

National Body ID	ID	Category (TH, TL, E, G)	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Disposition
IEEE	NB-1	TH	1	(1)	The scope of the standard has changed from 'software intensive systems' to 'general systems'. To my opinion, the term 'general systems' is too broad to be useful. Especially, there is nothing in the standard that represents systems that encloses 'humans, processes, procedures, facilities, materials and naturally occurring entities (e.g. water, organisms, minerals)'. The text of the standard, the references and examples are all clearly focused on software and not on this broad definition of 'general system', which is confusing. I'm far more interested in a standard that focuses on the description of a software architecture and the relation of this software architecture to its context (i.e. this could be hardware, data, humans, processes, procedures, ...).	The scope of this standard encompasses those products of system and software development that captures the architectural description of the software and the relation to its context, i.e. how it fits in a general system.	Revision is required to align with the scope of SC7 which includes engineering of general systems.
IEEE	NB-2	E	1	(1)	The first sentence 'The scope of ...' reads very badly, please separate this sentence in two parts (at the colon).		All editorial comments will be considered for WD2 without further disposition in this list.
IEEE	NB-3	E	1.2	(5)	The last bullet in the list has a different formatting and style of explaining then the previous bullets. Please make this uniform.	1. correct indentation, 2. examples between brackets	
IEEE	NB-4	TH	1.3	(2)	I strongly disagree that the standard imposes no requirements about the life cycle processes, architecture methods, language notation or models employed in the architectural descriptions. The standard explicitly requires that: (1) there is an architectural phase in which an architecture is described; (2) that the architectural description language, method or notation supports multiple views; ...	The standard imposes several minimal requirements before it can be uses: (1) there is an architectural phase in which an architecture is described; (2) the stakeholders and their concerns can be explicitly identified; and (3) the architectural description language, method or notation supports multiple views. For the remainder the architecture imposes no requirements about the life cycle processes, architecture methods, language notation or models employed in the architectural descriptions.	Reject. But perhaps this could be written more clearly. E.g., "imposes no requirements" or "does not prescribe" versus "makes no assumptions"...
IEEE	AT-1	E	2	(3)	I cannot find Clause 8 in this draft ISO 42010 document		Fix: references to 7 and 8 should be to 6 and 7, respectively.
IEEE	NB-5	TL	4		The order of the definition is not logical. The definition of a term that is used in another definition should come earlier in de list. E.g. architecture concern is 4.16, but is used for example used in 4.6 and 4.4.	I suggest to order the definition in a way that a definition only depends on definition that are specified earlier in the text.	Reject. ISO rules require definitions appear in alphabetical order.
IEEE	AT-2	TH	4.10	(1)	The scope of the circumstances in the statement "The context which determines the setting and circumstances of developmental, operational, political, regulatory, social, and other critical influences upon a system." should be broader. Often in the industry, the following are important environmental factors: (a) technological influences such as what COTS capabilities; (b) organizational influences such as management's preferences and budgeting.	Amend text to "The context which determines the setting and circumstances of developmental, technological, organizational, operational, political, regulatory, social, and other critical influences upon a system."	Accept in principle (AIP): we intend concerns to include all of these influences.
IEEE	AT-3	E	4.16	(1)	Consider moving this clause to 4.7 since it is referenced by the current 4.6 and 4.13		Reject
IEEE	AT-4	TH	4.16	(1)	The definition in this clause lacks two key aspects of concerns. They are "business" and "technical". If the architectural concerns form the basis of a view point (clause 4.6), then common viewpoints such as business viewpoint and technical application should be included. Examples are architectural frameworks such as Zachman, RM-ODP and TOGAF. This is also noted in Note 5 of clause 5.1.	Amend text to "An area of interest in a system pertaining to business, technical, developmental, operational, political, regulatory, social, and other influences which is important to one or more system stakeholders."	AIP. The intent of the Standard has always been to include business concerns and technical concerns, these are special cases of the definition.
IEEE	NB-6	TH	4.17		There is inconsistency in the definition of a view correspondence in 4.17 and 6.5.1. The definition of 4.17 explicitly mentions relations 'between elements of two architectural views', while 6.5.1 mentions that a view correspondence records 'a relation between two architectural views'. I like the second definition more, since it allows more types of relations between views. Relations between elements of the views would be a specific class of relations between views in general. For example, the definition of 4.17 excludes the 'Related view packets' relations (that can be 'child', 'parent' or 'sibling') that is used in Documenting software architectures (Clements et al.), because these relations are not defined on the level of architectural elements. The definition of 6.5.1 would also cover this types of relations.	view correspondence = A named relation between two architectural views within an architectural description. View correspondences can for example be used to ensure consistency between views, or to specify relations between specific elements of different views that apply to the architecture described.	Issue for further study: view correspondences.

Commenting template (Version 3)

IEEE	NB-7	TH	4.17		I find the sentence '... used to establish consistency or similar relationships...' confusing and too specific. Not every type of relations between views is used only for consistency. I find the term 'view correspondence' artificial, why not simply used 'view relation'?	I suggest to replace 'view correspondence' with 'view relation' and for the remainder refer to my suggestions in remark on 4.17	Issue for further study: view correspondences.
IEEE	NB-8	TL	4.18		I find the definition of a viewpoint correspondence rule long and confusing. Why this focus primary focus on a contract in the definition?	Viewpoint correspondence rule = A template for a view correspondence that must be present between the actual views of the architectural description. This represents a kind of contract between two architectural viewpoints, enforced on their respective views.	Issue for further study: view correspondences. "template" may be the solution or totally avoided as too confusing.
IEEE	NB-9	TH	4.18		I find the use of the term 'rule' rather strange. I think it should be either (1) 'view correspondence rule', then you can consider it as a rule on the view correspondences; or (2) 'viewpoint correspondence', then it is a specification of the relations between viewpoints in general, and thus also between the instances of these viewpoints, the views.		Issue for further study: view correspondences.
IEEE	NB-10	TL	4.5		I don't like the use of the term module in the definition of an architectural model, since the term module is also used in other contexts in software.	architectural model = a part of an architectural view. An architectural model is typically a minimal consistent set of (software) elements and their relations.	AIP. We will eliminate "module". WD2 will flesh out some details about ARCHITECTURAL MODELS.
IEEE	NB-11	TH	4.9		I think that the definition of an architecture is not inline with the current practice. 'The fundamental organization of a system embodied in its components, their...'. The use of the word 'component' is confusing, and for example the SEI (in Software architecture in practice, second edition, Bass et al. 2003) and Rozanski and Woods (Software System Architecture) have banned it from their definitions since it has a too narrow meaning in software development and does not cover the use of modules, processes, states, or any other type of elements you would like to use in your architectural description.	I would suggest to use the term 'software elements' instead, leading to: "The fundamental organization of a system embodied in its (software) elements, their..."	Issue for further study: definition of ARCHITECTURE. Agree components is confusing. Is elements better?
IEEE	NB-12	TH	5		I was kind of disappointed not to find view correspondence and viewpoint correspondence rule in the 'core realm' of the conceptual model of architectural description. It should have a clear place in this context. From my experience, concretely putting it into the context of other concepts will raise new issues. For example, our proposition to add relations (and composition) in the conceptual model in the AARCH workshop "View Composition as a First-Class Concept in Architectural Description" also put relations between models as a central concept. The influence of correspondence on the architectural models is currently unclear.		AIP. These terms will be added to the conceptual model as the sections where they are introduced are further refined.
IEEE	NB-13	E	5		There is an inconsistency between figure 1, using the term 'system' and figure 2, using the term 'system of interest' for the same thing.	I suggest using the term 'system'.	Reject. The difference in the 2 figures is intentional.
IEEE	NB-14	TH	5		In a separate document in attachment (that I send to Rich Hilliard) I have explained my vision on views, viewpoints and models and why model types are so important in my vision.	Look to document "Boucke_On Views, Viewpoints and Models.doc"	TBD: post for WG42
IEEE	RdB-1	TL	5.1	(12)	I don't understand what it means that a system fulfills a mission "in its environment"? The 'environment' part seems to be the main feature that distinguishes 'mission' from 'concern'. But is the mission really 'in the environment'? Couldn't a mission be seen as just a (special type of?) concern? Both mission and concern are related to the stakeholders' interests c.q. the system's characteristics.	<< 1. Explanation/elaboration of the meaning of the phrase "in its environment". 2. Clarification of the relation between 'mission' and 'concern'. >>	Issue for further study: ENVIRONMENT.
IEEE	RdB-2	E	5.1	(12)	The use of 'its' is ambiguous, especially the last one ("its identified concerns").		
IEEE	DF-1	TH	5.1	(13)	In my opinion each system has one and only one architecture. Hence, I do not understand the sentence "In this sense, every system has an architecture, and perhaps more than one." Does it mean that every system has one architecture and it may change based on different mission/environment? In any case, please provide examples of different architectures of the same system.	You should add one or more examples of different architectures of the same system.	Issue for further study: relation of SYSTEM to ARCHITECTURE. Some philosophy to work out. Originally said every system has an architecture with possibly differing descriptions of same. The evidence is that when presented with a system, and asked to describe its architecture, different people may arrive at different results. In this sense, a system may have different architectures.

Commenting template (Version 3)

IEEE	DF-2	TL	5.1	(13)	The sentence "every system has an architecture" may be better explained by underlying that the architecture may be explicit (i.e. intentional) or implicit (i.e. accidental).	An architecture is the highest-level, i.e., fundamental conception of a system in its environment. In this sense, every system has an architecture and it can be explicit (i.e. intentional) or implicit (i.e. accidental). While the former is already described in the available documents, the latter, being the accidental results of design decisions, becomes useful in case for instance of maintenance operations of the related system and it can be partially recovered (i.e. documented) by analyzing other artifacts.	AIP. Intentional/explicit and accidental/implicit may be useful here.
IEEE	RdB-3	TH	5.1	(13)	I don't understand how a system can have more than one architecture. Sure, a system's architecture can be described using different views or even different architecture descriptions. But in my opinion those are different reflections of the same (conceptual) architecture. In this sense I agree with 5.1P16 ("This standard distinguishes..."), which I believe to be in conflict with the notion of more than one architecture from 5.1P13. Figure 2 and its explanation on page 16 also state that a system has 1 architecture.	An architecture is the highest-level, i.e., fundamental conception of a system in its environment. In this sense, every system has an architecture. Or An architecture is the highest-level, i.e., fundamental conception of a system in its environment. In this sense, every system has one, and only one, architecture.	Issue for further study: relation of SYSTEM to ARCHITECTURE.
IEEE	AT-5	E	5.1	(21)	About proxy stakeholders. If there is an evaluation team, there ought to be stakeholders who represent the QA team of the system. Therefore strictly speaking, the fact that they are external contract teams to evaluate the architecture does not preclude them from being legitimate stakeholders. As such, I would think that they are another type of stakeholders.		Issue for further study: proxy STAKEHOLDERS. Another type of stakeholder, or just another stakeholder?
IEEE	AT-6	E	5.1	(6)	Are systems stakeholders? I would suggest that even as they are part of the environment, they are not a stakeholder. They are a kind of context (from 4.10) that are dictated by stakeholders, who are people or a person.		Remove: we will retire this suggestion. Stakeholders are humans and collections of humans (organizations, groups, types).
IEEE	RdB-4	TH	5.1	(6)	I disagree with the notion of systems (from the environment) as stakeholders. I believe that it is counter-intuitive and incorrect. The major implication of system as stakeholder is that a system can have concerns. But a system A would never care about e.g. the security of system B, simply because a system doesn't really 'care' about anything. On the other hand, the "stakeholders" of system A do care about such issues (i.e. they can have concerns). Hence, I would say that there is a certain transitivity in being stakeholder in one system when that system uses/depends on another system. But in this transitivity, the systems are only the link and not stakeholders themselves.	The environment can include other systems that interact with the system of interest, either directly via interfaces or indirectly in other ways. Therefore, stakeholders of other systems in the environment may be stakeholders of the system of interest, too.	Remove.
IEEE	RdB-5	TL	5.1	Figure 1	5.1 Paragraph 12 defines a mission as "a use or operation for which a system is intended by one or more of its stakeholders...". This means that a mission 'belongs to' one or more stakeholders. Figure 1 does not show the relation between mission and stakeholder.	Draw connection between 'stakeholder' and 'mission'.	Issue for further study: MISSION.
IEEE	RdB-6	TL	5.1	Figure 1	5.1 Paragraph 12 says that "a system exists to fulfill one or more missions in its environment". However, figure 1 does not show the relation between mission and environment.	Draw connection between 'mission' and 'environment'.	Issue for further study: MISSION.
IEEE	RdB-7	TL	5.1	Figure 1	The figure shows a 1..* cardinality between system and architecture. However, a system has exactly 1 architecture (cf. Fig.2).	Change cardinality from has 1..* to has 1	That is the intent. Need to clarify for readers.
IEEE	AT-7	TH	5.1	Figure 1	The text version of the conceptual realm has the "Architecture is described by 0..* Architectural Descriptions" but this relationship is not depicted in Figure 1.	Either add "Architectural Description" to Figure 1 or merge Figure 1 and Figure 2	Update textual version. Need to clarify.
IEEE	AT-8	TH	5.1	Figure 1 and Figure 2	Apart from "Environment" and "Mission" and their relationships with "System", elements in Figure 1 are already in Figure 2, therefore Figure 1 does not provide a different level of concept to figure 2. On the other hand, it makes the readers wonder what the differences are.	I suggest that Figure 1 and 2 be merged together.	TBD: Need to also look at Matt Hettinger's proposal.

Commenting template (Version 3)

IEEE	RdB-8	TH	5.1	Figure 2	I find the use of the concept 'model' potentially confusing. On the one hand since it is not made clear in the figure what is being modelled. Is it a model of the system? Of the architecture? Of the environment? On the other hand, in SE the word 'model' has a certain connotation of the use of diagrams, whereas a viewpoint could also sanction the use of other representations such as budget spreadsheets (cf. 6.4.1) or, for instance, Tyree and Akerman's template to capture architectural design decisions. This is in line with 5.1(24) "Each model can use a different language, notation or model type.". I understand from A.3 that 'model' in this standard should not be taken as having something to do with diagrams, but rather as something which can be used to answer questions about 'Y'. However, the discussion in A.3 on 'architectural model' (which is what I take is meant by 'model' in Fig.2) does not make it very clear to me what this 'Y' is.	1. Change Fig.2 'model' to 'architectural model'. 2a. Make the subject (the 'Y') of 'architectural model' explicit in A.3 AND/OR 2b. use a more neutral word (without strong connotations), such as '(architectural?) representation' instead of 'architectural model'. Another option might be 'view module' instead of 'architectural model', since "Architectural models provide a mechanism to modularize architectural views." [6.4.1(2)].	5.1 is only talking about ARCHITECTURAL MODEL. Annex A is more discursive. Keep "model" for compatibility, but clarify its use in the standard along the lines suggested here.
ES	ES-1		5.1 and 6.7	EDITOR	By sure other systems are stakeholders, especially in complex systems of systems which are mutually dependent. Each system has requirements and assumptions relative to other systems from which it depends, and that should be architected and / or evolved in a coordinated manner. This is more than interface descriptions. For example, in architecting my system I may assume that some existing system will be upgraded to a newer version or that security issues will be addressed by an 3rd party application still not selected. Documenting dependencies are relevant in systems of systems where one system may not be aware which other systems are using their services, in order to assess the impact of modifications, or which systems are needed in order to any other is able to work.	I'd elaborate more in depth the inclusion in the Architectural Descriptions of the documentation of the assumptions and dependencies from other systems. I think that chapter 6.7 is very interesting for real, complex systems of systems (a multinational bank, for example, or a transforming project in which a technology is replaced by other in a big company) that shall be developed in a coordinated manner.	Assumptions and dependencies pertaining to other systems can be identified as CONCERNS and therefore modeled as necessary in the AD. Remove Systems as Stakeholders.
IEEE	AT-9	TH	5.2	(1)	There are other key stakeholders such as designers, testers, integrators who are very likely to use the AD.	Amend text to "These stakeholders include clients, users, the architect, designers developers, testers, integrators and evaluators."	AIP these as additional EXAMPLES of stakeholders.
ES	ES-2		5.2	(3)	WD said: The architect creates and maintains an architecture for a system for the acquirer. Comment: This is ambiguous. The architecture is for the system and, in general, what is for the acquirer is the system itself, not the architecture. Of course, depending on the kind of relationship between both, the architectural description may be an item delivered to the acquirer with the system as part of the contract or agreement. WD Said: The architect may work from requirements provided by an acquirer or may be responsible for eliciting and developing requirements as part of the architecture development. Comments: Also ambiguous. My experience is that the Acquirers / users are much more worried in business requirements than in architectural requirements. I mean, the Business is not interested too much in the systems architecture provided it solves their business needs now and will be able to keep solving them in the foreseeable future according to the business plans. It is left to the CIO and the IT Area to translate those business needs into the enabling architecture	The architect is responsible to translate acquirer requirements into enabling architectures . Acquirer requirements are any kind of requirements from business to technical. I might see business architects eliciting business requirements, but I do not see (in general) software architects eliciting technical requirements with the acquirer, or discussing with them the best architectural solution for the data models and data repositories. The paragraph could be expanded to better address the requirements issues.	Issue for further study: context of Architecture Description
IEEE	AT-10	TH	5.2	(5)	Architecture also plays a key role in system integration.	Amend text to "... steer the downstream development, construction, integration, deployment, operation and evolution of the system."	AIP
IEEE	NB-15	TH	5.3	(4)	The standard states that "it neither assumes nor prescribes a specific life cycle model". While I agree with this statement, I believe that the standard does impose some requirements to what life cycle models can be used in combination with the standard. For example, the life cycle must incorporate at least some architectural phase. Please make these requirements explicit.		Need to clarify intent: The Standard says what to produce if you describe an architecture. It doesn't say, when, how or even if you do so. In that respect it is neutral with respect to life-cycle models that would say when/where/how you produce architectures or ADs.

IEEE	NB-16	E	5.3	(5)	The previous paragraph ends with 'the scenarios below...'. For me it was confusing that the paragraph below this sentence did not describe the scenario's. I suggest reordering the text.	This standard neither assumes nor prescribes a specific life cycle model—a life cycle model is to be separately chosen by users of the standard. System architecture potentially impacts all processes within the system life cycle. Examples of these impacts are shown in Table 1 [ISO/IEC 15288, Figure 4]. The scenarios below are intended to suggest the range of uses of the standard within a system life cycle.	
ES	ES-3	T	5.3	(3)	System concept' is ambiguous term.	Add 'system concept' to the glossary	AIP: this will be rewritten.
ES	ES-4	T	5.3.1	(2)	WD sais: the architectural description will typically evolve throughout the life cycle. ¿because it is refined, or because the architecture changes, or both? Anyway, wat may be used to assess changes to the system is the AD, not the AD evolution.	Explain the reason why the AD evolve. Modify explanation as in 5.3.2	AIP
ES	ES-5	T	5.3.2		It is confusing use the word develop both to the architecture and to the system. The expression: The architecture development will normally be carried out by the developer as part of the overall activity of developing a sequence of systems. It is not clear if the developer is an SW developer or an architect	Use expressions as create or evolve architectures instead of develop architectures. Or use architect instead of developer when referig to the person who creates the arqitecture, and use developer for the person who build the systems according to the pre-defined architecture.	Issue for further study: life cycle terminology.
IEEE	RdB-9	TL	5.3.4	(1)	The prime purpose of evaluation is to determine the quality of the "architecture" described in the architectural description. To determine the quality of the architectural description itself would imply evaluation of the document only, not of the architecture described in it.	The purpose of evaluation is to determine the quality of an architectural description and the architecture it describes, and to predict the quality of systems whose architectures conform to the architectural description.	Issue for further study: context of Architecture Description
ES	ES-6	T	5.3.4		It is not possible to assess the quality of system from bad ADs	I'd add that prior to be able to predict the quality of systems resulting from the architectural description, the quality of the AD should have been assessed OK	AIP: but evaluation may be beyond scope of the Standard and addressed as a future work item for WG42.
ES	ES-7	T	5.3.x		¿aren't these the same as reference architectures, such as the OSI layers o TCP/IP? They do not exist by themselves but you should conform to them.	Anyway, ¿how to document a reference architecture? Maybe this is a new scenario?	AIP: want to mention reference architectures, but this is not a new scenario.
IEEE	AT-11	TH	5.3.X		New scenario: architecture integration. Often in integration between large systems, architects start off by examining the architectural documents at a high-level to understand the structure and design of the system to check if there are any potential issues.	This scenario occurs when more than one systems are to be integrated. The standard provides guidance to facilitate systems integration through the architectural descriptions. Such information can include the various architectural views, their concerns, interfaces, components and quality attributes. An architectural description provides a high-level description as a starting point for investigating into the viability of system integration.	AIP
ES	ES-8	T	5.3.xx		Missing scenario?: intermediate architecturing are nedded to transform / replace a very complex system with a known final architecture. The difference with the 5.3.2 scenario is that you know from the begining the final system, but due to its complexity the evolution shall be made step by step, each one having an well defined intermediate architecture. So you have to document the final as well as the intermediate architectures as part of the architectural description of the final system	Describe the new scenario.	Issue for further study: context of Architecture Description. This list is not intended to cover all possible scenarios.
IEEE	RdB-10	E	5.4		The list a)-m) contains a mix of emphasis: one type of use seems to put emphasis on the task that one can perform based on an architectural description (e.g. analysis [a] or planning [b]) without specifying the role of the architectural description, whereas another type of use appears to emphasize the role the architectural description has in a particular task (e.g. input [g,h] or specification [m]). When reading the list top-down, one has to adapt to the change in emphasis (cf. k 'preparation OF acquisition documents' vs. m 'specification FOR a group of systems')	Use consistent emphasis on task or role.	AIP: good observation
IEEE	AT-12	TL	5.4.X		New use of architectural descriptions. This use is for sharing all or part of the architectural descriptions with potential buyers as well as where system integration is required.	Communicate the characteristics, features and design of a system to potential acquirers and integrators	AIP
IEEE	AT-13	TH	6.2	(6)	The bullet points that list the generic architectural concerns should be according with the definition provided in 4.16. If the suugestion made in point 5 of this document is accepted, i.e. architectural concerns should include business and technical concerns, then there should be two extra bullet points that illustrate that.	Add text: (A) "- The business requirements of the system" and (B) "The technical requirements of the system, e.g. performance, security and reliability". If technical requirements is a generic classification, then one may consider removing bullet point "Maintainability, deployability and evolvability" and quote them as examples of this generic class.	AIP

Commenting template (Version 3)

IEEE	AT-14	TL	6.4	(a)	Split point a into two points; the identifier of a view should be unique within the AD, and the other introductory description should be treated like comments on the view	Amend to "a) a view identifier" and "b) other introductory information and comments, as defined by the using organization"	<p>AIP</p> <p>42010 is about multiple views of one system, not multiple systems</p> <p>AIP: good example.</p> <p>AIP: and models should be related to concerns.</p> <p>Issue for further study: aspects and perspectives.</p> <p>Issue for further study: view consistency. Examples are a good idea.</p> <p>Issue for further study: rationale and decision capture.</p> <p>Issue for further study: rationale and decision capture.</p> <p>Issue for further study: rationale and decision capture.</p>
IEEE	AT-15	E	6.4	(b)	spelling	Amend to "...one or more architectural models..."	
IEEE	NB-17	TH	6.4	Note 1	I disagree with the content of this note 1. Being able to describe the system on different levels of abstraction is one of the cornerstones of architecting. If the conceptual model does not support describing an architecture, and later on zooming in on one of the components and describe its architecture in the same architectural description, what is the use of such model for complex systems? It is artificial and a kind of schizophrenic to think about it in this way.	I suggest to remove this note and to adapt the conceptual model to be useful in such situation, this would be possible if you would follow a more flexible interpretation of views and models as proposition in remark 14.	
IEEE	AT-16	TL	6.4.1	(3)	One may consider using an architectural example instead of a project management example, which is more relevant.	An example can be a reliability viewpoint framing concerns such as software reliability, hardware reliability and recovery process etc.	
IEEE	AT-17	TL	6.4.1	(5)	As a standard, we should identify the necessary attributes for architectural models	a) model name b) model notation type c) purpose of model	
IEEE	NB-18	TH	6.4.1	Note	I like the fact that the standard sees sharing models as an aspect oriented style of architectural description, but I disagree that sharing models allows to capture architectural perspectives. Firstly, a perspective is a set of activities and by this can not be 'captured in architectural models'. Secondly, I seriously doubt that you can easily capture the changes to the architectural description in separate models that you can share amongst architectural views. thirdly, on of the essential properties of aspect orientation is how the different elements are composed with each other. My suggestion is to the statements made in this note into a better perspective, as the current sentence suggests that this is definitely possible while I would say it only an untried idea.		
IEEE	AT-18	TH	6.5	(1)	(a) How shall AD record known inconsistencies? Does the standard assume that the architects would find them and then record them? Wouldn't the architects fix the consistencies as they discover them and if so, under what circumstances would they remain in the AD? (b) Similar to clauses 6.1 to 6.4, a list of attributes should be specified.	Perhaps some examples about viewpoint inconsistencies would help resolve this issue.	
IEEE	NB-19	TH	6.6	Figure 3	Why does a decision affects only a model? As rationale is given for architecture, architectural description and model, it would be far more logical if a decision affects these too. Secondly, it is rather confusing that the concept 'model' is mentioned twice in the figure.		
IEEE	RdB-11	TL	6.6	Figure 3	The figure shows how a decision affects 0..n models. This seems to be a shortcut: the decision affects the architecture (cf. 6.6P3, 6.6P4). The result of the decision may of course also affect 0..n models in an architectural description, but only indirectly. If a model would be affected by a decision but the architecture not, the model would no longer be a true representation of the architecture. Note also that 'Model' appears twice in Figure 3	Remove 'model' linked to 'decision' and draw the 'affects' relation from decision to architecture.	
IEEE	AT-19	TH	6.6	Figure 3	This figure shows that rationale comprises of architecture rationale, AD rationale and model rationale. However, the text describes viewpoint rationale and view rationale. So there are some inconsistencies between the text and Figure 3. In Figure 2, Rationale is associated with Architecture Description and AD contains views and models. Therefore, Rationale has associations with views and models already. The question is whether it is necessary to further classify rationale into sub-classes. I think that this is an overkill because the main purpose of the rationale is to explain the design. Reasoning for selecting a viewpoint is to justify the architectural design process, and not the architecture design itself.	I suggest to remove Architecture Rationale (Viewpoint Rationale), AD Rationale and the Levels of Rationale. I suggest to have one level of rationale, i.e. design rationale for the architectural model.	

IEEE	RdB-12	TL	7.1	(17)	I don't necessarily think this is too strong. Given the previous bullet in the list (AD concerns include AF concerns) it seems impossible to leave out a generic stakeholder, since that would leave out the concerns held by that stakeholder as well. However, and this is probably highly hypothetical, what happens if an AF defines more generic stakeholders than the number of identified stakeholders in AD? I.e. there are not 'enough' identified stakeholders. Can one stakeholder belong to multiple generic stakeholder 'types'? (this last question may be the way be less hypothetical; even if there are 'enough' identified stakeholders to have instantiations for each of the AF generic stakeholders, there may be stakeholders that act in multiple roles).		Issue for further study: architecture frameworks.
IEEE	AT-24	TH	7.1	(9)	In the paragraph starting "An architectural description, AD, conforms....", all elements in the core model are included in the metamodel except architectural rationale, is there any reason for such omission? Similarly, architecture rationale is missing from Figure 4.		Fix.
IEEE	AT-25	E	7.1	Figure 4	This figure is not referenced anywhere in section 7. I assume that it is the meta-model of of an architectural framework.		Fix.
IEEE	AT-26	TH	7.1	Figure 4	If "Viewpoint Correspondence Rules" are to define the consistency of the Viewpoints (see Clause 6.5), then shouldn't this entity in Figure 4 be associated with entity Viewpoint in some ways (as described in the last bullet point of AF definition - " the definition of zero or more viewpoint correspondence rules over those viewpoints")?		Fix.
IEEE	AT-27	TL	A.1	(3)	There are some ambiguity in the statement "separating their treatment by the resultant view from that of other concerns using other viewpoints and thereby allowing interested stakeholders to focus on a few things at a time" . What is a resultant view? What is the actual relationship between a concern and a viewpoint? From the reading of the standard, I gather that (a) both viewpoint and concerns are at M1 meta level, and they need to be instantiated; (b) a concern can belong to one or more viewpoints and one viewpoint can frame multiple concerns. So if a stakeholder wants to view a separation of concerns, the stakeholders can select a viewpoint and find all the instantiated concerns that belong to the viewpoint.	May amend to "... architectural viewpoint frames one or more types of concerns, allowing interested stakeholders to view the architectural design based on those concerns, thereby filtering irrelevant information."	AIP: fix terminology (e.g., "resultant view"). But I don't think we will add structure over concerns such at concern types. This is left to methods.
IEEE	RdB-13	TL	A.1	(3)	This paragraph seems to imply that the treatment of a particular concern is limited to a single view. I'm not sure whether that's what's really meant.		Fix. That is not the intent.
IEEE	AT-28	TL	A.3	(7)	About the definition of 3 kinds of models, especially architectural description as a model. The explanation offered here seems to be inconsistent with the definition in clause 4.3 and Figure 2. If architectural description is a collection of information products used to document the architecture, then it is a super-set that contains architectural view and architectural model. Therefore, it should answers questions more than just stakeholders and concerns related.	Perhaps the three models should be architectural view, architectural model and architectural rationale model (as in Figure 3). Architectural description is a "collection" of all these 3 models instead of an independent model in its own right.	Issue for further study: MODELS.
IEEE	RdB-14	TH	A.3	All	"The definition tells us every model has a subject" (model can be used to answer questions about <subject>). However, the subjects of the "concrete models" 'architectural view' and 'architectural model' are not immediately obvious from the stated purpose. For 'architectural view' I see how the purpose can be rephrased to "to answer from a specified viewpoint one or more specific stakeholders' questions about those stakeholders' identified concerns for the system of interest" (cf. the purpose for 'architectural description'). However, I'm unsure about how to rephrase the purpose for 'architectural model'.	Rephrase purpose for 'architectural view' and 'architectural model' in terms of 'can be used to answer questions about...'. (cf. 'architectural description' and definition of model).	AIP
IEEE	AT-29	E	A.3	last		Amend text "at least viewpoint" to "at least one viewpoint"	
IEEE	NB-20	G	Annexes		I noticed that the concepts 'view correspondences' and 'architectural framework' are lacking in all annexes. I assume that this will be updated in upcoming version of the standard.		Accept

Commenting template (Version 3)

IEEE	AT-30	TL	B		This should be a summary list from section 6 of the elements from those clauses. On the subject of describing an architecture model, one may use modelling language such as UML as an example.		<p>Issue for further study: examples.</p> <p>Issue for further study: examples.</p> <p>Accepted a number of changes to clause 1. Open issue: architecture v. architectural. One easy fix: architecture-related concerns. Issue for further study: Enterprise Architecture (EA). These comments raise a number of issues from the perspective of Enterprise Architecture (EA). WG42 should have a discussion in regard to the extent of revision in support of EA use, and take these comments under direct consideration.</p>
IEEE	AT-31	TL	C		It would be more representative to use viewpoints, or a combination of viewpoints, from widely accepted architectural frameworks such as RM-ODP and TOGAF in this section. Aligning viewpoints from these frameworks to the standard can help the readers on conformance.		
SE	JBZ				Johan Bendz' comments in a marked up Word file	See file: ISO-IEC-WD0a-42010_jbz3.doc	
IEEE	SC				Soumen Chatterjee comments in a commented PDF	See file: ISO-IEC-WD0a-42010_Soumen Chatterjee_Review comments.pdf	