

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Standardized product ontology register and transfer by spreadsheets –
Part 5: Interface for activity description**

**Enregistrement d'ontologie de produits normalisés et transfert par tableurs –
Partie 5: Interface pour la description des activités**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2017 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient 20 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

65 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Standardized product ontology register and transfer by spreadsheets –
Part 5: Interface for activity description**

**Enregistrement d'ontologie de produits normalisés et transfert par tableurs –
Partie 5: Interface pour la description des activités**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 01.040.01; 01.110

ISBN 978-2-8322-4369-5

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	4
1 Scope.....	6
2 Normative references	7
3 Terms, definitions and abbreviations	7
3.1 Terms and definitions.....	7
3.2 Abbreviations.....	9
4 Overview	9
4.1 Activity described as an ontology	9
4.2 Use cases and key technical concepts	10
4.3 Relation among properties of different activities.....	14
4.4 International Concept Identifier (ICID).....	14
5 Basic structure of the PAM	14
5.1 Activity and arrows.....	14
5.2 Subactivities	15
5.2.1 General	15
5.2.2 Specialized activity	15
5.2.3 Component activity	15
5.3 ICOM representation.....	16
5.4 Role of the mechanism (M) in the PAM	16
5.5 External function call	17
5.6 Basic PAM notation with function symbols	17
5.7 Joining arrows	20
5.8 Forking arrows.....	21
5.9 Branching or joining of arrows.....	21
5.10 Transcendental arrows.....	22
5.10.1 General	22
5.10.2 Modelling incoming arrows	24
5.10.3 Modelling outgoing arrows	24
5.10.4 Modelling connections of arrows at frame boundary.....	25
5.10.5 Contracted form of representation for branching and joining arrows.....	26
5.10.6 Domain or codomain overloading for transcendent arrows	27
5.11 Extended semantics beyond IDEF0.....	28
5.11.1 Specialized types of activity and its icon	28
5.11.2 Conjunction node.....	30
5.11.3 Disjunction node.....	30
5.11.4 Complementation node	31
5.11.5 Selection node.....	31
5.11.6 Transformation node.....	31
5.11.7 Decision tree	31
5.12 Graphic properties of arrows.....	31
5.13 Arrow specialization.....	31
5.14 Delegated formula interpretation	32
Annex A (normative) Meta-properties for activity description	34
A.1 General.....	34
A.2 List of meta-properties	34
Annex B (informative) Description examples for the PAM.....	37

B.1	Design product.....	37
B.2	Sample IDEF0 Diagram	48
Annex C (informative)	Example PAM data for production operations management.....	51
Bibliography.....		59
Figure 1	– See fine arts at Museum	11
Figure 2	– Production operations management (extracted from IEC 62264-3).....	12
Figure 3	– Production operations management modelled in PAM and depicted as IDEF-0 diagram	13
Figure 4	– Basic activity and its subcomponents	16
Figure 5	– Corresponding IDEF0 diagram for basic PAM notation	18
Figure 6	– Sample activity drawing in IDEF0 and ICOM	19
Figure 7	– Subactivities and arrows	19
Figure 8	– Joining arrow example	20
Figure 9	– Forking arrow example.....	22
Figure 10	– Transcendental arrows to be taken over by child nodes	23
Figure 11	– Transcendental arrows from the parent node	23
Figure 12	– IDEF0 extension for specialized activity node in the PAM.....	29
Figure 13	– An implementation example of Conjunction node in the PAM	30
Figure 14	– Super relation and its application for specialized activity	32
Figure B.1	– Class meta-class example of the PAM for “design product” activity	38
Figure B.2	– Property meta-class example of the PAM for “design product” activity	40
Figure B.3	– Relation meta-class example of the PAM for “design product” activity	42
Figure B.4	– IDEF0 diagram image corresponding to A-0 (frame containing A0)	49
Figure B.5	– IDEF0 diagram image corresponding to A0 (frame containing subactivities of A0)	50
Figure C.1	– Class meta-class example for production operations management defined in IEC 62264-3.....	52
Figure C.2	– Property meta-class example for production operations management defined in IEC 62264-3	53
Figure C.3	– Relation meta-class example for production operations management defined in IEC 62264-3	54
Figure C.4	– Autogenerated IDEF 0 A-0 (top) node for production operations management defined in IEC 62264-3	57
Figure C.5	– Autogenerated IDEF A0 node for production operations management defined in IEC 62264-3	58
Table 1	– Basic PAM notation for arrows	18
Table 2	– Extracts of relation meta-class definitions for activities.....	26
Table 3	– Contracted representation for connectivity of activities	28
Table 4	– Reserved keywords for formula interpretation.....	33
Table A.1	– Meta-properties of relation meta-class used for activity description	35

INTERNATIONAL ELECTROTECHNICAL COMMISSION

STANDARDIZED PRODUCT ONTOLOGY REGISTER AND TRANSFER BY SPREADSHEETS –

Part 5: Interface for activity description

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62656-5 has been prepared by subcommittee 3D: Product properties and classes and their identification, of IEC technical committee 3: Information structures and elements, identification and marking principles, documentation and graphical symbols.

The text of this International Standard is based on the following documents:

CDV	Report on voting
3D/257/CDV	3D/287/RVC

Full information on the voting for the approval of this International Standard can be found in the report on voting indicated in the above table.

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.  **Documents**
Document Sharing Hub

A list of all parts in the IEC 62656 series, published under the general title *Standardized product ontology register and transfer by spreadsheets*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC website under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

STANDARDIZED PRODUCT ONTOLOGY REGISTER AND TRANSFER BY SPREADSHEETS –

Part 5: Interface for activity description

1 Scope

This part of IEC 62656 specifies a method for representing activities and relations among the activities by a tabular ontology representation, called “parcellized activity model” or PAM for short, which is a specialized use of a generic tabular ontology data model, known as the parcellized ontology model (POM) defined in Part 1 of the IEC 62656 series. The activities that can be described by this document include part or whole of an enterprise, an organization or a collection of services, a set of events or processes which interact with each other by exchanging physical or non-physical entities. This part of IEC 62656 also defines a method for uniquely identifying activities, or their homologues happenings in a certain sequence. In addition, this document identifies flows of information, objects or materials exchanged among activities, where each of the activities is represented by a class and each flow by a relation.

Consequently, this document enables characterization, classification, and identification of a set of activities as part of a normalized ontology. And this enables registering of a pattern of activities as a set of metadata and uploading it onto the IEC 61360 Common Data Dictionary (CDD), maintained as an online database of the electrotechnical concepts.

Additionally, this part of IEC 62656 provides a method to integrate ontologies of products and activities including services, in a single model. This means a product can be analyzed in its operational context for service. Such an integrated view will help people of different technical backgrounds to see and share knowledge about the extent of an enterprise that requires the products and services as indispensable resources. Such a data representation will also help analyse the key functionalities of an enterprise and its available resources, with clear definitions, limitations and interactions among them, when people are required to respond or react to a new external condition or situation in a short time frame, in particular, at an emergency or natural hazard.

Meanwhile, this part of IEC 62656 does not intend to provide a detailed algorithmic description of a flow of information, timing chart of processes, or sequential ordering of events that will be necessary in a software design or programming phase of an information system that handles activities or events. These detailed specifications of the algorithms and associated construction of the data structures are left to the realm of software engineering methodology and tools where there are so many schools and styles already, such as UML (Unified Modelling Language), BPMN, SysML, DFD, IDEF, and other CASE (Computer Aided Software Engineering) tools.

This International Standard neither intends to standardize nor introduce a new method of graphic description for activities or processes. Ideally, an ontology of activities modelled by this International Standard must be expressible by a number of existing graphical presentation tools and process description languages for activities.

Nevertheless, some graphical presentations in the style of such tools or languages are helpful for making the people understand the content of the PAM, and therefore, they are used in this International Standard. In most of the cases, IDEF-0 is preferred for the purpose, because it describes both activities and flows of things among the activities, but any other choices of tools or languages can be made, wherever they are appropriate and relevant.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61360-2:2012, *Standard data element types with associated classification scheme for electric components - Part 2: EXPRESS dictionary schema*

IEC 61360-4, *Standard data element types with associated classification scheme for electric components – Part 4: IEC reference collection of standard data element types and component classes* (available at <http://cdd.iec.ch/>)

IEC 62656-1, *Standardized product ontology register and transfer by spreadsheets – Part 1: Logical structure for data parcels*

IEC 62264-3:2007, *Enterprise control system integration – Part 3: Activity models of manufacturing operations management*

ISO 13584-24, *Industrial automation systems and integration – Parts library – Part 24: Logical resource: Logical model of supplier library*

ISO 13584-42:2010, *Industrial automation systems and integration – Parts library – Part 42: Description methodology: Methodology for structuring part families*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purpose of this document, the terms and definitions given in ISO 13584-24 and IEC 62656-1, as well as the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1.1 activity

organizational, logical or conceptual unit for performing a set of specific actions or functionalities

Note 1 to entry: An activity can be just for performing one action or functionality, and in an extreme case, for zero action or functionality, meaning the activity is just an endpoint for terminating activities.

Note 2 to entry: An activity is not necessarily a process in time sequence in the PAM. Two or more activities may concurrently work and interact with each other.

3.1.2 arrow

mapping from one category of things to another, yielding an information flow, a movement of physical items, a change of states from one state to another, or a directional correspondence from one collection of things to another, which is embodied as a functional relation

Note 1 to entry: Arrow as an information construct in this part of IEC 62656 embodies an mathematical entity named "arrow" originating in the category theory of mathematics, which is synonymous with function, but maps elements of one collection specified as "domain" to another collection specified as "codomain", with a strong sense of direction.

Note 2 to entry: Arrows can also be used in a formula, such as in $F : X_1 \times X_2 \times \dots \times X_n \rightarrow X_{n+1}$ where an

arrow exists as the mapping F from an n -ary set $X_1 \times X_2 \times \dots \times X_n$ to X_{n+1} .

3.1.3

arrow overloading

specialization of an arrow by narrowing or detailing either one of or both of the domain and codomain of the arrow, being considered as a function

Note 1 to entry: Overloading of an arrow typically takes place on the frame boundary of a diagram in a lower node of an activity.

3.1.4

aspect

way things appear, are looked at or expressed as an association among properties, classes, or ontological elements in general, by the use of a relation

Note 1 to entry: A property may belong to one or several aspects.

3.1.5

branching point

forking point

point from which an arrow forks out into two or more arrows

3.1.6

connection point

point at which arrows or lines fork out, or at which several arrows or lines meet each other

Note 1 to entry: Both branching point and junction points are included in the connection points.

3.1.7

IDEF0

Integration Definition for Function Modelling

graphical language to model decisions, actions, and activities of an organization or system, defined in the IDEF series of data modelling methods

Note 1 to entry: IDEF0 was once adopted as Federal Information Processing Standard (FIPS) 183 by ANSI (American National Standards Institute). However, it was withdrawn from FIPS in 2012, for it is no longer necessary to designate only one method among many as the graphical language for modelling activities.

Note 2 to entry: This note applies to the French language only.

3.1.8

ontology

shared and formal modelling of knowledge about a domain

3.1.9

parcellized ontology model

POM

formal specification of being of things as a concept, including state of things and configuration of things, by their properties and relations, formed in a set of relational tables each of which represents a category of the ontological entities

Note 1 to entry: Parcellized ontology model is formally specified in IEC 62656-1.

Note 2 to entry: Examples of the categories of the ontological entities are class, property, relation, datatype, unit of measurement, etc., which are essential constructs for the description of being.

Note 3 to entry: This note applies to the French language only.

3.1.10**parcellized activity model****PAM**

specialized use of the parcellized ontology model defined in IEC 62656-1 for representing an activity as part of an ontology

Note 1 to entry: This note applies to the French language only.

3.1.11**service**

activity performed or being performed for the purpose of someone or something

Note 1 to entry: Usually an activity is performed else than for oneself or for itself.

3.1.12**subactivity**

subcomponent of an activity

Note 1 to entry: One activity can be broken down into several subactivities. They can be a set of diachronical divisions of an activity, as well as a set of synchronical divisions of an activity. For example, a collaborative work done by different task forces of an enterprise.

3.1.13**transcendent arrow**

arrow that enters into an activity and interacts with some or all of its subactivities

Note 1 to entry: A transcendent arrow may be forking into multiple arrows or joining another in a lower node frame.

3.2 Abbreviations

For the purposes of this document, the following abbreviations apply.

BPML	Business Process Modelling Language
BPEL	Business Process Execution Language
BPMN	Business Process Modelling Notation
CDD	Common Data Dictionary
DFD	Data Flow Diagram
KPI	Key Performance Indicator
POM	Parcellized Ontology Model
SysML	Systems Modelling Language
UML	Unified Modelling Language

4 Overview**4.1 Activity described as an ontology**

According to an acknowledged web resource, ontology is an explicit and formal specification of conceptualization in a domain. If ontology is such, then an activity should be definitely a part of the ontology. For an activity usually has an objective, ways and means to achieve the objective, person or organization to be engaged, time, financial constraints and requirements to be observed. In addition, an activity has a relationship to other activities, having them as prior or posterior activities with which it may exchange information, goods or material. Such an activity may have child or subactivities that assume part of the role or objectives of the main activity. Moreover, some patterns of activities are repeatedly performed, so they can be considered as a model and its instances. Furthermore, an activity is often a collaborative process conducted in plural languages by different peoples of different countries on different continents, with a plethora of cultural backgrounds and traditions. In consequence, there are ample needs and necessities to share understanding about activities in an internationally

normalized manner with their identification, canonical properties and concepts being expressed in a machine sensible way.

The parcellized activity model (PAM) is a specialized use of the parcellized ontology model (POM), defined in IEC 62656-1, for the purpose of modelling an activity or a set of activities and possible interactions among them, with the data structure provided by the latter standard. In the PAM, activities are expressed by classes whilst the flows of information, goods and materials among activities are expressed by functional relations, with which not just information flow but also the flow of products, materials and personnel can be included into the scope of the model. This contrasts clearly with some other activity models that allow only the information to be exchanged among the activities. However, the PAM is not intended as a language for graphical presentation of the activities; for such a purpose there are already BPML, BPEL, BPNM, IEDF0 and DFD, etc. Neither, the model is intended as a means to write a flow chart or algorithmic diagram for designing a specific software application. Rather it is designed to share a common understanding about the characteristics of activities and the relations among them as a set of metadata, in other words, classes and properties of an ontology expressible in a standard format of ontology, which allows text based queries. Yet the metadata are also expected to be in a way translatable and graphically reproducible in at least one of the graphical presentation languages of activities, or in an appropriate combination of them. Here, the language we selected for primary translation is IDEF0. But the PAM is not a data model exclusive to the language. Rather, it is designed as a means to coordinate ontologies of products, services and activities together. Thus it is a prerequisite that the description by the PAM must be well coordinated with the ontologies of products and materials that have been stored and maintained in the common metadata registry of IEC, i.e., the IEC 61360-4 DB, or so called “IEC CDD” or sometimes more simply, “CDD”(Common Data Dictionary), for which the Parcel series of standards provides input and output interfaces and its model extension.

4.2 Use cases and key technical concepts

The principal applications of this document might be in the field of engineering. However, importance of guiding or assisting people in visiting places by handy electrotechnic means is not limited to the engineering, but well recognized for a wide range of industrial activities, including process automation, agriculture, fishery, hazard prevention, jurisdiction, defence, transportation, medicine, nursery and tourism. Whilst a pattern of specialized engineering activities is comprehensible only within a limited number of experts of the domain, a holiday visit to an art museum can be understood by much a larger audience or readers. Thus, for illustration purpose, an activity to “see fine arts at museum” is picked up for the first example in 4.3 as an exemplary use case, because most of the readers must have visited some museum more than once to see fine arts in their lifetime. And we expect they must have had a similar experience, seeing different genres of arts at different locations where local people speak different languages. En suite, we deliver a use case that lies in the midst of the process automation, i.e., an ontological description of a segment of the activities of enterprise-control system integration, described in IEC 62264-3, which latter is often touted as a cornerstone of the smart manufacturing.

Figure 1 illustrates how a touristic activity to see fine arts at a museum can be modelled as an ontology. Under a generic activity named “See fine arts at museum”, there are two types of hierarchy; one is a “specialization tree” or so called “is-a” tree that has a small open triangle at the head of the relation line in the diagram, another is a “composition tree” or “part-of” tree that shows how an activity can be decomposed into sub-parts, which is represented by a relation line with a black diamond at the head. Of course, properties that are held by a generic activity shall be inherited into specialized activities in the case of specialization hierarchy. In the given case, the activity to “see fine arts at museum” may have, for example, “language(s) for arts description”, “admission fee”, “open hours”, and “genres of arts” and they are uniformly inherited into specialized activities, such as an activity to “see fine arts at American museum”, to “see fine arts at French museum”, to “see fine arts at Japanese museum”, and to “see fine arts at Spanish museum” in the same way. Ultimately at all leaf nodes, be it an activity to see arts at a globally renowned big French museums like Musée du Louvre or Musée d'Orsay, or to see some select arts at a Japanese city local museum of a reasonable scale, like the ones at Kagoshima city museum of arts, the properties are inherited from the

generic activity to “see fine arts at Museum” in the same manner. Only the parametric values of them are different. In addition to those inherited properties, there shall be some native properties that distinguish one activity from another. In exception of those properties, basic components of the activities of the generic activity, depicted as yellow boxes in Figure 1, are the same as those specialized activities at leaf nodes; for basically at every museum, you have to get a ticket or pass, then get admitted into an exhibition, visit some art halls in series, then you may exit from the exhibition at one of the art halls. Note that an arrow means people spread among activity boxes (marked in yellow) move from one activity at the source of an arrow to another at the destination of the arrow, rather than a description of a sequence of actions for one person.

Figure 2 shows a typical example of a set of standardized industrial activities extracted from IEC 62264-3 (Figure 14 in the original) that is expressed by means of DFD. Figure 3 is a figure in IDEF0 which corresponds to Figure 2, and in line with the PAM specification according to IEC 62656-5. According to IEC 62264-3, all production activities in manufacturing can be patterned after the model depicted in Figure 2, and each specialized manufacturing should be adequately explained by a specialization of the diagram. Here, the readers should not misconstrue the objective of this use case; it is not intended to claim that the content of IEC 62264-3 is replaceable by a functionality of the PAM. Rather it is simply to state that significant part of the semantic content of IEC 62264-3 can be translated and reproduced in a form of ontology data and can be stored and maintained in IEC CDD by means of the PAM, which enables interaction and cross references with the related product definitions stored in the CDD. Incidentally, the readers are reminded that the internal communication flows, such as operational commands, depicted in Figure 2 between the current level (level 3) of enterprise-control and the level 1 and level 2 functions according to the respective definitions in IEC 62264-3 are removed from Figure 3 because they shall be elaborated when the “Production execution management” indexed as A5 in Figure 3 is unfolded into subactivities in the PAM, as a base node for unfolding. Since the PAM itself is not a graphic presentation language for activities, the figures and diagrams are the work created by tools and applications that interpret the ontology data according to IEC 62656-5. Nevertheless, In Annex C, a set of sample data according to IEC 62656-5 are included (Figure C.1 through Figure C.), which semantically correspond to Figure 3. In addition, some actual examples of auto-generated figures from the PAM are shown in Annex C (Figure C.4 and Figure C.5).

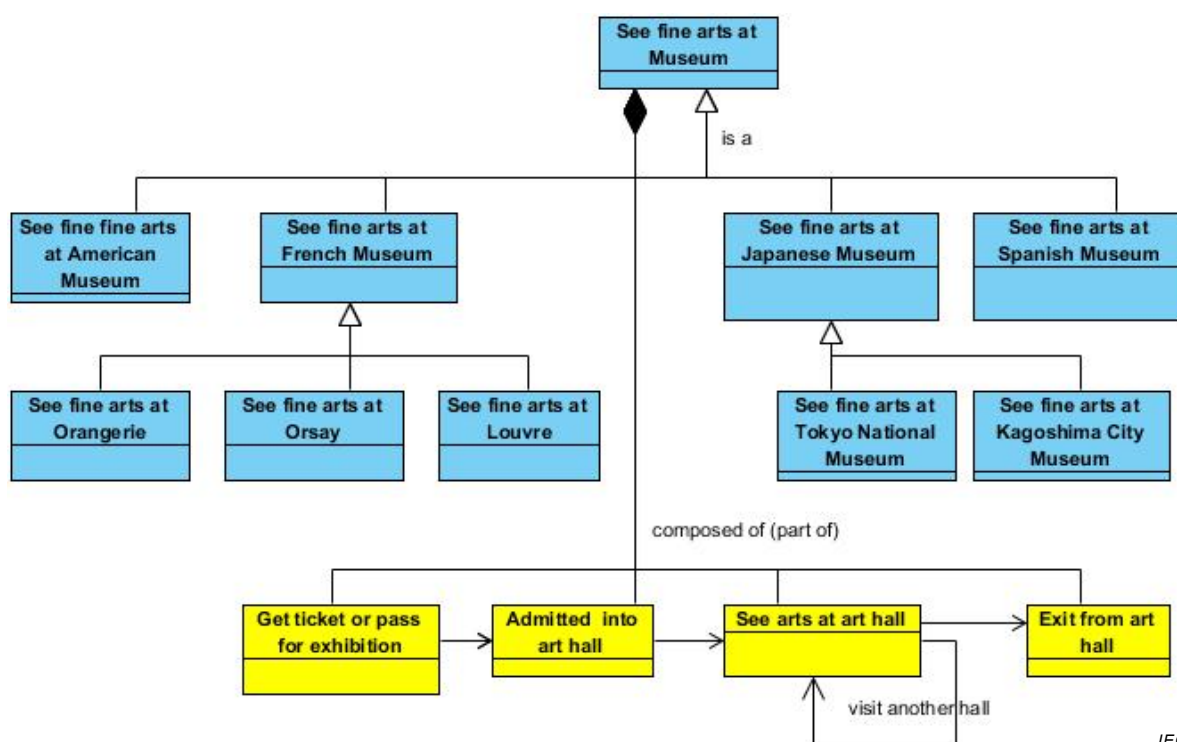
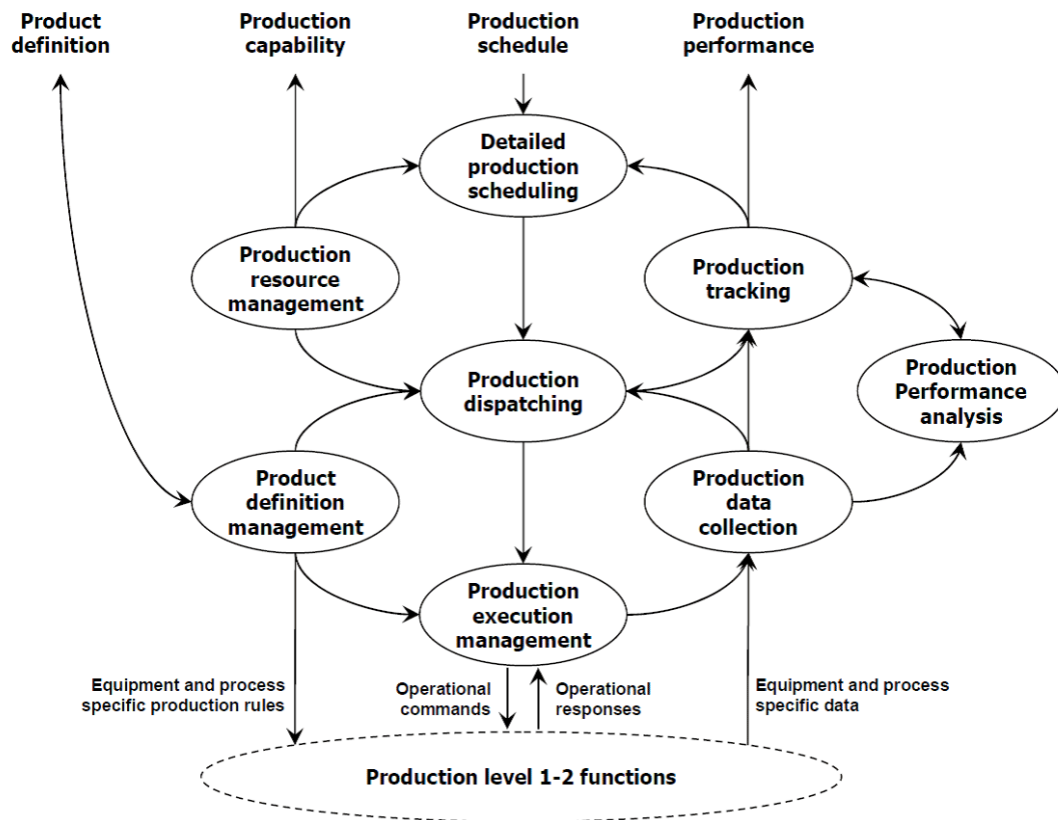


Figure 1 – See fine arts at Museum



IEC

Figure 2 – Production operations management (extracted from IEC 62264-3)

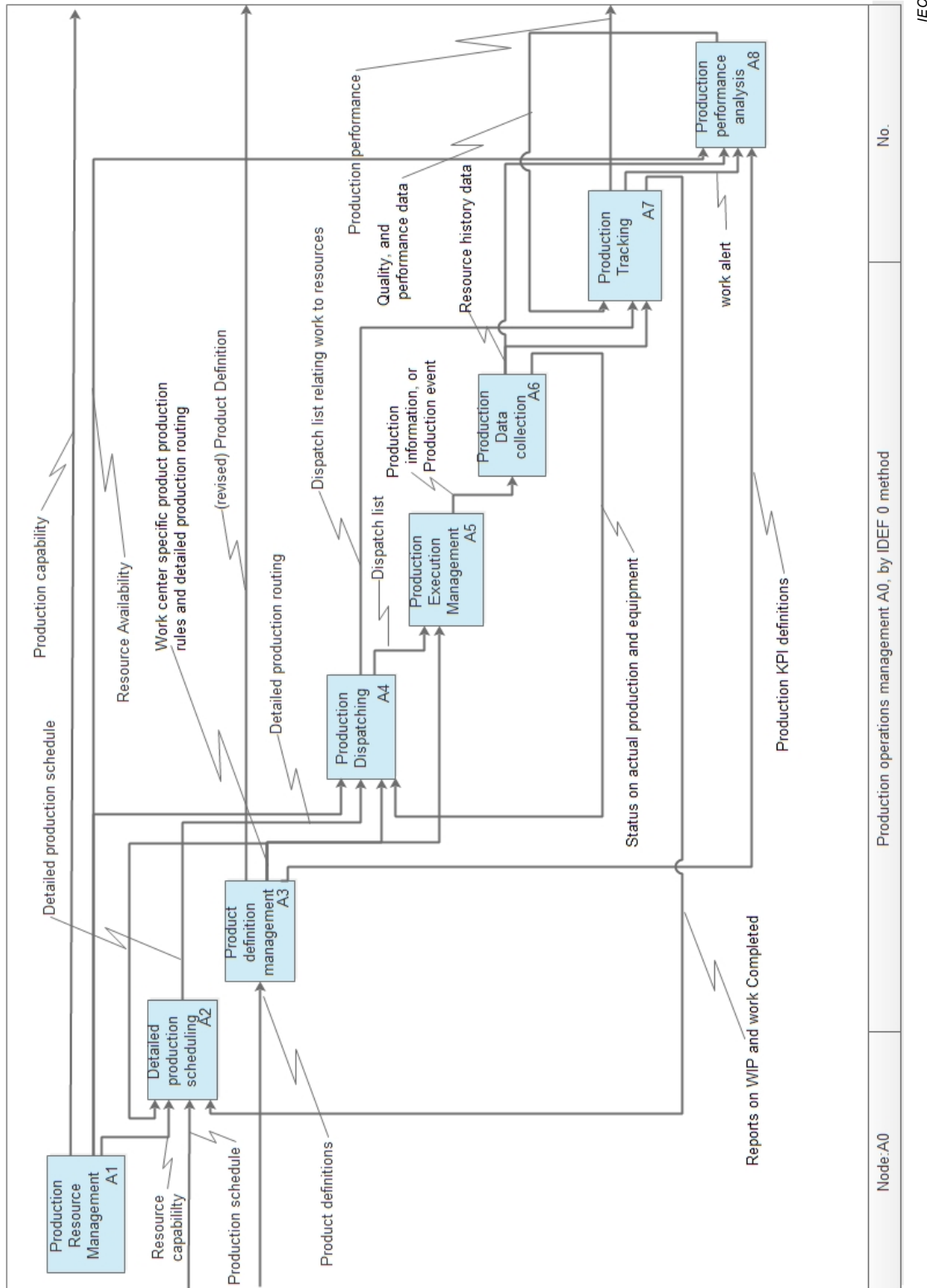


Figure 3 – Production operations management modelled in PAM and depicted as IDEF-0 diagram

4.3 Relation among properties of different activities

Since an activity is modelled as a class and each class has properties, there may be several types of relations, among a set of activities, a set of properties, or a set of activities and properties. Some of those relations have a direction from a source to a destination, where both the source and the destination may comprise a collection of activities, whilst some others have no direction among such entities constrained by the relation. Note that the IDEF0 depicts flows of thing among activities, and does not specify further constraints among properties of both the source and the destination, thus it deploys a directed relation from a set of activities to another. This type of relation is realized by a functional relation whilst the non-directional one is by a predicative relation, both of which are defined in IEC 62656-1. There are cases, however, where the constraints among the properties of the source and destination need to be further specified beyond the capability corresponding to IDEF0. The method to superimpose this detailed relation on another relation expressing a flow is described in 5.13.

4.4 International Concept Identifier (ICID)

The International concept identifier or ICID is defined in IEC 62656-1 as a specialization of the International Registration Data Identifier (IRDI) defined in ISO/IEC 11179-6, where the difference is only in the choice of separator characters between the significant parts of the identifier sequence, i.e., the Registration Authority Identifier (RAI), the Data Identifier (DI), and the Version Identifier (VI). Namely, the ICID uses a set of pound signs “#” for marking separation among the parts (for more details, see IEC 62656-1);

ICID:= RAI “#” DI “##” VI “###” NOTE,

where RAI stands for Registration Authority Identifier, DI for Data Identifier and VI for Version Identifier. The “NOTE” is a string of characters that includes no “#” within and serves as annotation for users. Besides, the RAI has been traditionally called as “information supplier” “data supplier” or more simply “supplier” in an old IEC 61360-2 and ISO 13584-42 vocabulary.

However in the case of the POM, the RAI and VI are usually pre-set by “#DEFAULT_SUPPLIER” and “#DEFAULT_VERSION” instructions for any occurrences in the header part and “#DEFAULT_DATA_SUPPLIER” and “#DEFAULT_DATA_VERSION” instructions for any occurrences in the data section, of a Parcel sheet. Thus in the following, RAI and VI are assumed to be predefined as default in the header section of each parcel sheet. For use of the default parameters, see IEC 62656-2 for more detail.

The use of ICID allows a sequence of activities to be explained in multiple languages whilst all the identification and referencing are done through a language independent global ID that is fully version controlled. This facilitates the related knowledge about activities to be easily shared and updated among users, which expects less confusion and more rational decisions among users. This feature is all the more important when we consider database maintenance of industrial ontologies. For more details, see IEC 62656-3. Also note in Figure 2 that each of the subcomponents of the generic activity “to see fine arts at museum” can have a specialization tree, and also a composition tree which latter comprises subcomponents.

5 Basic structure of the PAM

5.1 Activity and arrows

An activity is modelled as a class. Thus it may have several characteristic properties. For example, if an activity is to maintain the temperature of a coolant of a reactor cooling system, temperature range of the coolant must be one of such properties. A property “flow rate of the coolant” may be another. Such properties may be defined as characteristics of an activity or may be imported through case-of from a product class, such as the reactor cooling system, which must be registered as a component class in IEC 61360 CDD. Typically, in the case of properties of an activity, there may be several status properties that specify the conditions of a device or equipment.

An arrow represents an information flow or material flow among activities including the flow of people, which is modelled as a function and is instantiable as specific arrows in the relation meta-class (MDC_C011) in the POM. A function is a specialization of a relation thus it has the domain and the codomain (range) of the function.

To designate exactly that a function means an arrow that maps some instances in an activity specified as source to an instance in another activity specified as destination (target), the attribute of the function “role of the relation”(MDC_P210) shall be assigned a predefined value “arrow” in the relation meta-class. Simultaneously, in the definition of the function, the activity or set of activities in the source shall be designated in an attribute named “domain of the function”(MDC_P202) whilst the activity that serves as the destination of an arrow shall be designated in an attribute named “codomain of the function” (MDC_P203).

5.2 Subactivities

5.2.1 General

Before all, as already briefly sketched in 4.2, there are two types of hierarchical structures in the POM. One is called a “specialization tree” and the other is called a “composition tree”. The PAM is just a specialized use case of the POM for modelling a collection of phenomena known as activities, thus the PAM basically inherits the basic hierarchical structures of the POM.

5.2.2 Specialized activity

Since an activity is defined to be a class, there may be more a generic description of the activity as well as more a specialized description of the activity. This is nothing different from a class specialization hierarchy (i.e. tree) of products, as we find in IEC 61360 CDD. For example, an activity “design products” must be more a generic activity concept than an activity “design electro-technical products”, while the activity “design electro-technical products” must be more a generic concept than that of “design electric motors”. In brief, in an analogy with a product specialization hierarchy, there must be an activity specialization hierarchy, provided there are certain characteristic properties of activities (not of subject products) that characterize the specialized activities in a lower branch of the tree.

5.2.3 Component activity

An activity may have several sub-components, in the sense of being a part of the former, where each sub-component is called a “component-activity”, “child activity” or “subactivity” in this International Standard. Conversely an activity having sub-components is called a “composite activity” or “parent-activity” of the latter. Note that the term “super-activity” is not used in this document for fear of being mistaken for superclass. A composite-activity is sometimes referred to as the “holding activity” of its sub-components, whilst the subcomponents are sometimes referred to as the “child activities” of the parent activity. A component-activity of an activity is a part-of the latter, and not a special kind of (is-a, i.e., specialization of) it, although some of the arrows that are applied to the parent activity node can be selectively applied to each of its child activities. The “part-of” relationship is formally called in the POM terminology as the composition and it is conceptualized as “is-composed-of” but currently realized through each class reference type property. The selective inheritance of arrows (flows) to a subactivity is realized in the PAM through the use of a “is-case-of” relationship between two classes originally defined in the IEC 61360-2/ISO 13584-42 common data model. Incidentally, a component-activity normally inherits some of the arrows of its parent, and a child-activity is sometimes referred to as “subactivity”, as if it were a specialization of the parent. But in the true sense, a subactivity is not a specialization of the parent, but rather a decomposition of the parent, thus it is in a composition relation. Figure 4 illustrates this situation, where a subcomponent and its parent are modelled simply as a class named “activity” and it has several components through the attribute “*is composed of*” and it imports several arrows through the attribute “*is case of*”. Note that the “superclass” here is not used in the sense of “is_composed_of” (part-of) relationship, but strictly in the sense of “is-a” relationship to point to a generic class, the use cases of which are explained in 4.2.

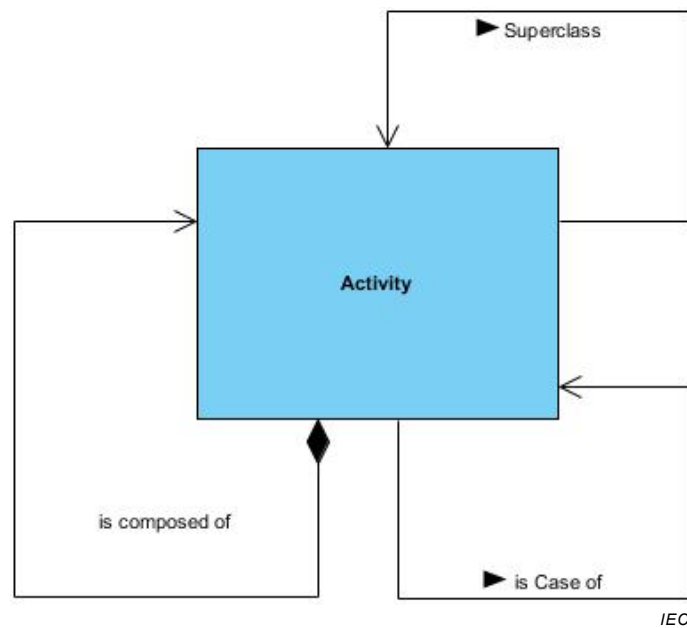


Figure 4 – Basic activity and its subcomponents

5.3 ICOM representation

Flows that enter into an activity are grouped into 3 types in IDEF0; input (I), control(C), or mechanism (M), whilst the flows emanated from an activity are all categorized in the output (O) group in IDEF0. As a whole, this grouping mechanism is known by an acronym "ICOM" that concatenates the initial letters of the labels of 4 groups. In an IDEF0 diagram, a flow belonging to each group may emanate from or arrive at an activity only from a designated side of the activity, as depicted in Figure 6. Namely, the input to an activity enters into the activity through the front side, whilst the control signal, message or information comes into the activity from the upper side, and the mechanism (or sometimes "M" is understood to be standing for the "material" or "means") provides support to the activity from the lower side, whilst the output flows from the rear side of the box, as depicted in the figure. To correspond to this system of labelling, the PAM models the distinction by assigning either one of "input", "control", "output", and "mechanism" label to an attribute named "segment" (MDC_P211) of an arrow, which is a function modelled in the relation parcel. The difference between a flow labelled as "input" and another labelled as "control" is quite subtle. However, the control can be considered as a flow of information that provides constant criteria for the actions of an activity, or as an application of the constraints on an activity that affects the output. In an enterprise, it will typically take the form of management, or sometimes a due diligence check. There is another label named "connectivity" that is used to describe a connection point of arrows at frame boundaries of an activity. The use of this label is special, and it is elaborated in 5.10.4 in more detail.

5.4 Role of the mechanism (M) in the PAM

Before explaining a parcel sheet specification conformant to the PAM, it is worthwhile studying what role the mechanism (M) plays in integration of the various types of ontologies. Unlike an activity diagram in UML, an activity in IDEF0 can elaborate on the natural or industrial environment in which the activity takes place, by virtue of the flows entering into an activity from mechanism(M) side specified in the "segment". In other words, what an activity in the PAM provides with its structure is not just a functional description of the process between input and output, but also a specification of an adverbial context in which the activity takes place. It is partly because an activity in the PAM can deal not only with the flow of information, but also with the flow of physical resources, let alone the human resource. Suppose we describe a factory of semiconductor chips, then some facility in the factory should take some base semiconductive material as input and transform it into a batch of semiconductor chips as output. In addition, the facility must be looked over by some process engineers and managers. In this case, such a facility and humans are indispensable goods for the activity called

“semiconductor production”, but they are neither input nor output, because this activity will not consume, transform nor reduce those resources. It is rather appropriate to regard them as an indispensable environment or means for the activity, which is labelled as Mechanism in the IDEF0 terminology. Conversely, in the IEC CDD, such means will be typically provided as product classes. Hence, the PAM whose activity description can differentiate input, output, control, and mechanism will help integrate the descriptions of product classes, material classes, activities and their interactions, which are typically deployed within an enterprise.

5.5 External function call

So called “call” arrow which is seldom drawn in IDEF 0 diagram from the mechanism side of an activity to downward, as shown in Figure 6, is not directly used in the PAM, because it is used to call an external function outside the diagram, thus the function itself might not be stored in IEC CDD as part of an ontology. For this reason, in Figure 6, it is shown as an arrow with a dotted shaft. This can be usually replaced with a functional relation with the same name associated with an activity from which the call arrow emanates. Nevertheless the keyword “call” is reserved for specifically relating the call arrow to an externally prepared function outside the diagram at hand, or possibly to a function prepared outside the current ontology in the POM. Example is such that a service provided by an operating system. In other words, it is not being owned by the activity. This means in a figure generated by the PAM, the call arrow will not be drawn nor seen, but the external function must reside in a parcel and be identifiable. The specification can be given in the same way as other arrows, but a call arrow shall be marked “call” in the attribute identified by the code MDC_P211, and named “segment”. The call arrow need not be actually drawn in the figure produced by a parcel tool, but it shall reference an external function when it is evoked. However, it is up to each implementer of the tool to install the actual reference mechanism. In any case, the body of the definition of the external function is platform-dependent, thus cannot be stored in the IEC CDD.

5.6 Basic PAM notation with function symbols

As noted in 5.1, arrows are represented as functions in the PAM. Thus they may be referred to by their relation IDs. Here, it is reminded that the use of italics for the IDs of the functions is just for ease of understanding, and thus f_1 , f_2 and f_3 will be just f1, f2, f3 or F1, F2, F3, respectively in the actual specification in the parcel sheets. Likewise A_1 , A_2 , A_3 shall be used for A1, A2, A3.

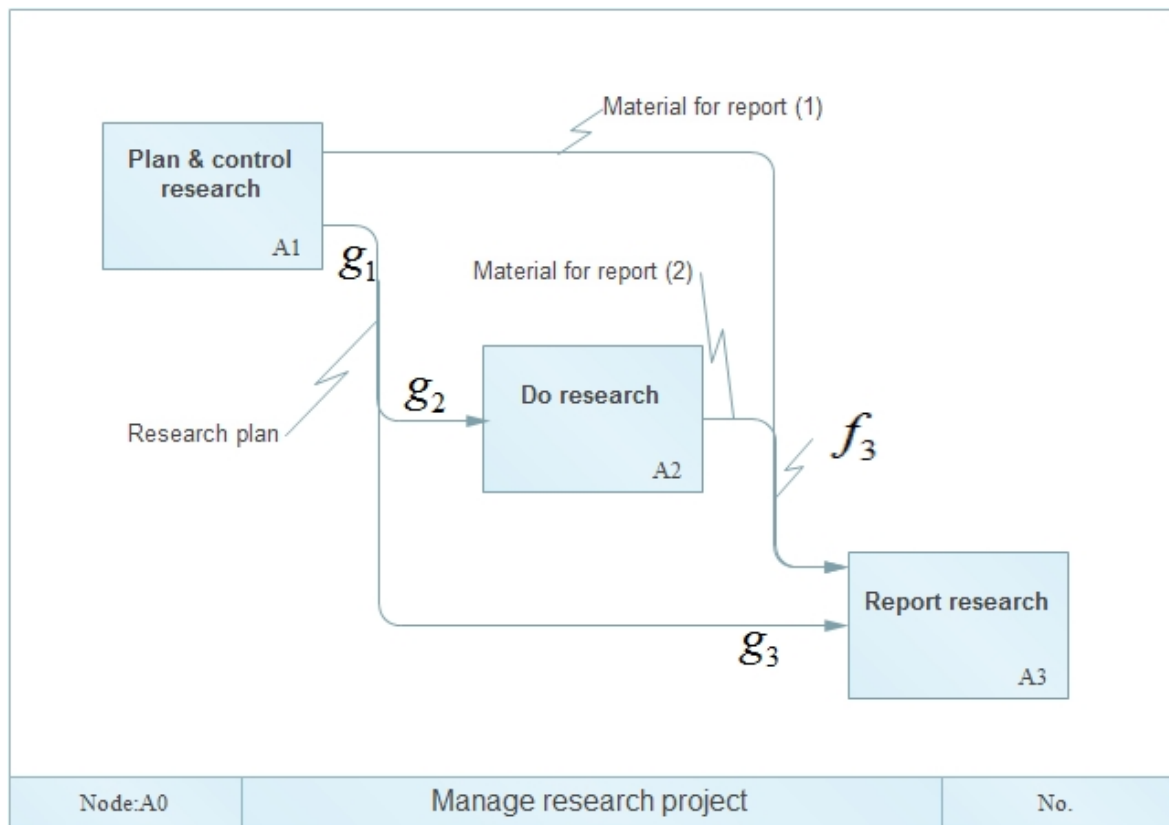
Note also that instead of $f_3(f_1(A_1) + f_2(A_2)) \rightarrow A_3$ for the situation that an arrow f_1 and f_2 meet at some point and converge into another arrow f_3 , we prefer to express this by a simpler formula $f_3(A_1, A_2) \rightarrow A_3$, signifying the arrow f_3 takes two activities A_1 and A_2 as the domain and will map them to the codomain A_3 , regardless of whether the arrows from A_1 and A_2 actually meet at a point before they reach A_3 in graphics or not, and transform themselves into another arrow f_3 . This is expressed in the parcel as in the row marked “G3” in the attribute “code”(MDC_P001_13) in Table 1.

On the other hand, if $g_2(g_1(A_1)) \rightarrow A_2$ and $g_3(g_1(A_1)) \rightarrow A_3$, namely if an arrow g_1 from A_1 forks into two arrows, g_2 and g_3 , and they reach A_2 and A_3 separately as the target (i.e., codomain), the forking shall be explicitly marked as a connectivity in the “segment” attribute (MDC_P211).

In the table below, it is assumed that arrows f_3 , g_2 and g_3 all enter into activities as their inputs.

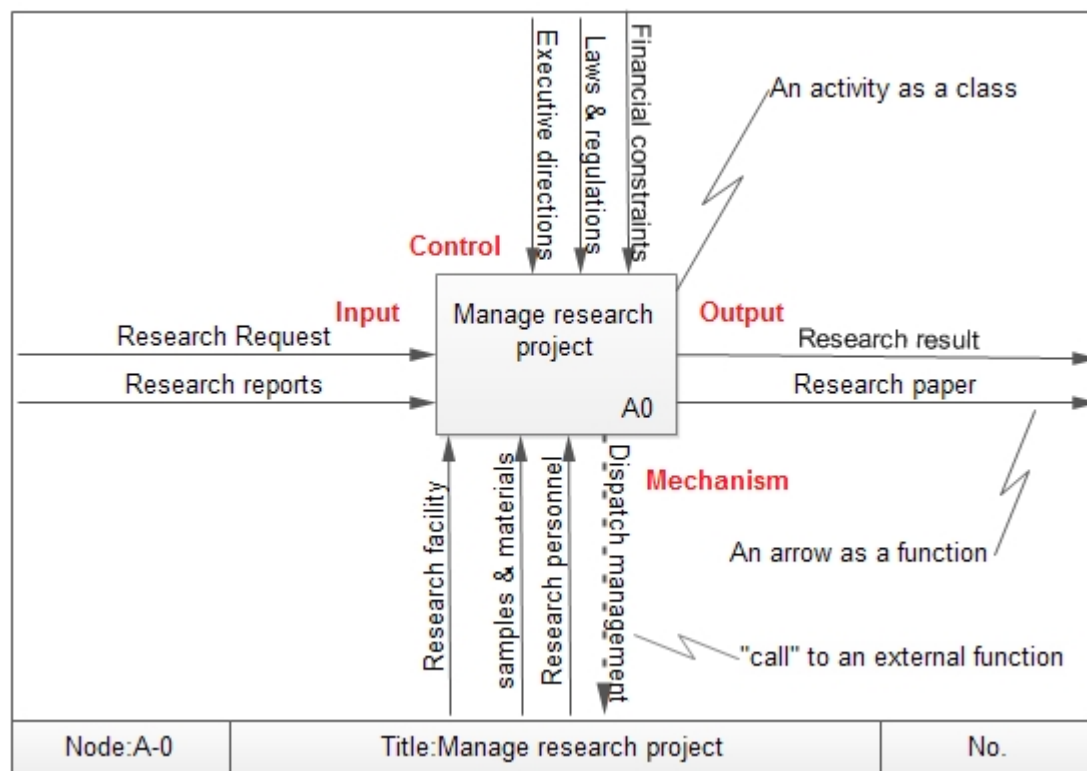
Table 1 – Basic PAM notation for arrows

MDC_P001_13	MDC_P004_1.en	MDC_P202	MDC_P203	MDC_P210	MDC_P211
Code	Preferred name	Domain of the function	Codomain of the function	Role	Segment
F3	Material for report	{A1, A2}	A3	arrow	input
G1	Research plan	A1	{G2, G3}	arrow	connectivity
G2	Research plan	G1	A2	arrow	Input
G3	Research plan	G1	A3	arrow	input



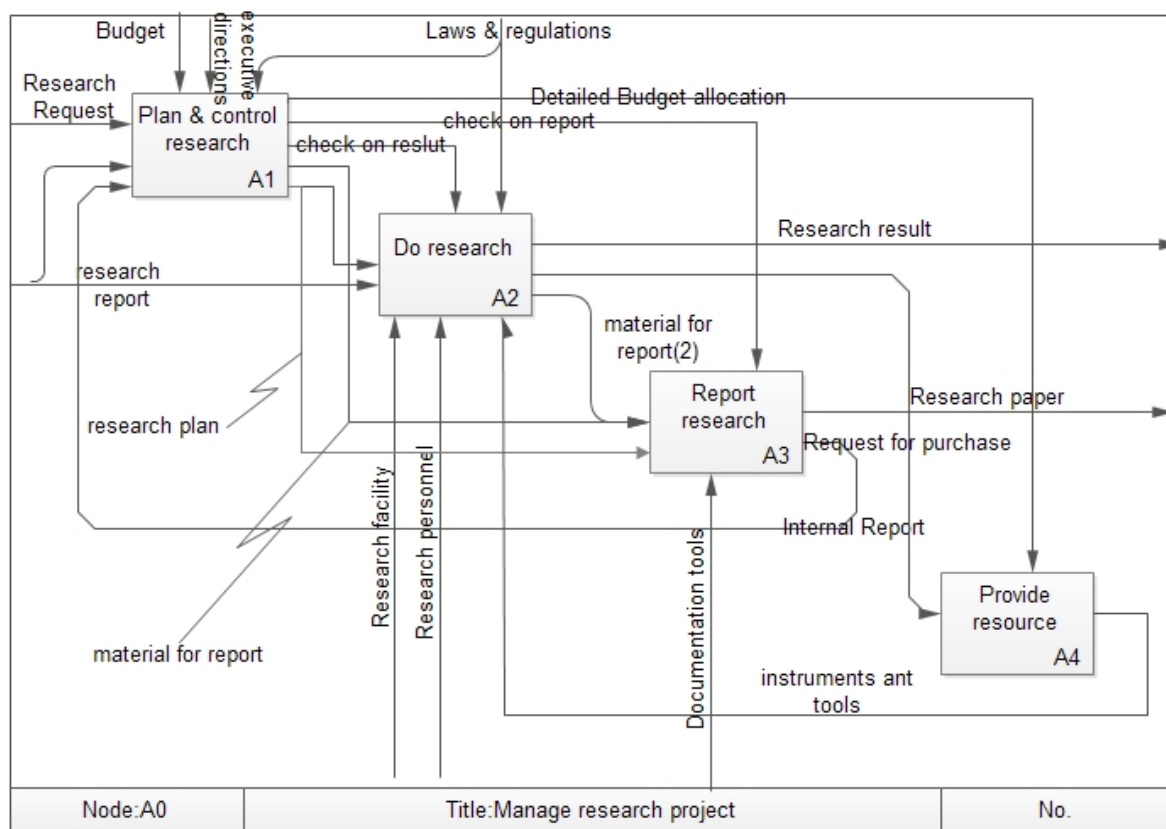
IEC

Figure 5 – Corresponding IDEF0 diagram for basic PAM notation



IEC

Figure 6 – Sample activity drawing in IDEF0 and ICOM



IEC

Figure 7 – Subactivities and arrows

5.7 Joining arrows

Joining of arrows means more than one source activities have arrows into the input side of one and the same target activity, where the codomains of the arrows are identical. Otherwise, seemingly overlapping arrows at the target are just by coincidence in a graphical presentation, and they are distinct by definition.

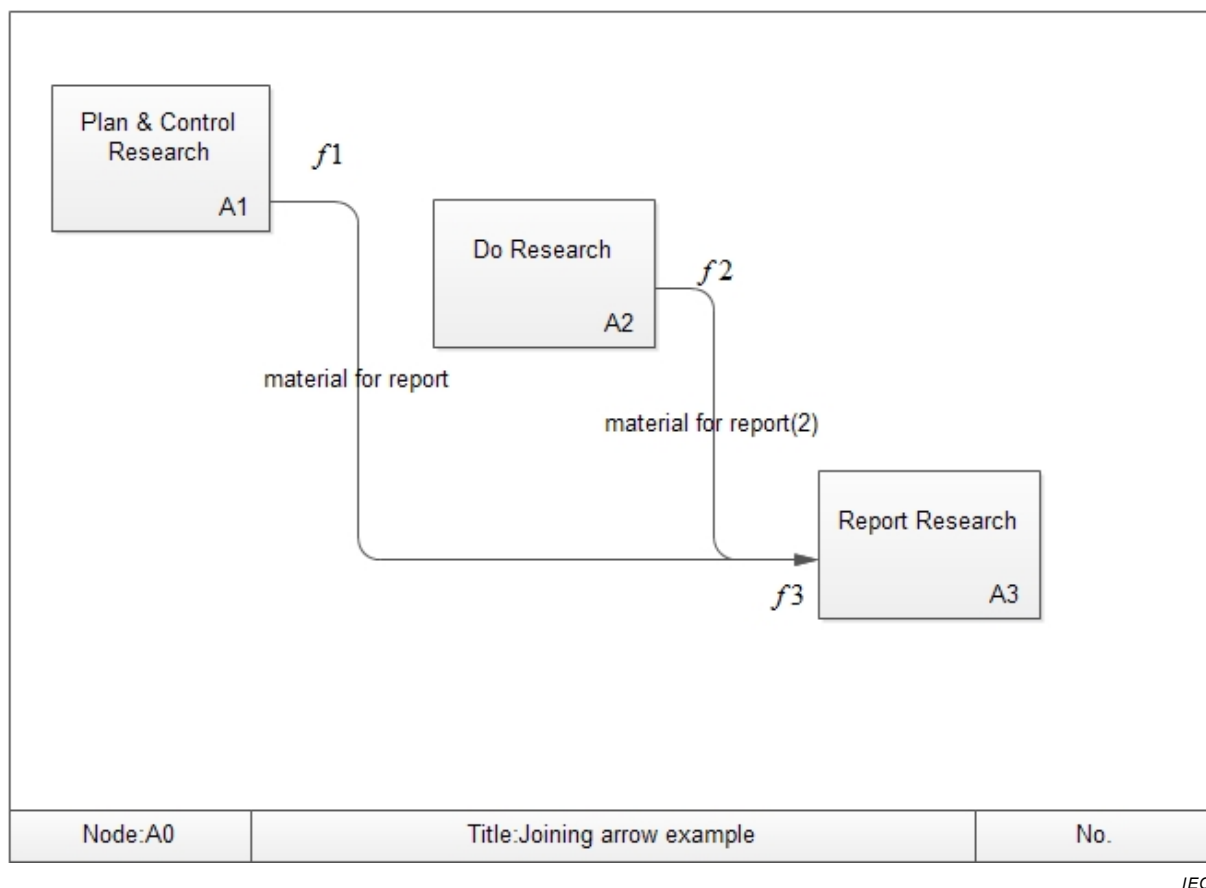


Figure 8 – Joining arrow example

Now, let f_1 , f_2 and f_3 be the identifiers (IDs) of arrows and A_1 , A_2 and A_3 be the identifiers of activities. Suppose that arrows f_1 and f_2 emanate from the activities A_1 and A_2 respectively and they merge into another arrow f_3 which arrives at A_3 . If the arrows f_1 and f_2 are joining into the arrow f_3 , this means f_3 has both of the arrows f_1 and f_2 in the “*domain of the function*” (MDC_P202) of the f_3 arrow. This may be represented as follows:

$$f_3(f_1(A_1) + f_2(A_2)) \rightarrow A_3,$$

where the summation operator “+” must be understood such that the resulting instances from $f_1(A_1)$ and $f_2(A_2)$ are added as rows of instances in the domain of the function f_3 .

However, there is much a simpler way to achieve the same end. In brief, it can be realized by putting A_1 and A_2 as the domain of a function f_3 without direct reference to either of the functions, f_1 and f_2 . This means users actually don’t have to define f_1 and f_2 but just f_3 (so it may be labelled anew as a composite function g_3). In this case, the users should be

conscious about the final destination of an arrow when they draw g_3 as an arrow, but may forget about which other arrows, the arrow shall be joined with. In formalism,

$$g_3(A_1, A_2) = A_3, \text{ or } g_3: A_1 \times A_2 \rightarrow A_3.$$

5.8 Forking arrows

An arrow may branch into several arrows at a middle point or points and each of which may reach a separate activity at the end. Let f_4 , f_5 and f_6 be identifiers (IDs) of arrows and A_1 , A_2 and A_3 be those of activities. Now we suppose the arrow f_4 bifurcates into arrows f_5 and f_6 , where f_5 and f_6 reach activities A_2 and A_3 , respectively. Then the consequent forking arrow can be regarded as two distinct composite functions, i.e.;

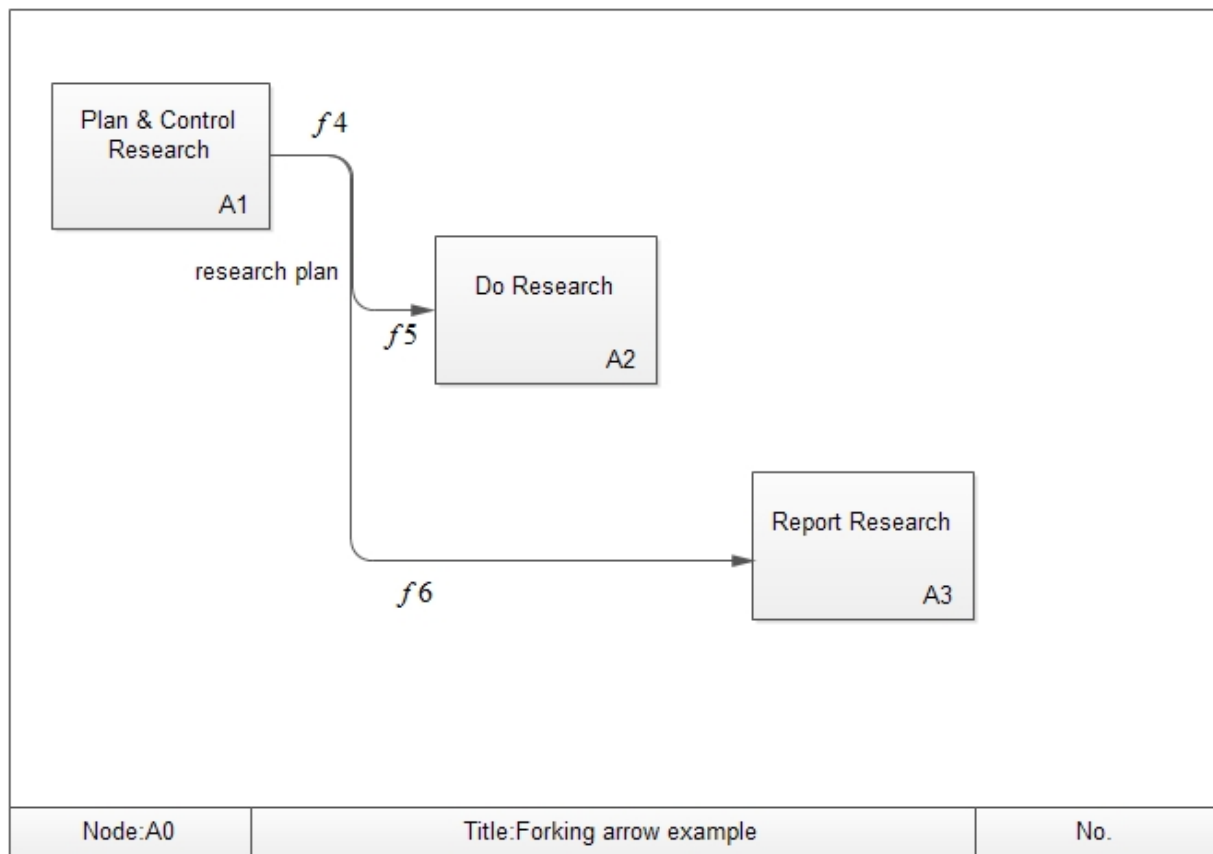
$$f_5 \cdot f_4: A_1 \rightarrow A_2 \Leftrightarrow f_5(f_4(A_1)) = A_2 \Leftrightarrow g_5(A_1) = A_2, \text{ where } g_5(A_1) = f_5(f_4(A_1)), \text{ and}$$

$$f_6 \cdot f_4: A_1 \rightarrow A_3 \Leftrightarrow f_6(f_4(A_1)) = A_3 \Leftrightarrow g_6(A_1) = A_3, \text{ where } g_6(A_1) = f_6(f_4(A_1)).$$

The above functions are simply achieved by describing the code of the arrow f_4 in the attribute “domain of the function” of both f_5 and f_6 , respectively, in the PAM.

5.9 Branching or joining of arrows

Branching, joining or connection points of arrows are not expressly modelled as distinct points. Neither, they are graphically marked in IDEF0 diagram. However, some other graphical modelling languages of activities expressly model the branching or joining.



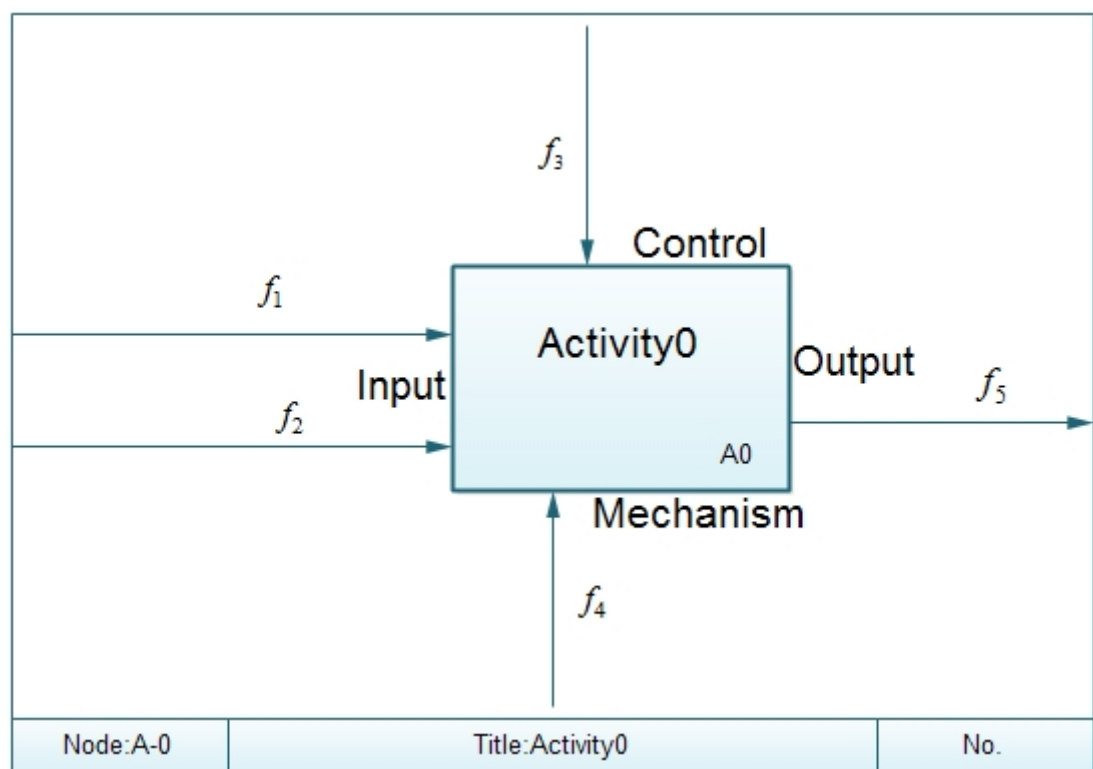
IEC

Figure 9 – Forking arrow example

5.10 Transcendental arrows

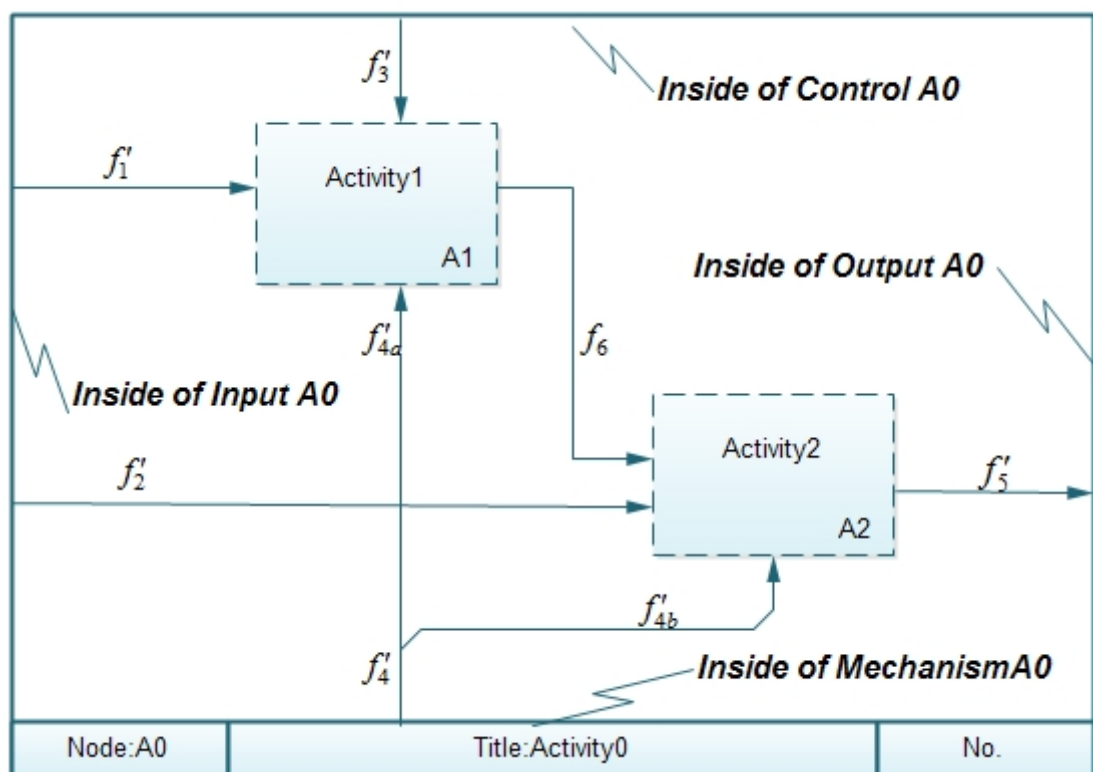
5.10.1 General

In the true sense of the word, subactivities do not inherit arrows from their parent node, rather the frame just below the node where the subactivities are drawn contains all the arrows that are coming into and going out of the parent node, because the inside of the frame means being inside of the parent activity. For example, in Figure 7 the frame identified as “Node:A0” depicts the interactions among the activities A1 through A4 inside the activity A0, and all the incoming and outgoing arrows into and out of A0 illustrated in Figure 6 appear within the rectangular frame shown in Figure 7. For further understanding, please see Figure 10 and Figure 11 below. The reason why the IDs of arrows in Figure 11 are primed is explained later.



IEC

Figure 10 – Transcendental arrows to be taken over by child nodes



IEC

Figure 11 – Transcendental arrows from the parent node

As you see, arrows f_1 and f_2 entering into the activity A_0 from the **Input** side of the activity box (in Figure 10) emerge as arrows f'_1 and f'_2 respectively, from the corresponding side of the rectangular frame A_0 (in Figure 11) in its sub-node. In the same manner, an arrow f_3 that enters into the activity A_0 (in Figure 10) from the **Control** side shoots up as an arrow f'_3 toward the activity A_1 within the rectangular frame (in Figure 11), whilst the arrow f_4 that enters into A_0 from the **Mechanism** side (in Figure 10) extends upward as an arrow f'_4 and reaches the activities A_1 and A_2 (in Figure 11) as arrows f'_{4a} and f'_{4b} after bifurcation. The outgoing arrow f_5 that emanates from the **Output** side of the activity A_0 (in Figure 10) emerges as an arrow f'_5 emanating from the **Output** side of the activity box A_2 (in Figure 11). In the node A0 shown as Figure 11, an arrow f_6 is newly created to connect activities A_1 and A_2 .

5.10.2 Modelling incoming arrows

Modelling incoming arrows requires that forking, joining and transcendent arrows be correctly represented as relations or functions in the PAM. This necessitates all types of arrows be correctly represented as functions on the boundary of activity box where an arrow enters into an activity through either one of the 3 sides of the box, i.e., Input, Control, or Mechanism (from the side of Output no arrow may enter). Note that the arrow f_2 extends from the activity A_{-0} (minus zero) to the activity A_0 whilst the arrow f'_2 extends from the activity A_0 to the activity A_2 . This means the domains and codomains of the two functions are completely different (but connected). In the same manner, the arrow f_3 extends from the activity A_{-0} to the activity A_0 whilst the arrow f'_3 extends from the activity A_0 to the activity A_1 . Following the pattern, the arrow f_4 extends from the activity A_{-0} to the activity and the arrow f'_4 extends from the activity A_0 to both of the activities A_1 and A_2 after having bifurcated its head as arrows f'_{4a} and f'_{4b} . Thus, they need to be expressed as distinct functions, on mathematical terms. The following Table 2 summarizes the above relationship, where the codes, AM0, A0, A1, and A2 denote A_{-0} , A_0 , A_1 and A_2 , respectively.

5.10.3 Modelling outgoing arrows

Analogous to the case of incoming arrows, modelling the outgoing arrows requires that forking, joining and transcendent arrows be correctly represented as relations or functions in the PAM. Unlike incoming arrows, however, an outgoing arrow always emanates from the **Output** side of an activity box and it either enters into other activities from one of **Input**, **Control**, or **Mechanism** side of the box, or reaches the output side of the frame to eventually go out of the frame. In Figure 11, an arrow f_6 is an example of the former case, and an arrow f'_5 is an example of the latter case. Namely the arrow f_6 emanates from the output side of the activity A_1 and enters into the input side of the activity A_2 , whilst the arrow f'_5 emanates from the output side of the activity A_2 and reaches the frame that corresponds to the inside of the output of the activity A_{-0} . The PAM representation of the outgoing arrows is already described in Table 2. See the rows of f_5 and f'_5 (By codes, they are represented as F0005 and F0015, respectively) in the table.

5.10.4 Modelling connections of arrows at frame boundary

In Table 2, at the connection of functions between the arrows f_1 and f'_1 at the frame boundary, the domain of the function f'_1 is specified as f_1 . Likewise, for the arrows f_2 and f'_2 , the arrows f_3 and f'_3 , and the arrows f_4 and f'_4 , the respective domains of the functions f'_2 , f'_3 and f'_4 are specified as f_2 , f_3 and f_4 and not as A0. This is because, the domains of the functions are not a property value of the A0, but exactly the codomains (i.e., the outputs) of the functions f_1 , f_2 , f_3 and f_4 . In the same manner, at the connections of the functions f_5 and f'_5 , the codomain of the function f'_5 is specified as f_5 , instead of A0, with an aim to specify that f'_5 overwrites the domain of f_5 , as the value available from the codomain of the f'_5 . This does not entail that the codomain of the functions f_1 , f_2 , f_3 and f_4 need to be changed, because when functions f_1 , f_2 , f_3 and f_4 are originally designed, functions f'_1 , f'_2 , f'_3 and f'_4 in the sub-node are not yet decided. For the same reason, the domain of the function f_5 is not required to be rewritten as f_5 , because the function f_5 is not known at the node A0. In the practical notation in the PAM, all these activity identifiers described between a pair of curly brackets beside a function ID, in the column for domain of the function or in the column for the codomain of the function can be omitted.

In principle, an arrow that appears within a frame of the code A_k ($k = -0, 0, 1$) must have its domain in activity A_k or one of its subactivities. The former case is that the arrow is transcendent from an upper node, and the latter case is that the arrow emanates from (the output side of) one of its activities in the frame. To understand where an arrow appears, see the “Definition class” (MDC_P021).

Note that the values appearing in the column under the “Definition class” (MDC_P021) originally defined in IEC 62656-1 are essential values in the POM and the attribute is designated as “key” for its requirement (MMDC_P102), but this may be used as the “frame of the arrow” in this part of IEC 62656 for facilitating the understanding about the layout of the activities in a frame like IDEF0. The values shall be defined by the supplier of a dictionary (ontology) of a set of activities and may be useful for understanding and implementing the tree structure formed by the activities and arrows, by users as well as applications.

Table 2 – Extracts of relation meta-class definitions for activities

#Preferred name	Symbols in the figures	CODE	Domain of the function	Codomain of the function	Definition class ^b
#Property ID	Not Available	MDC_P001_13	MDC_P202	MDC_P203	PDC_P021 ^b
	f_1	F0001	AM0 ^a	A0	AM0
	f'_1	F0011	F0001	A1	A0
	f_2	F0002	AM0 ^a	A0	AM0
	f'_2	F0012	F0002	A2	A0
	f_3	F0003	AM0 ^a	A0	AM0
	f'_3	F0013	F0003	A1	A0
	f_4	F0004	AM0 ^a	A0	AM0
	f'_4	F0014	F0004	{FA014, FB014}	A0
	f'_{4a}	FA014	F0014	A1	A0
	f'_{4b}	FB014	F0014	A2	A0
	f_5	F0005	A0	AM0 ^a	AM0
	f'_5	F0015	A2	F0005	A0
	f_6	F0006	A1	A2	A0
^a “AM0” signifies “A-0”, that reads “A minus zero” and designates the exterior of the Activity A0, depicted as the frame in which the activity box A0 is drawn. ^b “Definition class”(MDC_P021) is a meta-property originating in IEC 62656-1, but this may be used as the “frame of the arrow” in this part of IEC 62656 for facilitating the understanding about the layout of the activities in a frame of IDEF0-like diagram. The value must be defined by the supplier of a dictionary (ontology) of a set of activities and may be helpful for understanding of the tree structure formed by the activities and arrows. Readers are reminded that AM0, A0, A1, etc., in the “Domain of the function”, “Codomain of the function” and “Definition class” are not the real codes of the activities as classes defined in the attribute MDC_P001_5. They are simplified and set identical to activity IDs usually generated on the fly by a graphical tool.					

5.10.5 Contracted form of representation for branching and joining arrows

Table 3 shows a contracted form of connectivity between an arrow (f_4) and 2 forking arrows (f'_{4a}, f'_{4b}), where bold letters are used for the codes of the arrows affected by the contracted form of representation. Namely, f_4 (F0004) and f'_{4a} (FA014), f'_{4b} (FB014) are directly connected without an intermediary arrow f'_4 (F0014) by specifying f_4 (F0004) as the domain of f'_{4a} (FA014) and f'_{4b} (FB014), respectively. The rest of the rows in the table remains unaffected. Note that whether f'_4 (F0014) exists or not does not mathematically affect the specification of the forking arrows, since f'_4 is identical to f_4 , and the forking from f_4 into f'_{4a} and f'_{4b} , may take place directly on the frame boundary, with the arrow f'_4 being degenerated into one with zero length.

5.10.6 Domain or codomain overloading for transcendent arrows

Notwithstanding the shortcut of intermediary arrows for forking arrows explained in the foregoing clauses, it is reminded that users cannot completely remove the “dashed” arrows f'_1 , f'_2 , f'_3 and f'_5 for the original arrows f_1 , f_2 , f_3 and f_5 , in which no forking takes place. This is because, when the arrows f_1 , f_2 , f_3 and f_5 are designed, decomposition of the A0 into subactivities is not yet thought of or determined. In other words, A0 in Figure 10 is an abstraction of the activities A1 to A2 in Figure 11, or A0 is still at a stage of process of conception, therefore arrows f_1 , f_2 , f_3 and f_5 do not know the presence of activities A_1 (A1, by code) to A_2 (A2, by code), thus the codomain of the arrows f_1 , f_2 , f_3 shall be specified as A0, and the domain of the arrow f_5 shall be specified as A0. Whereas the arrows f'_1 , f'_2 , f'_3 and f'_5 are distinct from f_1 , f_2 , f_3 and f_5 because they no longer consider A0 as the true domain or codomain, thus the codomain of the arrows f_1 , f_2 , f_3 shall be overloaded in the frame A0, and the domain of the arrow f_5 shall be overloaded in the same manner (i.e., Figure 11). In consequence, the dashed arrows f'_1 , f'_2 , f'_3 and f'_5 which are specialization of arrows f_1 , f_2 , f_3 and f_5 are indispensable for correctly representing the overloaded arrows.

In other words, in the case of forking arrows, because of the presence of dashed arrows f'_{4a} and f'_{4b} , that represent the arrow heads after forking, the intermediary arrow f'_4 can be omitted.

Table 3 – Contracted representation for connectivity of activities

#Preferred name	Symbols in the figures	Code	Domain of the function	Codomain of the function	Definition class ^b
#Property ID	Not Available	MDC_P001_13	MDC_P202	MDC_P203	PDC_P021 ^b
	f_1	F0001	AM0 ^a	A0	AM0
	f'_1	F0011	F0001	A1	A0
	f_2	F0002	AM0 ^a	A0	AM0
	f'_2	F0012	F0002	A2	A0
	f_3	F0003	AM0 ^a	A0	AM0
	f'_3	F0013	F0003	A1	A0
	f_4	F0004	AM0 ^a	A0	AM0
	f'_{4a}	FA014	F0004	A1	A0
	f'_{4b}	FB014	F0004	A2	A0
	f_5	F0005	A0	AM0 ^a	AM0
	f'_5	F0015	A2	F0005	A0
	f_6	F0006	A1	A2	A0

^a “AM0” signifies “A-0”, that reads “A minus zero” and designates the exterior of the Activity A0, depicted as the frame in which the activity box A0 is drawn.

^b “Definition class”(MDC_P021) is a meta-property originating in IEC 62656-1, but this may be used as the “frame of the arrow” in this part of IEC 62656 for facilitating the understanding about the layout of the activities in a frame of IDEF0-like diagram. The value must be defined by the supplier of a dictionary (ontology) of a set of activities and may be helpful for understanding of the tree structure formed by the activities and arrows. Readers are reminded that AM0, A0, A1, etc., in the “Domain of the function”, “Codomain of the function” and “Definition class” are not the real codes of the activities as classes defined in the attribute MDC_P001_5. They are simplified and set identical to activity IDs usually generated on the fly by a graphical tool.

5.11 Extended semantics beyond IDEF0

5.11.1 Specialized types of activity and its icon

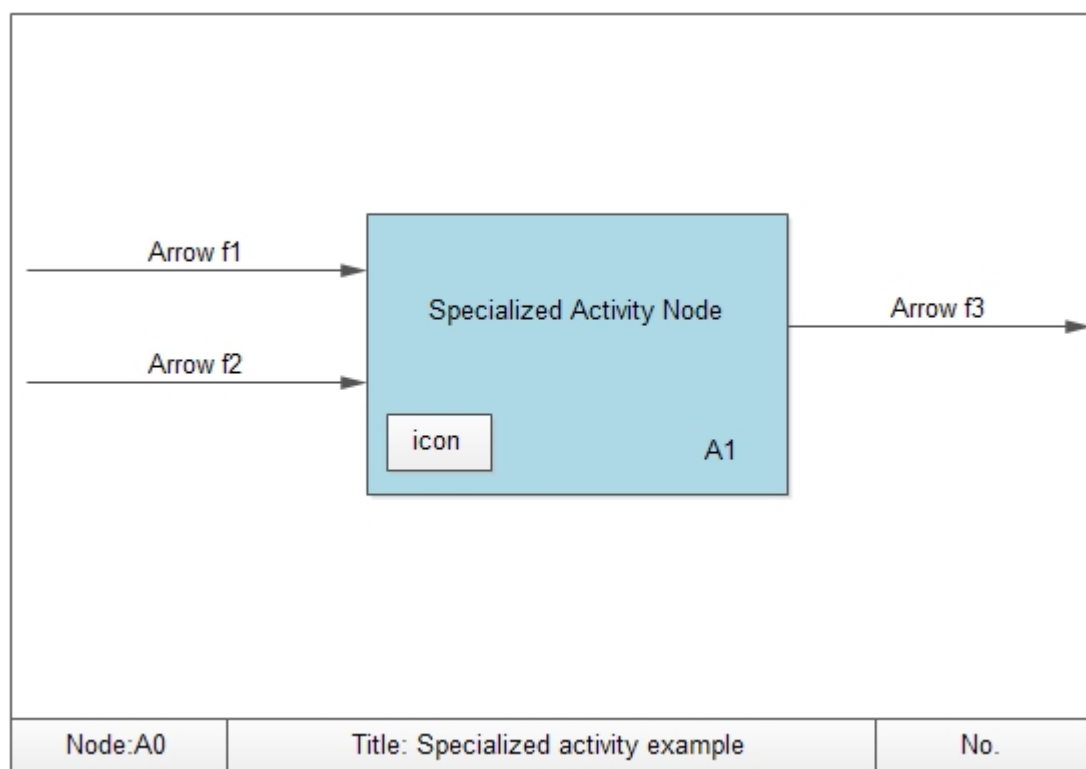
With respect to activities, IDEF0 has no provisions for attaching a condition, by which the outgoing arrows are triggered, i.e., the flow of data or material will be initiated. However, some graphical process description languages have the capability to do it. As noted earlier, detailed algorithmic description of the trigger of such processes is beyond the scope of this document, nevertheless, the following specification for the specialized types of activities may be useful for clarifying the context of the activities for the users:

- Conjunction, or “AND” node;
- Disjunction, or “OR” node;
- Complementation, or “NOT” node;
- Selection node, or “Dialogue” node;
- Transformation node.

Since the above specialization is outside the IDEF0, it is convenient to have some conventions for expressing or signalling the specialized use of IDEF0 in a graphical presentation of the activities. This part of IEC 62656 stipulates the conventions in the following manner:

- An icon or letters specifying the kind of the specialized node should be displayed at the lower left corner of an activity box;
- For conjunction, disjunction, and complementation, it is recommended that the symbols used for logical ICs be used for such an icon. For selection node, a symbol for switch is preferred for use;
- Exact size, colour, and position of the icon may be decided by each implementer, but the size of the icon must not occupy more than 1/4 of the area of an activity box.

The above three requirements are illustrated in the following figure.

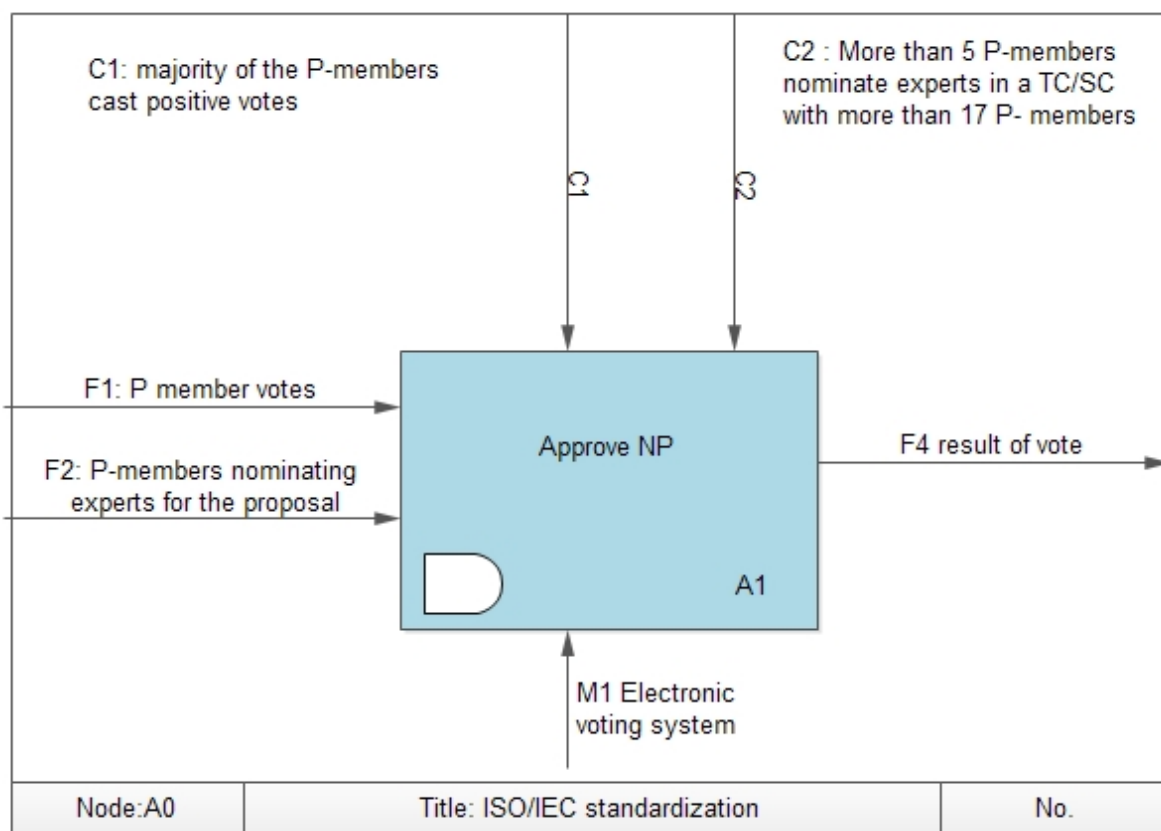


IEC

Figure 12 – IDEF0 extension for specialized activity node in the PAM

The following Figure 13 is an example, where the above principles are applied to a conjunction node.

In the example, the box labelled A0 is assumed to be an activity to approve a new work item proposal (NP) in a committee or subcommittee of IEC or ISO. It is also assumed that the arrow F1 sends the P-member votes about the NP with a list of value pairs, such as "(P-member country code, Yes/No/Abstain)", and the F2 gives a list of value-pairs, such as "(P-member country code, expert name)". It is also assumed that C1 and C2 set respectively the conditions for the input functions F1 and F2 to be judged "True". This can be literally achieved by setting the formula of the function of F1 and F2 by the logical values set by C1 and C2, provided the activity A0 is equipped with a function to count the number of P-member votes that replied "Yes" to the NP and among them the P-member countries that nominated experts to work on the NP.



IEC

Figure 13 – An implementation example of Conjunction node in the PAM

In implementation, it is totally possible that an entire activity box shape may be replaced by a specialized shape or an icon, such as one used for an AND node or an OR node in a digital logic IC circuits, but such a specialized specification of the graphic user interface is out of the scope of this document, and this part of IEC 62656 does not intend to normalize the interface beyond what the original IDEF0 specifies: what this document requests is that if a specialized node type is used, it shall be explicitly marked as such by a graphical means or by a string of letters.

5.11.2 Conjunction node

Conjunction node assumes that when all the arrows that enter into an activity are TRUE, in other words, having correct and just parameters in the domain and codomain of the arrow as a function, then the output arrow(s) shall follow. If some arrows do not satisfy this condition, then the activity shall be subdivided and such output arrows that do not satisfy this condition shall be separated in order to make them emanate from the separate activity. This process may be repeated and in some special case, each arrow may depend on the conjunctive “AND” of all incoming arrows of the AND node.

5.11.3 Disjunction node

Disjunction node assumes that at least one of the arrows that enter into an activity is TRUE, in other words, having correct and just parameters in the domain and codomain of the arrow as a function, then the output arrow(s) will follow. If some arrows do not satisfy this condition, then the activity shall be subdivided and such output arrows that do not satisfy this condition shall be separated in order to make them emanate from the separate activity. This process may be repeated and in some special case, each arrow may depend on disjunctive OR of all incoming arrows of the OR node.

5.11.4 Complementation node

Complementation node takes the complement of the value of the codomain of the function. Consequently, the arrow returns FALSE, then the complement of the arrow becomes TRUE. The complementation may be represented as a regular box with a designated icon, or it may be reduced to just a small “open circle” added to the arrow head, when it is drawn in a picture.

5.11.5 Selection node

Completely different from the foregoing contexts, the selection node assumes that any one of the outgoing arrows may be triggered by an external event, i.e., human selection or judgement, through an interface for dialogue. It is useful to combine an enumeration type property with simultaneous selection number more than 2, which may be interpreted to allow multiple simultaneous selections of outputs at a time, from a selection node. However, an actual graphical user interface specification is outside the scope of this document. The icon could be a rhombus shape typically used in an algorithmic flow chart, or a question mark in a letter.

5.11.6 Transformation node

The transformation node may be used to apply some defined operation to the values of arrows at domain or codomain. Thus it may be considered a kind of “filter”, “amplifier” or “modulator” for the functions represented as arrows.

5.11.7 Decision tree

Note that subactivities below a specialized type of activity may form a tree structure often known as “decision tree”, being composed of the subcomponents of the activity. An appropriate integration of selection nodes in the tree may provide a decision making guide for users of this part of IEC 62656.

5.12 Graphic properties of arrows

According to the POM defined in IEC 62656-1, an arrow is a function, therefore it does not have any characteristic properties as classes have, but it may have some graphic properties, that are used to draw a graph of the function. The graphic properties shall be specified using a meta-property named “graphic properties”(MDC_P008_3) of the relation meta-class. Note also that the POM allows a reference to an algorithm or program for generating graphics to be stored in another meta-property named “graphics”(MDC_P008_2). Thus, the parameters for the algorithm may be extracted from the aforementioned meta-property “graphic properties”. The practical use and specification of those meta-properties for generating images are left to each implementation.

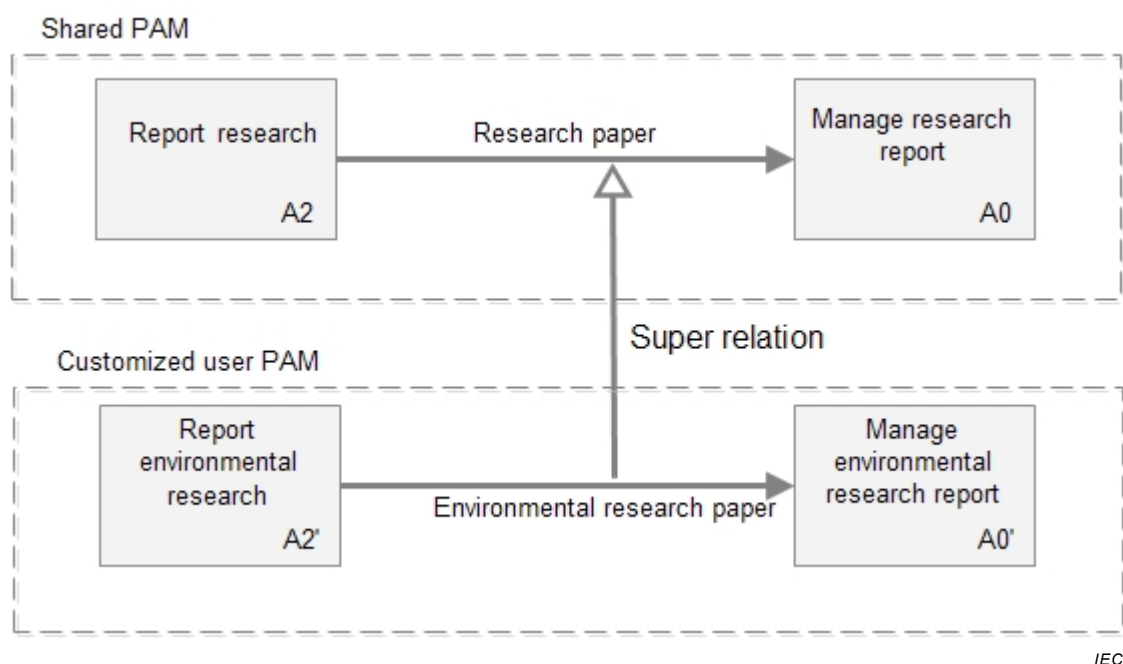
5.13 Arrow specialization

Methods of overloading an existing arrow to and from an activity with more specific details at lower nodes of the activity are already explained in 5.10.6. The elaborated arrows at lower nodes imply that they are specialized at a detailed design stage with subactivities, which usually takes place at some later stage in engineering design. Apart from this arrow overloading, there are general cases of specialization of relations including those used as arrows. This may occur when an activity at hand is more a specialized one than one postulated in a shared and known activity model. For example, in Figure 6 arrows representing flows of “research result” and “research paper” go out from “Manage research project” (A0). They are supposed to be overloaded by the arrows of the same names in Figure 7. Note that the subactivities of A0 in Figure 7 and the A0 are related by a component-of (part-whole) relationship as it is explained in Figure 4, and the subactivities only inherit some properties of A0 selectively through a use of case-of relationship. Thus the “research result” and “research paper” in Figure 7 are not necessarily the specialization of the arrows with the same names in Figure 6, but somehow deeply related. Thus they are overloaded functions for component activities which are classes. However, if a research to be managed is a clear subset of the research in A0, say for example, an environmental research or an

aerospatial research, then all the sub-activities of A0 in Figure 7 shall be addressing the specialized research originally postulated in A0. In this case, “research result” and “research paper” can be appropriately specialized as “environmental research result” and “environmental research paper” in essence, regardless of whether the names are actually modified or not. This is achieved by creating a specialized function, by connecting the “super-relation” (MDC_P212) attribute (meta-property) of the specialized function to the original function defined for either the original “environmental research result” or “environmental research paper” function, i.e., functional relation, respectively. Note that in this case, the subactivities of A0 in Figure 7 become subsets of the original ones.

This capability of the POM is very important and useful for application of a standardized and shared PAM by user organizations and enterprises, because it allows customization of the original model by each user entity without hampering the interoperability of the original shared model. See also Figure 1 for understanding where specialized visits for each subtype of A0 are conceivable.

Figure 14 depicts this relationship in a simplified diagram. Note that the box noted A0 at the right hand side of the arrow named “research paper” is the A0 frame-boundary that contains the subactivities of A0.



IEC

Figure 14 – Super relation and its application for specialized activity

5.14 Delegated formula interpretation

What needs to be accomplished by a function is not just drawing a line between the domain and the codomain of the function, but exactly specifying a functional relationship that exists among the properties of the activities, that are either designated as the domain or as the codomain of a function. In either case, the properties should be referenceable within a formula described in a meta-property named “formula” (MDC_P204), whilst the formula can be processed by an external solver, whose URL is specified in the meta-property named “external solver for the formula” (MDC_P206). Hence, the token representing a property with its ICID shall be universally identifiable and accessible, if the formula is intended to be understood by both the sender and the receiver of a parcel exchange. Thus the keywords used for the tokens given in Table 4 shall be reserved for the purpose. They shall not be used in user-proprietary formula expressions and in the external solvers.

Table 4 – Reserved keywords for formula interpretation

Keyword	Shortform	Synonym	Shortform for the synonym
\$item	N/A	\$thing	N/A
\$dictionary	\$dict	N/A	N/A
\$class	\$cls	\$concept	N/A
\$property	\$prop	N/A	N/A
\$identifier	\$ident	\$icid	N/A
\$enumeration	\$enum	N/A	N/A
\$datatype	\$type	N/A	N/A
\$document	\$doc	\$url	N/A
\$object	\$obj	\$instance	\$inst
\$uom	N/A	\$unit	N/A
\$term	N/A	\$constant	\$const
\$relation	\$rel	N/A	N/A
\$predicate	\$pred	N/A	N/A
\$function	\$func	N/A	N/A

Note that keywords will appear in such a way that they are entailed by a pair of parentheses in which an icid code is placed. For example, “\$class(*icid*)”, where an italic term means it shall be replaced by an actual icid code. It is also important to remember that keywords starting with ‘\$’ are case insensitive. Thus expressions like “\$CLASS”, “\$Class” and “\$class” mean one and the same entity.

Nevertheless, this part of IEC 62656 does not intend to define the precise use of the above keywords within this standard. Rather it indicates they will be specified in some other part of IEC 62656 and will be effective in interpreting the formula referenced in this standard, so the users are recommended not to use them for compatibility reasons in their proprietary formula and solver applications.

Annex A

(normative)

Meta-properties for activity description

A.1 General

This annex lists the meta-properties that may be used for activity descriptions. Note that for the description of arrows, the domain of the function (MDC_P202) and the codomain of the function (MDC_P203) are mandatory in this part of IEC 62656.

A.2 List of meta-properties

Table A.1 lists the meta-properties of the relation meta-class defined in the POM that may be used for description of the arrows between activities and the activities. Since the POM is completely defined in IEC 62656-1, this document only defines an interface for reusing the meta-properties of the POM.

Table A.1 – Meta-properties of relation meta-class used for activity description (1 of 2)

MMDC_P001	MMDC_P102	MMDC_P004_1.EN	MMDC_P004_1.FR	MMDC_P008	MMDC_P013
Property ID	Requirement	Preferred name in English	Preferred name in French	Data type	Version number
Identificateur de propriété	Exigence	Nom préférentiel en Anglais	Nom préférentiel en Français	Type de données	Numéro de version
MDC_P001_13	KEY	Code	Code	ICID_STRING	001
MDC_P002_1	KEY	Version number	Numéro de version	STRING_TYPE	001
MDC_P002_2.<lang>	MAND	Revision number	Numéro de révision	STRING_TYPE	001
MDC_P003_1	MAND	Date of original definition	Date de la définition originale	STRING_TYPE	001
MDC_P003_2	MAND	Date of current version	Date de la version actuelle	STRING_TYPE	001
MDC_P003_3.<lang>	OPT	Date of current revision	Date de la révision actuelle	STRING_TYPE	001
MDC_P004_1.<lang>	MAND	Preferred name	Nom préférentiel	TRANSLATABLE_STRING_TYPE	001
MDC_P004_2	OPT	Synonymous name	Nom synonyme	SET(0,?) OF LIST(2,2) OF STRING_TYPE	001
MDC_P004_3.<lang>	OPT	Short name	Nom abrégé	TRANSLATABLE_STRING_TYPE	001
MDC_P004_4	OPT	Name icon	Icône de nom	STRING_TYPE	001
MDC_P005.<lang>	MAND	Definition	Définition	TRANSLATABLE_STRING_TYPE	001
MDC_P006_1	OPT	Source document of definition	Document source de définition	STRING_TYPE	001
MDC_P007_1.<lang>	OPT	Note	Note	TRANSLATABLE_STRING_TYPE	001
MDC_P007_2.<lang>	OPT	Remark	Remarque	TRANSLATABLE_STRING_TYPE	001
MDC_P008_2	OPT	Graphics	Graphisme	STRING_TYPE	001
MDC_P008_3	OPT	Graphic Properties	Propriétés graphiques	LIST OF ICID_STRING	001

Table A.1 (2 of 2)

MMDC_P001	MMDC_P102	MMDC_P004_1.EN	MMDC_P004_1.FR	MMDC_P008	MMDC_P013
Property ID	Requirement	Preferred name in English	Preferred name in French	Data type	Version number
Identificateur de propriété	Exigence	Nom préférentiel en Anglais	Nom préférentiel en Français	Type de données	Numéro de version
MDC_P021	KEY	Definition Class	Classe de définition	STRING_TYPE	001
MDC_P112	OPT	Description	Description	TRANSLATABLE_STRING_TYPE	001
MDC_P113	OPT	Example	Exemple	STRING / LIST OF STRING	001
MDC_P200	MAND	Relation type	Type de relation	STRING_TYPE	001
MDC_P201	OPT	Domain of the relation	Domaine de la relation	LIST OF ICID_STRING	001
MDC_P202	MAND	Domain of the function	Domaine de la fonction	LIST OF ICID_STRING	001
MDC_P203	MAND	Codomain of the function	Co-domaine de la fonction	ICID_STRING	001
MDC_P204	OPT	Formula	Formule	STRING_TYPE	001
MDC_P205	OPT	Language for formula interpretation	Langage pour l'interprétation de la formule	STRING_TYPE	001
MDC_P206	OPT	External solver for the formula	Résolveur externe pour la formule	STRING_TYPE	001
MDC_P207	OPT	Trigger event	Déclencheur d'événement	STRING_TYPE	001
MDC_P208	OPT	Domain element type	Type d'élément de domaine	STRING_TYPE	001
MDC_P209	OPT	Codomain element type	Type d'élément de co-domaine	STRING_TYPE	001
MDC_P210	OPT	Role of the relation	Rôle de la relation	STRING_TYPE	001
MDC_P211	OPT	Segment	Segment	LIST OF STRING	001
MDC_P212	OPT	Super relation	Super-relation	ICID_STRING	001

Annex B (informative)

Description examples for the PAM

B.1 Design product

The following is a description example of the PAM for a fictive activity named “design product”. Some of the corresponding IDEF0 like graphical images are auto-generated from a prototype tool that tests the concept of the data model described in this International Standard. They are shown in Clause B.2 and Figures B.4 and B.5.

Note that in the following example, the codes of the activities (MDC_P001_5) are identical to the activity number by coincidence. This is for convenience of the data preparation, but not a requirement in the PAM.

#CLASS_ID:=MDC_C002								
#CLASS_NAME.en:= Class meta-class								
#PROPERTY_ID	MDC_P001_5	MDC_P004_1.en	MDC_P010	MDC_P014	MDC_P221			
#PROPERTY_NAME.en	Code	Preferred_name	Superclass	Applicable properties	Segment			
#DATATYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE			
#REQUIREMENT	KEY	MAND						
	A0	Design product	UNIVERSE	{CAE001, CAE002, CAE003}	Activity			
	A1	Plan product		{CAE011,CAE012, CAE013,CAE014, CAE015}	Activity			
	A2	Do specification		{CAE021,CAE022, CAE023,CAE024, CAE025}	Activity			
	A3	Review spec		{CAE031,CAE032, CAE033,CAE034, CAE035}	Activity			
	A4	Authorize design		{CAE041,CAE042, CAE043 }	Activity			
	A11	Clarify requirement			Activity			
	A12	Set required specification			Activity			
	A13	Schedule product design			Activity			
	A14	Estimate rough cost			Activity			
	A15	Set product concept			Activity			

Figure B.1 – Class meta-class example of the PAM for “design product” activity (1 of 2)

#PROPERTY_ID	MDC_P001_5	MDC_P004_1.en	MDC_P010	MDC_P014	MDC_P221
#PROPERTY_NAME.en	Code	Prefered_name	Superclass	Applicable properties	Segment
#DATATYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE
#REQUIREMENT	KEY	MAND			
	A21	Plan specification			Activity
	A22	Gather data for spec			Activity
	A23	Set spec			Activity
	A24	Check errors			Activity
	A25	Provide resource			Activity
	A31	Plan review			Activity
	A32	Show spec for review			Activity
	A33	Exchange comments			Activity
	A34	Advisory comments			Activity
	A35	Compile report			Activity
	A41	Request authorization			Activity
	A42	Authorize specification			Activity
	A43	Publish design			Activity

Figure B.1 (2 of 2)

#CLASS_ID:=MDC_003				
#CLASS_NAME.en:= Property meta-class				
#PROPERTY_ID	MDC_P001_6	MDC_P004_1.en	MDC_P022	
#PROPERTY_NAME.en	Code	Preferred name	Data type	
#DATATYPE	STRING_TYPE	TRANSLATABLE_STRING	STRING_TYPE	
#REQUIREMENT	KEY	MAND	MAND	
	CAE001	Plan product	CLASS_REFERENCE_TYPE(A1)	
	CAE002	Do specification	CLASS_REFERENCE_TYPE(A2)	
	CAE003	Review specification	CLASS_REFERENCE_TYPE(A3)	
	CAE004	Authorize design	CLASS_REFERENCE_TYPE(A4)	
	CAE011	Clarify requirement	CLASS_REFERENCE_TYPE(A11)	
	CAE012	Set required specification	CLASS_REFERENCE_TYPE(A12)	
	CAE013	Schedule product design	CLASS_REFERENCE_TYPE(A13)	
	CAE014	Estimate rough cost	CLASS_REFERENCE_TYPE(A14)	
	CAE015	Set product concept	CLASS_REFERENCE_TYPE(A15)	

Figure B.2 – Property meta-class example of the PAM for “design product” activity (1 of 2)

#PROPERTY_ID	MDC_P001_6	MDC_P004_1.en	MDC_P022
#PROPERTY_NAME.en	Code	Preferred name	Data type
#DATATYPE	STRING_TYPE	TRANSLATABLE_STRING	STRING_TYPE
#REQUIREMENT	KEY	MAND	MAND
	CAE021	Plan specification	CLASS_REFERENCE_TYPE(A21)
	CAE022	Gather data for spec	CLASS_REFERENCE_TYPE(A22)
	CAE023	Set spec	CLASS_REFERENCE_TYPE(A23)
	CAE024	Check errors	CLASS_REFERENCE_TYPE(A24)
	CAE025	Provide resource	CLASS_REFERENCE_TYPE(A25)
	CAE031	Plan review	CLASS_REFERENCE_TYPE(A31)
	CAE032	Show spec for review	CLASS_REFERENCE_TYPE(A32)
	CAE033	Exchange comments	CLASS_REFERENCE_TYPE(A33)
	CAE034	Advisory comments	CLASS_REFERENCE_TYPE(A34)
	CAE035	Compile report	CLASS_REFERENCE_TYPE(A35)
	CAE041	Request authorization	CLASS_REFERENCE_TYPE(A41)
	CAE042	Authorize specification	CLASS_REFERENCE_TYPE(A42)
	CAE043	Publish design	CLASS_REFERENCE_TYPE(A43)

Figure B.2 (2 of 2)

#CLASS_ID:=MDC_003													
#CLASS_NAME.en:= Relation meta-class													
#PROPERTY_ID	MDC_P001_13	MDC_P004_1.en	MDC_P200	MDC_P202	MDC_P203	MDC_P210	MDC_P211						
#PROPERTY_NAME.en	Code	Preferred name	Relation type	Domain of the function	Codomain of the function	Role of relation	Segment						
#DATATYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE						
#REQUIREMENT	KEY	MAND	MAND										
	F1	Client request	FUNC	A-1	A0	Arrow	input						
	F1'	Client request	FUNC	F1	C1	Arrow	input						
	F2	Parts spec	FUNC	A-1	A0	Arrow	input						
	F2'	Parts spec	FUNC	F2	C2	Arrow	input						
	F3	Budget	FUNC	A-1	A0	Arrow	control						
	F3a	Budget	FUNC	F3	C1	Arrow	control						
	F3b	Budget	FUNC	F3	A4	Arrow	control						
	F4	International standards	FUNC	A-1	A0	Arrow	control						
	F4a	International standards	FUNC	F4	C2	Arrow	control						
	F4b	International standards	FUNC	F4	A3	Arrow	control						

Figure B.3 – Relation meta-class example of the PAM for “design product” activity (1 of 6)

#PROPERTY_ID	MDC_P001_13	MDC_P004_1.en	MDC_P200	MDC_P202	MDC_P203	MDC_P210	MDC_P211
#PROPERTY_NAME.en	Code	Preferred name	Relation type	Domain of the function	Codomain of the function	Role of relation	Segment
#DATATYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE	STRING_TYPE
#REQUIREMENT	KEY	MAND	MAND				
	F5	Design tools	FUNC	A-1	A0	Arrow	mechanics
	F5a	Design tools	FUNC	F5	C2	Arrow	mechanics
	F5b	Design tools	FUNC	F5	A3	Arrow	mechanics
	F6	Law & regulation	FUNC	A-1	A0	Arrow	control
	F6'	Law & regulation	FUNC	F6	A4	Arrow	control
	F7	Agreed spec & evaluation	FUNC	A-1	A0	Arrow	mechanics
	F7'	Agreed spec & evaluation	FUNC	F7	A4	Arrow	mechanics
	F8	Product design	FUNC	A0	A-1	Arrow	output
	F8'	Product design	FUNC	A4	F8	Arrow	output
	F11	Product idea	FUNC	C1	{F11a, F11b }	Arrow	connectivity
	F11a	Product idea	FUNC	F11	C2	Arrow	connectivity
	F11b	Product idea	FUNC	F11	A4	Arrow	input

Figure B.3 (2 of 6)