation does not present the required information, the pairwise matrix uses illustrative judgments. As we can see, UAV Pilot has low importance

(5/9=0,556) when compared with IT people. Each stakeholder has a priority compared with the rest of stakeholders (the value "1,000" is used when compared to itself).

Step 3: Calculate the criteria priority vector, normalize the respective weights and calculate the consistency ratio: HAM calculates the normalized priority vector (or stakeholders' weights). The priority vector for our conflict is (ordered by importance): IT People: 0,2751; Organization: 0,1979; Government: 0,1864; UAV pilot: 0,1754; Farmer: 0,1652. The stakeholder IT People is the most important stakeholder according to the values in the priority vector (0,2751). The next step is to calculate the value of the logical consistency ratio, to ensure that the analyst consistently elicited the stakeholders' priorities. In this case the ratio is 3,08% (logical consistency is guaranteed when value if lower than 10%).

Step 4: Identify contributions of each alternative: We elicit the contributions of each "Effect" between concerns with respect to each stakeholder, using a regular decision matrix. The contributions are illustrative values because the case study's documentation does not have this information. This matrix may be observed in Table X. The effect of the positive contribution between the responsibilities "Reuse and preserve hardware and software components" and "Economic/ Costs of building a new UAV" (see Table VI) is high (9/5=1,8) for UAV Pilot (see Table X).

Step 5: Calculate the alternatives ratings: Let us now discuss the final ranking. First, we start analyzing the concerns. As we can see in the results obtained in Table X, the Economic concerns have higher significance than the Environmental one, from the Technical concern point of view. Analyzing the responsibilities, for example, the Technical responsibility "Reuse and preserve hardware and software components" positively contributes (+) to several other concerns' responsibilities, namely to the "Costs of building a new UAV" of the Economic concern and the responsibility "Reduce natural consumption" from the Environmental concern. However, we can also see that the responsibility "Adapt software to regulation changes" has also a negative (-) contribution in the costs of building new UAVs since it implies creating software that is context adaptive. The Technical responsibility "Adapt software to regulation" contributes negatively (-) to the Economic concern responsibility "Costs of building a new UAV", but it also contributes positively (+) to the Economic responsibility "Costs of building a new UAV", based on the responsibility "Reuse and preserve hardware and software". This is a conflicting situation that needs to be handled by HAM. According to Table X the positive contribution has more importance to the stakeholders than the negative one.

V. Discussion

This section discusses some issues from the concern-driven approach perspective that have been triggered by its application to our running example. Note that our goal is not to contribute to the UAVs domain, but to approaches to handle sustainability concerns integrated with other kinds of concerns.

Positive and negative effects. Identifying effects is hard, despite the use of catalogs and access stakeholders' knowledge. For example, we claimed that the use of drones may drastically reduce the fuel consumed when these devices are used to replace traditional vehicles. However, since drones usually rely on batteries, their autonomy may be limited and, thus, the number of drones needed to cover a big area may be higher than the number of traditional vehicles.

However, the inclusion of Effects contribution in our metamodel allows the specification of these situation. In that sense, effects (positive or negative) can be indicated for any kind of concern, independently of being sustainable, functional or non-functional. An example between two sustainable concerns is the effect Technical/Adapt software to regulation changes (-) Economic/Costs of building a new UAV. A different example between non-functional and sustainable concerns is Regulation/Avoid populated areas (-) Economic/Evaluate cost of new process. Finally, the effect Early detection of fires/UAV should fly with thermal cameras (+) Environmental/Reduce fire areas is an example between functional and sustainable concerns.

Trade off analysis. Considering the effects relationships, several possible conflicting situations may arise in several situations:

- A conflict occurs anytime two or more concerns and their responsibilities affect each other negatively. If at least two concerns have the same priority, a trade-off must be negotiated with the stakeholders.
- Additionally, other conflicts may occur when concerns affect each other negatively and positively through their internal responsibilities. For example, a concern responsibility may affect positively another concern responsibility and negatively another responsibility. Such situations also require a trade-off analysis to be performed to evaluate the global impact of the effect.
- Even concerns that affect each other positively may be involved in a conflict if they need to compete with others for their satisfaction, due to scarce resources (e.g., cost).

We take advantage of the information collected by this approach and the flexibility of the multi-criteria concepts and techniques to solve conflicts, and also consider conflicts triggered by the list of effects in the concern specification.

Section III-B, by using an adaptation of HAM and according to the Stakeholders' opinion, a ranking of concerns' effects is provided that supports the trade off analysis needed in these situations.

Priority. Allocating priorities is difficult due to the required expertise in the domain and the knowledge of the organizational business values and goals. Here we use priorities in two different moments. First, we prioritize concerns using a fuzzy scale of five values, ranging from Very Important to Very Low. Then, in HAM's first phase we prioritize stakeholders using a geometric fuzzy scale with nine

TABLE IX
PAIRWISE COMPARISON MATRIX FOR TECHNOLOGICAL CONCERN

	Stakeholders						
Stakeholders	UAV Pilot	Organization	IT People	Farmer	Government	**Priority Vector**	**Priority Ranking**
UAV Pilot	1,000	0,889	0,556	1,286	0,889	1,1754	4
Organization	1,125	1,000	0,556	1,286	1,286	0,1979	2
IT People	1,800	1,800	1,000	1,286	1,286	0,2751	1
Farmer	0,778	0,778	0,778	1,000	0,889	0,1652	5
Government	1,125	0,778	0,778	1,125	1,000	0,1864	3

TABLE X
COMPARISON MATRIX FOR EFFECTS AND STAKEHOLDERS

Geometric scale							
Stakeholder Priority Vector [NW]	0,1754	0,1979	0,2751	0,1652	0,1864		
Effects between Concerns / Stakeholders	UAV Pilot	Organization	IT People	Farmer	Government	**Priority Vector**	**Priority Ranking**
Reuse and preserve hardware and software components (+) Economic/ Costs of building a new UAV	1,800	3,000	3,000	0,550	3,000	34,69%	1
Reuse and preserve hardware and software components (+) Environmental/Reduce natural consumption	1,800	1,800	0,550	1,8000	3,000	23,94%	3
Optimize the flight routes (+) Economic/Saving money	1,800	3,000	0,550	0,550	3,000	21,82%	4
Optimize the flight routes (+) Environmental/Reduce natural consumption	0,770	1,800	0,550	1,280	3,000	19,55%	5
Adapt software to regulation changes (-) Economic/Costs of building a new UAV	1,280	3,000	1,280	0,550	3,000	25,90%	2

values (from Extremely High Importance to Extremely Low Importance). Finally, in HAM's second phase we prioritize effects using a geometric scale of nine values (from Extremely High Importance to Extremely Low Importance).

Indicator. Similarly, identifying indicators is not easy (those depend on the stakeholder point of view and their knowledge, as well as the problem domain). In this paper, we use the indicators catalogs[6] [7]. This however, is not sufficient, as it is very generic and needs to be adapted to the problem's context. Hence, for future work we suggest establishing a measure (scale) for sustainable indicators.

Refinement. Sustainability is a complex domain, affecting various dimensions of our lives, each dimensions discussed and included in the presented metamodel. The granularity level at which each dimension is studied here is still very high. So, we are investigating ways to refine sustainability concerns. Handling with sustainability is more complex than handling with non-functional requirements, although we know that quality attributes (or non-functional requirements in general) are intrinsic elements of sustainability. We also need to better understand and formulate the impact between the various complex elements forming each dimension and the target software system.

Crosscutting. In this work, we could identify responsibilities that cut across several other concerns. For example, "Reduce natural resources consumption" responsibility is the same for the Environmental and the Economic concerns. These situations need to be identified so that the target system can be better modularized, keeping such responsibilities encapsulated in separate modules. Our believe is that similar situations exist with respect to concerns that are part of the refinements of more coarse-grained concerns. Modularizing such type of crosscutting elements will increase reuse and facilitate evolution.

[6]https://www.gdrc.org/sustdev/indicators.html
[7]http://www.sustainablemeasures.com/node/89

VI. RELATED WORK

Some existing works provide mechanisms and frameworks to model sustainability concepts in software engineering [32], [2], [20], [3], [33], [1], [34], [35], while others focus on sustainability as quality [36], [37], [38], [39], [40].

Sustainability modeling and metamodeling. In [32], [2], Penzenstadler and Femmer introduce a metamodel used to instantiate generic models for sustainability. Naumann et al. [20] proposed a framework to help developers implement sustainable web pages. Cabot et al. [3] shows an example of how to model sustainability goals using i* models. Mussbacher and Nuttall [33] also use goal oriented models, but focus on specific requirements related to time cost. Combemale et al. [1] highlight the importance of sustainability models, claiming that engineering models should be complemented with scientific models as a way to not only build a sustainable products but also to understand the influence between the various dimensions of sustainability and balance the trade-offs among them. Roher and Richardson [34] propose a sustainability requirement pattern to guide on how to write specific types of sustainability requirements. The pattern provides information for its use, a starting point for developing sustainability requirements, and other relevant information needed to develop such requirements. Saputri and Lee In [35] proposed an approach to determine the sustainability requirements from stakeholders needs. The approach consists of determining sustainability goals using GQM, followed by the analysis of sustainability properties where the impact and trade-off analysis of those requirements is performed. Finally, sustainability requirements are mapped into a runtime model for adaptability. A metamodel for sustainability is also provided, but without adopting, as we do, a concerning-driven approach or considering concern effects at a more fine grained (responsibility) level, which has an impact on the precision of the trade-off analysis. Unlike all these approaches, the work presented in this paper focuses on providing a new instantiation of a sustainability model for a concrete domain whose applications are usually related to sustainability. Additionally, we describe concerns systematically and manage conflicts using a rigorous technique.

Note that most of these approaches (including ours) consider sustainability at a high abstraction level (modeling) and at an early stage of development (requirements). The importance of considering sustainability from the very beginning of the development was also highlighted in [4].

Sustainability as quality. The importance of sustainability in software engineering has been also treated from a quality perspective. Some works have extended traditional quality frameworks (like ISO/IEC 25000) with new concepts related to sustainability [36] or analyzed the relationship between sustainability concerns and other quality requirements [37], [38], especially security [39]. In that sense, the model for sustainability presented here helps to specify the relations among drones and sustainability and also the security restrictions imposed by local regulations. In recent works on software sustainability there has been an increasing interest in understanding the impact of design decisions on the sustainability of software systems. In [40] a survey was applied to different stakeholders to determine which quality requirements contribute to the sustainability of software-intensive systems with respect to the sustainability dimensions. In short, they argue that the key challenge for software sustainability is its characterization as a software quality requirement. The difference between this work and the one presented here is that, in our case, quality requirements and sustainability requirements are treated as first-class entities and in a "modularized" way, i.e., both are treated as equals and in the same fashion, and the decision process is based on effects between all types of concerns (functional, non-functional and sustainable).

VII. CONCLUSIONS

This paper presents a sustainability approach to identify and describe concerns (which may be functional, non-functional or sustainable), and to handle conflicts among concerns. Our approach offers a metamodel organized as a general concern-driven model which is then extended (or specialized) to express additional semantics for sustainability concepts, such as, effects, priorities and trade-off analysis (thus addressing limitations identified in existing ones). This metamodel allows a language designer to better capture, analyze and understand the approach and offers an abstract syntax to tool developers. The trade-off mechanism offered is based on multi-criteria decision making techniques, what results in a rigorous process of resolving conflicting situations by ranking the involved elements according to a combined set of priorities given to stakeholders, concerns and effects in different moments of the development process. The technique used guarantees the consistency of the allocated priorities, allowing a drastic reduction of the number of combined pairwise comparison matrices traditionally required.

Our plan for future work is to include refinements of sustainability concerns (and potential responsibilities), establish a measure (scale) for sustainable indicators, reduce the granularity level at which each sustainability dimension is studied, propose a systematic way to identify crosscutting responsibilities and concerns so that reuse can be increased and evolution of the target system facilitated.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of NOVA LINCS Research Laboratory (Ref. UID/CEC/04516/2013), UID/CEC/4668/2016(LISP), TIN2015-69957-R (MINECO/FEDER, UE) project and Junta de Extremadura (Spain)- European Regional Development Fund (ERDF)- GR15098 and IB16055 projects to the work presented here.

REFERENCES

[1] B. Combemale, B. H. C. Cheng, A. Moreira, J.-M. Bruel, and J. Gray, "Modeling for sustainability," in *Proceedings of the 8th International Workshop on Modeling in Software Engineering*, ser. MiSE '16. New York, NY, USA: ACM, 2016, pp. 62–66. [Online]. Available: http://doi.acm.org/10.1145/2896982.2896992

[2] B. Penzenstadler and H. Femmer, "A generic model for sustainability with process- and product-specific instances," in *Workshop on Green in/by software engineering - GIBSE*. New York, New York, USA: ACM Press, 2013, pp. 3–8. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2451605.2451609

[3] J. Cabot, S. Easterbrook, J. Horkoff, L. Lessard, S. Liaskos, and J.-N. Mazon, "Integrating sustainability in decision-making processes: A modelling strategy," in *31st International Conference on Software Engineering*. IEEE, 2009, pp. 207–210. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5070983

[4] C. Becker, S. Betz, R. Chitchyan, L. Duboc, S. M. Easterbrook, N. Seyff, and C. C. Venters, "Requirements : The Key to Sustainability," *IEEE Software*, vol. 33, no. 1, pp. 56–65, 2016.

[5] I. S. Brito, A. Moreira, J. M. Lopes, and J. Araújo, "A metamodel for aspect-oriented analysis approach," in *XIII Congresso Ibero-americano em Software Engineering*. CIbSE2010, 2010, p. 199–212.

[6] A. Pimentel, "Multi-criteria analysis for architectural choices in software product lines," 2009.

[7] R. A. Ribeiro, A. M. Moreira, P. van den Broek, and A. Pimentel, "Hybrid assessment method for software engineering decisions," *Decis. Support Syst.*, vol. 51, no. 1, pp. 208–219, Apr. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2010.12.009

[8] S.-J. J. Chen and C. L. Hwang, *Fuzzy Multiple Attribute Decision Making: Methods and Applications*, M. J. Beckmann and W. Krelle, Eds. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1992.

[9] E. Triantaphyllou, *Multi-Criteria Decision Making Methods: A Comparative Study*. Kluwer Academic Publishers, 2000.

[10] H. Zimmerman and L. Gutsche, *Multi-Criteria Decision Making Methods: A Comparative Study*. Springer, 1991.

[11] I. Brito, "Aspect-oriented requirements analysis," Ph.D. dissertation, Universidade NOVA de Lisboa, 2008.

[12] P. G. Neumann, "The foresight saga," *Communications of the ACM*, vol. 55, no. 10, pp. 26–29, 2012. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1151030.1151060

[13] B. Penzenstadler, A. Raturi, D. Richardson, C. Calero, H. Femmer, and X. Franch, "Systematic mapping study on software engineering for sustainability (SE4S)," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14*. New York, New York, USA: ACM Press, 2014, pp. 1–14. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2601248.2601256

[14] C. Becker, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, N. Seyff, and C. C. Venters, "Sustainability Design and Software: The Karlskrona Manifesto," in *Proceedings - International Conference on Software Engineering*, vol. 2, no. May, 2015, pp. 467–476.

[15] P. Lago, "Challenges and opportunities for sustainable software," in *Proceedings of the Fifth International Workshop on Product LinE Approaches in Software Engineering*. IEEE Press, 2015, pp. 1–2.

[16] M. Couto, P. Borba, J. Cunha, J. a. P. Fernandes, R. Pereira, and J. a. Saraiva, "Products go green: Worst-case energy consumption in software product lines," in *Proceedings of the 21st International Systems and Software Product Line Conference - Volume A*, ser. SPLC '17. New York, NY, USA: ACM, 2017, pp. 84–93. [Online]. Available: http://doi.acm.org/10.1145/3106195.3106214

[17] C. Venters, C. Jay, L. Lau, M. Griffiths, V. Holmes, R. Ward, J. Austin, C. Dibsdale, and J. Xu, "Software Sustainability: The Modern Tower of Babel," in *International Workshop on Requirements Engineering for Sustainable Systems*. CEUR, 2014, pp. 7–12.

[18] C. C. Venters, L. Lau, M. K. Griffiths, V. Holmes, R. R. Ward, C. Jay, C. E. Dibsdale, and J. Xu, "The Blind Men and the Elephant Towards an Empirical Evaluation Framework for Software Sustainability," *Jorunal of Open Research Software*, vol. 2, no. 1, pp. 1–6, 2014.

[19] N. Amsel, Z. Ibrahim, A. Malik, and B. Tomlinson, "Toward sustainable software engineering," in *Proceeding of the 33rd international conference on Software engineering - ICSE '11*. New York, New York, USA: ACM Press, 2011, pp. 976–979. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1985793.1985964

[20] S. Naumann, M. Dick, E. Kern, and T. Johann, "The GREENSOFT Model: A reference model for green and sustainable software and its engineering," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 4, pp. 294–304, 2011.

[21] R. Goodland, "Sustainability : Human , Social , Economic and Environmental," in *Encyclopedia of Global Environmental Change*, W. &. Sons, Ed., 2002, vol. 6, no. 11, pp. 220–225. [Online]. Available: www.balticuniv.uu.se/index.php/download/doc_download/435-sustainability-human-social-economic-and-environmental

[22] J. M. Conejero, I. S. Brito, A. Moreira, J. Cunha, and J. Araújo, "Modeling the impact of uavs in sustainability," in *24th IEEE International Requirements Engineering Conference Workshops (REW)*, 2016, p. 208–216, n/a. [Online]. Available: https://doi.org/10.1109/REW.2016.044

[23] Y. Dong, Y. Xu, H. Li, and M. Dai, "A comparative study of the numerical scales and the prioritization methods in ahp," *European Journal of Operational Research*, vol. 186, no. 1, pp. 229–242, 2008. [Online]. Available: https://EconPapers.repec.org/RePEc:eee:ejores:v:186:y:2008:i:1:p:229-242

[24] I. S. Brito, F. Vieira, A. Moreira, and R. A. Ribeiro, *Handling Conflicts in Aspectual Requirements Compositions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 144–166. [Online]. Available: https://doi.org/10.1007/978-3-540-75162-5_6

[25] J. Bézivin, "Tutorial on model driven engineering: Principels, scope, deployments and applicability," in *International Summer School, Generative and Transformational Technique in Software Engineering (GTTSE)*, J. V. Ralf Lämmel, João Saraiva, Ed. Springer, 2005, pp. 1–33.

[26] R. Wirfs-Brock, L. R. Wiener, and B. Wilkerson, *Designing object-oriented software.* Prentice Hall, 1990.

[27] G. Schneider and J. P. Winters, *Applying Use Cases: A Practical Guide*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.

[28] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*. Boston/Dordrecht/London: Kluwer Academic Publishers, 2000.

[29] E. Yu, "Modelling strategic relationships for process reengineering," Ph.D. dissertation, University of Toronto, Canada, 1995.

[30] A. Van Lamsweerde, "The kaos project: Knowledge acquistion in automated specification of software," in *Proc. AAAI Spring Symposium Series, Stanford University, American Association for Artificial Intelligence, March 1991*, 1991.

[31] R. Chitchyan, S. Betz, L. Duboc, B. Penzenstadler, C. Ponsard, and C. C. Venters, "Evidencing sustainability design through examples," in *RE4SuSy Workshop at Requirements Engineering Conference*, vol. 1416, 2015, pp. 45–54.

[32] B. Penzenstadler and H. Femmer, "A Generic Model for Sustainability," Tech. Rep., 2012.

[33] G. Mussbacher and D. Nuttall, "Goal modeling for sustainability: The case of time," in *4th International Model-Driven Requirements Engineering Workshop, MoDRE*, Karlskrona, Sweden, 2014, pp. 7–16.

[34] K. Roher and D. J. Richardson, "Sustainability requirement patterns," in *Third IEEE International Workshop on Requirements Patterns, RePa 2013, Rio de Janeiro, Brazil, July 16, 2013*, 2013, pp. 8–11. [Online]. Available: https://doi.org/10.1109/RePa.2013.6602665

[35] T. R. D. Saputri and S. Lee, "Incorporating sustainability design in requirements engineering process: A preliminary study," in *Requirements Engineering Toward Sustainable World - Third Asia-Pacific Symposium, APRES 2016, Nagoya, Japan, November 10-12, 2016, Proceedings*, 2016, pp. 53–67. [Online]. Available: https://doi.org/10.1007/978-981-10-3256-1_4

[36] C. Calero, M. angeles Moraga, M. F. Bertoa, and L. Duboc, "Quality in use and software greenability," in *RE4SuSy Workshop at Requirements Engineering Conference*, vol. 1216, 2014, pp. 28–36. [Online]. Available: http://ceur-ws.org/Vol-1216/paper6.pdf

[37] N. C. Fernandez, P. Lago, and C. Calero, "How do Quality Requirements Contribute to Software Sustainability?" in *REFSQ Workshop*, 2016, pp. 7–10. [Online]. Available: http://ceur-ws.org/Vol-1564/paper33.pdf

[38] S. A. Koçak, G. I. Alptekin, and A. B. Bener, "Integrating Environmental Sustainability in Software Product Quality," 2015.

[39] G. Rodriguez-Navas, L. Duboc, S. Betz, R. Chitchyan, B. Penzenstadler, and C. C. Venters, "Safety vs. sustainability design: Analogies, differences and potential synergies," *CEUR Workshop Proceedings*, vol. 1416, pp. 25–34, 2015.

[40] N. Condori-Fernandez and P. Lago, "Characterizing the contribution of quality requirements to software sustainability," *Journal of Systems and Software*, vol. 137, pp. 289–305, Dec. 2017.