4. MODULE FOR REGISTRATION AND RECORDS MANAGEMENT

4.1 Purpose of module

The module for registration and records management («records management module») covers a number of functions, of which the most important are:

- registrating, storing and retrieving information on case documents and attachments (registry information, etc.)
- linking case documents to cases
- managing access to information on cases and documents, as well as access to documents which are stored electronically
- linking cases and associated documents with the record structure and physical records of the organization
- providing an overview of basic work-flow elements:
- administrative basis, case-responsible, executive officer
- processing deadlines
- depreciation status (cfr. arrears control)
- registering, storing and retrieving notes, activity logs, etc., associated with the decision-making process
- producing reports in connection with the recordkeeping functions of the organization, including:
- registry printouts (public registry, etc.)
- lists of senders/addressees
- arrears lists
- processing statistics

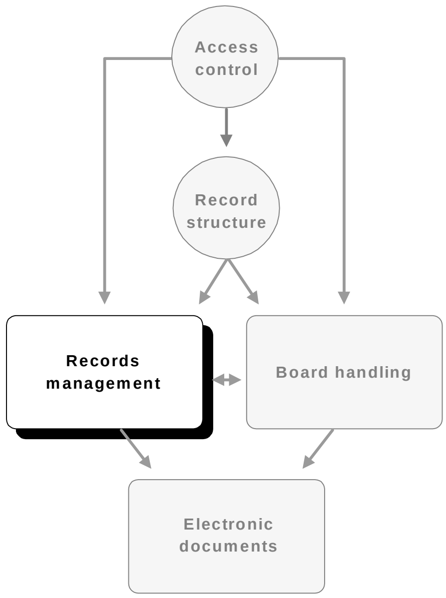

Figure 4-1 Position of records management module in Noark.

The requirements with regard to the design of the records management module are given in this chapter, except the requirements regarding reports, which are treated in separate chapter together with the other reports in the Noark system (ch. 11).

4.2 Module design

The essential tables of the records management module are *Case* and *Registry entry*. In *Case*, new cases are created and information on individual cases registered, i.e., information which is common to the documents of the case (cfr. 4.2.1 on the concept of *case*). In *Registry entry*, information on individual documents is registered. A case includes one or more registry entries, and a registry entry is always associated with one case. The relationship between *Case* and *Registry entry* is, in other words, one to many (1:M), and for a new registry entry to be created, there must always be a case to associate it with.

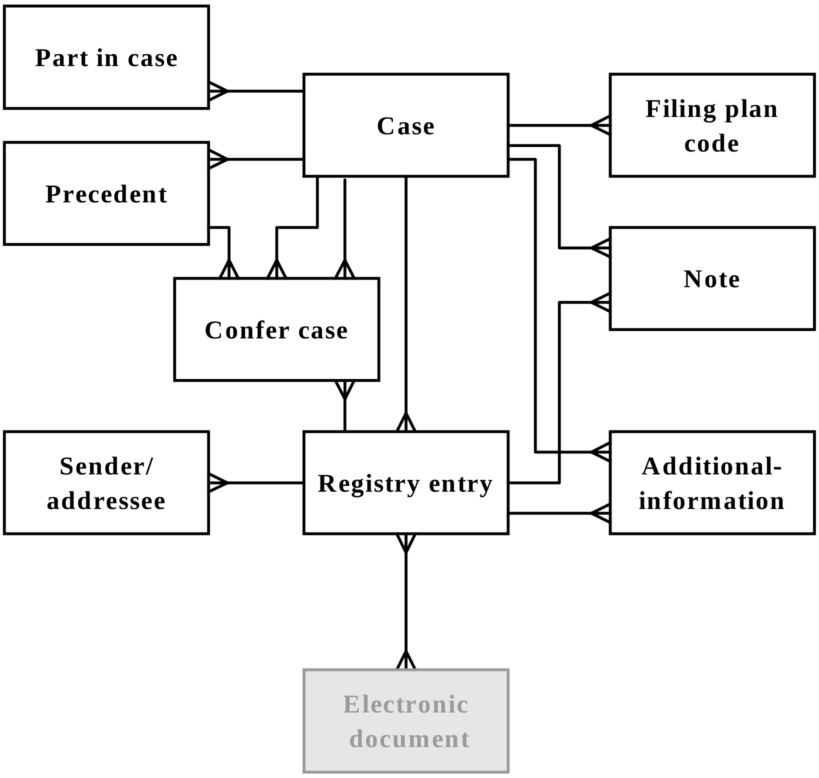

Figure 4-2: Simplified data model for records management module (see chapter 14 for complete overview)

The *Registry entry* table corresponds to *Document* in Noark-3 and Koark (compare also the concepts of "document register", "document level", "document information", etc.), and it thus contains information on the case documents. The name of the table has been changed because Noark-4 also specifies electronic storage of the documents themselves (case documents and other documents), and the concept of *document* must be reserved for these. Note also that some of the information in the document register of Noark-3 is taken out of the transaction-record table of Noark-4 and inserted in separate tables.

The following general functional requirements apply to the records management module:

K4.1	It should be possible to register a document received or produced by an organization in a registry entry. As a minimum, it should be possible to register information which is obligatory in the table *Registry entry* (see chapter 14).	O
K4.2	When created, a registry entry should always be associated with a case. One or more registry entries may be associated with a case.	O
K4.3	As common information on a case, it should as a minimum be possible to register the information which is obligatory in the table *Case* (see chapter 14).	O
K4.4	It should appear from the relevant panels that information is registered on two levels: one for the case and one for the associated registry entry. It should be possible to register on both levels in one operation.	O
K4.5	It should appear from the relevant panels how many registry entries exist for a case, and it should be possible to move between these records and view the contents of a specific record.	O

4.2.1 The concept of case

The relationship between *Case* and *Registry entry* must be considered in view of the concept of case, which has always been essential in the Noark standard. The concept of *case* is based on a similar use of the concept in the Public Administration Act and the Freedom of Information Act. None of these acts defines the concept; they use it in an abstract way, partly to denote a *case that is being dealt with or processed* (based on an external inquiry or internal initiative), partly about the *processing itself*. In a concrete sense, it applies to the case documents, registrations, notes, etc., which arise during and/or are part of the processing.

In Noark, a case is something concrete, consisting of one or more registry entries and their associated documents linked together under a common identity. The documents of a case deal with a matter that is being processed, and they represent (or "constitute") the processing of this matter.

It may occasionally be difficult to decide what is a matter for processing, and thus which documents belong to one case. There is no universally valid definition for this, but it is usually quite easy to decide. If, for instance, there is an inquiry which is to lead to a decision through a decision process, it is usually appropriate to define this as one case. The most common example of this is a case which only requires a simple answer. However, there may also be comprehensive and complex cases of this kind, e.g., employment cases, licensing cases (applications for licenses) or construction cases (applications for construction licence).

It must be stressed that the case concept of Noark is not primarily a question of what is correct and what is incorrect, but rather of how it can be used in an appropriate way. Registering cases and case information is primarily a tool for grouping case documents which belong together, under a common identity, which facilitates clarity and retrieval for both electronic records and physical records. The use of a separate case level replaces the traditional linking that was used with manual registration.

In Noark-4, it is also possible to associate cases with each other in a higher-level group *(project)*, and to establish sub-groups of registry entries (documents) within a case *(division of cases)*. This is described in more detail in paragrapgh 4.2.3.

4.2.2 Identifying case and registry entry

The identification of cases and registry entries is among the most fundamental elements in a Noark system. The following requirements apply:

K4.6	*A case* is identified through its *case number*. The case number consists of a year (see the attribute C*ase year*) followed by a serial number (sequence number of case) within that year (cfr. the attribute *Sequence number of case*).	O
K4.7	The case number should be displayed to the user as yy/######, where yy are the two last digits of the year and ###### is the sequence number, displayed with up to six digits, but omitting preceeding zeros.	A
K4.8	It should not be possible to delete a registered case or to modify a case number.	O
K4.9	A *registry entry* is uniquely identified through its *serial number*. The serial number consists of the year (cfr. the attribute *Record year*) and a sequence number within that year (the attribute *Sequence number of registry entry*).	O

The serial number is primarily an internal identification number for the system, but it may also be useful to the user, especially in a presentation of chronological document lists, be they on screen or in a paper printout.

K4.10	The serial number should be displayed to the user as #######/yy, where yy are the last two digits of the year and ####### is the sequence number, displayed as up to seven digits, but omitting preceeding zeros.	A
K4.11	It should not be possible to delete a registered registry entry or to modify its serial number (which is the internal identification of the system).	O
K4.12	It should also be possible to identify a registry entry through its *document number*, which is a serial number within the individual case (4 digits).	O

The document number is the main number for the users' identification of a registry entry. Complete identification of a registry entry involves a combination of case and document number (usually called "case and doc number").

K4.13	Case and document number should be displayed as yy/######-dddd, where yy/###### is the case number as specified above and dddd is the document number of the registry entry within the case, displayed as up to four digits, but omitting preceeding zeros.	A

The case and doc number of a registry entry may be modified if a case is split up and/or records moved to another case. This is discussed in more detail in paragraph 4.2.4.

4.2.3 References and main structures in terms of case and registry entry

Individual *cases* are linked to the record structure of an organization through references to *registry management unit* and *records section*, as well as through *coding according to filing plan*. The latter involves assigning one or more *order values* or *filing plan codes* (subject codes and/or object codes) to a case document. The coding links together cases which belong together thematically. In paper-based recordkeeping, the filing plan code is also the address of the folder where the case is stored physically. The record structure is described in chapter 7.

K4.14	When a case is created, the attributes *Registry management unit* and *Records section* should be filled in automatically. The values are obtained from the roles the assigned to the user, cfr. the table *Person/Role*. The user should be able to change the values.	O
K4.15	In the basic version of Noark-4, it should, as a minimum, be possible to classify individual cases using two order values (file codes). The file codes should be subject codes and/or object codes, and it should be possible to rank them as primary and secondary codes. The organization should be able to choose filing plan code according to the standard key of the state administration (and other keys which follow the same principles) as well as the municipal K-code system.	O
K4.16	In the enhanced version of Noark-4, it should be possible to classify a case with an arbitrary number of order values (file codes) – subject codes and/or object codes. It should be possible to rank the file codes (primary, secondary, tertiary, etc.), but it should also be possible to register file codes without rank. They will then function as references to other subjects or objects to which the case belongs.	O1

A case is assigned administrative basis for processing through references to *Case-responsible unit* and *Case-responsible (person).* The layout of the administrative structure belongs to the record-structure module and is described inchapter 7.

K4.17	It should be possible to associate a case with the administrative structure of an organization by filling in the attributes *Case-responsible unit* and *Case-responsible* (person - initials). The system should only allow values which have have already been registered in the system - see chapter 8. If the initials of the person responsible for the case are unique across administrative units, then *Case-responsible unit* should be filled in automatically as the initials of the person responsible for the case are registered.	O

Several cases may be grouped together in a *project.* A project is thus on a higher logical level than a case, but the project is only a shared category belonging to the cases, and it is used when it is appropriate to group together cases with something in common, such as an actual project. By searching for projects, it is possible to retrieve the cases which the project consists of.

K4.18	It should be possible to group together cases in a *project.* A project is a shared category which may be used to search for cases which are	O

	grouped together.	

A case may also refer to one or more other cases, or to documents within cases, without being defined as belonging to a project. Such references are likely to be used in situations where a case contains information which is relevant to one or more other cases, as a kind of cross-reference. (See also chapter 4.6 on changes in the cross-reference field as compared to Noark-3 and Koark).

K4.19	In the basic version, it should be possible to refer from one case to another.	O
K4.20	In the enhanced version, it should be possible to register references from one case to one or more other cases, and to one or more individual documents (registry entries) within a case. The table *Confer case* is used for this purpose.	O1

A *registry entry* is directly linked to a case and thereby the record structure, administrative basis and, possibly, a project and other cases. In Noark-4, functions are specified for grouping registry entries in several sub-groups *(case sections)* within a case. Such *division of cases* is primarily adapted for categorization of various types of documents within a case, which may, for instance, be used to specify groups of documents which shed light on different aspects of the case. Case sections may also in some situations be stored separately - see below concerning procedures. It must, however, be pointed out that the division of cases does not directly concern the referencing from registry entry to case, nor the numbering in the form of case and document number. Division of cases is further described in chapter 7 concerning record structure.

4.2.4 Splitting up and combining cases; moving documents

Noark-4 specifies functionality for splitting a case into several cases, and for combining several cases in one. In practice, this means that one or more registry entries are moved from one case to another, possibly after one or more new cases have been created.

The need to *split up* a case may arise as a consequence of erroneous registrations (a document may, for instance, be registered as a new document in the wrong case) or because the case develops in several directions. The split is is effectuated by removing one or more registry entries from the case in question and associating it/them with another case (possibly after creating a new case). Technically, the move is carried out by changing the case number for the registry entries in question and giving them new serial numbers (chronological, in ascending order) within the new case. Registry entries which remain with the old case, may keep their existing serial numbers, or renumbering may be carried out. If the existing document numbers are kept, the system must be able to handle holes in the document-number sequence within a case.

The need to *combine* several cases in one may arise as a consequence of erroneous registrations (for instance, a new case is created for a document which belongs to an existing case) or because one takes a different view of the case in question after some time. The combination is effectuated by moving all registry entries from one case to another, which means, technically, that the case number of those records is changed. Thus, the case from which the records are moved, loses all its registry entries and ceases to exist.

However, as said before, the case itself cannot be deleted. The system must be able to handle "empty" case numbers. In the case to which registry entries are moved, it should be possible to choose whether to sort all records chronologically (in ascending order of serial number) and assign new document numbers, or if the records that result from the combination action should come after the old ones.

Splitting up and combining cases require resources, accuracy and checking (see ch. 4.3 on procedures). For this reason, it is recommended that this be carried out only where strictly necessary for the expedient use of the system. Note also that the possibility of dividing cases (see 4.2.3 above) may in some situations be an alternative to splitting them up.

K4.21	It should be possible to move one or more registry entries from one case to another. This move involves assigning a new case number to the record(s). The case numbers must not be changed.	O
K4.22	Registry entries which are moved, should automatically be assigned new document numbers starting with the first available number in the case to which they are moved. New document numbers are assigned in ascending order according to the sequence (document numbers) the registry entries had in the case from which they were moved.	O
K4.23	Registry entries which are *not* moved, should not have their document numbers changed, unless renumbering is carried out for the records within a case (see K4.24).	O
K4.24	It should be possible to move all registry entries which are associated with one case section, in one operation.	A
K4.25	It should be possible to renumber the document numbers for all registry entries within a case. Renumbering should always include all registry entries within a case and be effectuated in one operation. The sequence should follow the ascending serial numbers of the records.	O
K4.26	If moving and/or renumbering affects references to or from the registry entries concerned (e.g., for depreciation - see paragraph 4.2.7 - or for references as described in paragraph 4.2.3), the references should be updated automatically, so that the system is consistent after moving/renumbering.	O
K4.27	The system should not allow the moving of a registry entry which depreciates or is depreciated by other records which are not moved - see depreciation linking, paragraph 4.2.7. If this is attempted, the user should receive a message informing him of which links are blocking the move operation.	O
K4.28	All moving and renumbering should be carried out using a special set of commands, and only by authorized personnel according to the rights defined in chapter 8. All moving and renumbering should be logged by the system in a clear way.	O
K4.29	During moving and renumbering, the user should be reminded to change references as necessary on paper documents in the records.	O

4.2.5 Document types

Registry entries in Noark are associated with different document types, which indicate the kind of function that the registered documents have. Permissible values for *document type* in a registry entry are specified in the table *Document type* and include the following:

- I Incoming letters
- U Outgoing letters
- N Internal documents (memos, reports, etc.) which require following-up and depreciation in the registry
- X Internal documents which do not require following-up or depreciation
- S Case drafts and other registered documents associated with the decision-making process (minute, opinion).

Note that "internal documents" in Noark means documents which are communicated internally within a unit, or between units which register in the same Noark base, hence <u>*base-internal*</u> *documents. This does not always correspond to the concept of internal documents in the sense of the Freedom of Information Act - see paragraph 7.3.2.*

K4.30	Document types are registered in the attribute *Document type* in the registry entry. The system should only allow the values which are specified in the table *Document type*.	O

4.2.6 Sender and addressee; parts in a case

Noark-4 four has moved some essential registration information, such as *registry management unit, executive officer* and attributes for *depreciation*, from the registry entry itself to a separate table for S*ender/addressee.* This makes it possible to register several (senders and) addressees on one and the same document, and to follow up the processing separately among the individual addressees. In the basic version of Noark-4, it is still possible to let the system restrict the number of senders and addressees to one, for instance by keeping these attributes in the registry entry.

The following functionality is assumed with regard to the registration of sender and addressee for the individual document types:

K4.31	In the basic version, it should as a *minimum* be possible to register • for document type I: one sender (must be filled in) • for document type U: one addressee (must be filled in) • for document types N, X and S: one sender (must be filled in) and one addressee (must be filled in for document type N).	O
K4.32	In the enhanced version, it should be possible to register the *sender* as follows: • Document type I: it should be possible to register one or more senders (one is obligatory). It should be possible to register all but one as co-senders. • Document types U, N, X, S: there should always be only one (responsible) sender. In addition, it should be possible to register co-senders. • Specifically for document type U: the default value for sender	O1

	should be the organization itself; this should be registered automatically by the system. It should be possible to change the value.	
K4.33	In the enhanced version, it should be possible to register *addressee* as follows: • Document type I: there should always be only one (responsible) addresse. The default value should be the organization itself; this should be registered automatically by the system. It should still be possible to change the value. In addition, it should be possible to register CC addressees. • Document types U, N, X, S: it should be possible to register one or more addressees (one is obligatory for document types U and N). It should be possible to register all addressees except one as CC addressee.	O1
K4.34	For all internal senders and addressees, it should be possible to register administrative unit and executive officer (initials). The system should only allow values which have already been registered in the system - see chapter 8.	O
K4.35	If the initials of the executive officer are unique across administrative units, then the administrative unit should be filled in automatically as the initials of the executive officer are registered.	A
K4.36	It should be possible to have help indices with names and addresses of clients (address registers), and it should be possible to use these for looking-up during registration of senders, addressees and parts in a case. It should be possible to refer to the individual entries in the address register by way of short names. When the short name for a sender or addressee is registered, the system should automatically retrieve information from the address register to the registry entry (or the sender/addressee table).	O
K4.37	It should be possible to group clients in the address register and refer to such address groups through an abbreviated name and/or common identifier. It should be possible to register address groups together as addressees (or as senders) on a document. In summary panels, the system should be able to display the common identifier and the number of senders/addressees the group consists of. There should be functions for toggling between the common identifier and detailed information on individual senders/addressees.	O1

It should be possible to register persons and organizations who are parts in a case, in Noark to ensure, for instance, that they get sent documents they are entitled to in connection with the procesing.

K4.38	It should be possible to register one or more parts in a case (see the table *Part in case*.	O1
K4.39	It should be possible to enter in one operation parts in a case as addressees associated with a registry entry in the case.	O1

4.2.7 Functions for depreciation and completion of cases

Registry entries containing document type I or N are subjected to arrears control and must be depreciated in order to be kept outside the arrears list. According to the technical specification of Noark-4, the depreciation attributes no longer reside in the transaction-record table itself, but in the subordinate sender/addressee table. Separate depreciation may be effectuated for each addressee, which is particularly relevant for document type N. Depreciation is carried out in the following two ways:

- Direct depreciation: the user registers *mode of depreciation* (e.g., "T.E.", i.e., ”for notification”), and *depreciation date* is filled in automatically by the system.
- Automatic depreciation: during registration of reply document, the user specifies document number for the document(s) which is/are replied to (cfr. the attribute *Replies to document*). The system links the reply document to the document(s) replied to. When the registry status of the document is set to J (registered in the records) - see chapter 6 - the system automatically carries out depreciation of the document(s) replied to. The reply document must, of course, be registered in the same case as the document(s) replied to.

K4.40	It should be possible to depreciate a received document of document type I or N in the registry by registering *mode of depreciation*. This indicates that the document has been processed by the addressee. The system should automatically set *depreciation date* to the current date, but it should be possible to change this date.	O
K4.41	It should also be possible to depreciate one or more received documents of document type I or N by registering a reply document. The reply document must be registered in the same case as the documents for depreciation. The reply document is associated with the received document(s) through registration of the document number of the latter. This should have the following effects: • The attribute *Replies to document* on the reply document is filled in automatically with reference to the received document. The reference should be displayed as document number in the relevant panels. If more than one document is replied to, the reference is set to 0. • The attribute *Depreciated by document* on the received document(s) should be filled in automatically with reference to the reply document. As long as *registry status* for the reply document is R (see paragraph 6.2.3), the reference should be displayed in the relevant panels indicating that »Reply document is beaing created» or some similar indication. When *registry status* is changed til F, E or J, the reference should be displayed by way of document number (see K6.22). When *registry status* for the reply document is set to J, the depreciation is automatically completed through the filling-in of the following attributes and values on the received documents: • *Depreciation date*: date of reply document (from the attribute *Document date*). • *Mode of depreciation*: the value BU ("brev ut", i.e., outgoing letter) if the reply document is of type U, NN ("nytt notat", i.e., new memo) if reply document is of type N or X.	O

K4.42	In the enhanced version, it should be possible to carry out depreciation of document type N according to K4.40 - K4.41 for each individual addressee. This means that a received document may be depreciated for some addressees but not for others. Depreciation should not be registered on CC addressees.	O1
K4.43	It should be possible to depreciate documents of type I only with a document of type U. It should be possible to depreciate documents of type N with a document of type N, X or U.	O
K4.44	It should be possible to register documents of type U, N or X as preliminary replies to one or more received documents. The mode of depreciation for the received documents should be set automatically to ***. Later, when final replies are registered, the depreciation attibutes should be updated in accordance with K4.41-K4.42.	O
K4.45	Registry entries/addressees whose mode of depreciation is blank or ***, should be included in the arrears list, cfr. chapter 11.	O

Depreciation (and arrears control) is associated with individual registry entries (and addressees). It should, however, also be possible to complete an entire case by assigning the value A as *case status*. A completed case is locked for registration of new registry entries, but it should be possible to reopen the case by changing the status value. This is described in more detail in chapter 6.

K4.46	It should be possible to depreciate all non-depreciated documents in one operation by setting the *case status* to A. The mode of depreciation is then automatically set to A for all registry entries/addressees. Such collective depreciation should not be effectuated without requiring the user to confirm the operation, and the system should list the registry entries (and addressees) which will be automaticaly depreciated. Should the user choose not to depreciate all registry entries, the *case status* cannot be set to A.	O

If the case handling is paper-based, the executive officer or manager will be able to indicate on the document itself that it should be depreciated, and depreciation in the registry may be performed by the registry when the document gets there. If the case handling is based on electronic documents, this opportunity does not exist. In such cases, the executive officer/manager may carry out depreciation directly in the registry, or the registry may carry out depreciation after reply has been given. It may then be necessary to log who has carried out the depreciation, and possibly on behalf of whom. Such functions may also be useful if the recordkeeping is paper-based.

K4.47	The system should be able to log who has effectuated depreciation, and it should be possible to register that depreciation has been carried out on behalf of somebody else – see the table *Additional information.* Each organization should be able to choose whether this function is to be implemented or not.	O1

The depreciation attributes may also be used to check if outgoing letters have been replied to.

K4.48	It should also be possible to use the attributes *Depreciated by document*, *Depreciation date* and *Mode of depreciation* for document type U, to indicate that reply has been received for an outgoing letter. It should also be possible to associate a received reply (document type I) with the outgoing letter according to principles similar to those of K4.41-K4.42.	A

This kind of association between documents does not constitute depreciation in a normal sense. It is not linked with the arrears control and not subject to any kind of regulation.

4.2.8 Notes, logs and other additional information

The table called *Note* is created in order to register *notes*, and the table called *Additional information* in order to store *change logs* produced by the system as well as *activity logs* and other kinds of additional infomation associated with the processing of a case or a document.

Notes may contain any information relevant to the case/document, e.g., comments regarding the handling procedure, internal comments on the reality of the case/document, etc. Notes in electronic form may replace what is today scribbled on the documents themselves, on files, etc. This transition will be necessary when case documents no longer exist in paper form, but the opportunity may be used independently of how the case documents are stored.

K4.49	There should be functions for registering notes on cases, registry entries and documents, cfr. the table *Note*.	O2

Change logs are used when changes whose documentation is important, are made in the database. Requirements for change logs are part of the specifications. Examples of functions where change logs are required, are splitting up and combination of cases – see 4.2.4 above.

Storing *activity logs* is an issue primarily in situations where Noark is integrated in or with an electronic case handling system. Storing activity logs makes it possible to document the internal processing for posterity and improves traceability. One example of acitivity logs which are required in Noark, is the logging of who effectuates depreciation, cfr. K4.49 above. The organization itself must decide what activity logs to store.

The tables *Note* and *Additional information* are also used by the module for electronic recordkeeping, and are included in the technical description (paragraphs 14.2.30 and 14.2.31). This includes a more detailed description of what is to be logged.

4.2.9 Precedents

A decision of principal improtance is called a precedent. Precedents are normative for the processing of similar cases. If it is decided during processing that a specific case may constitute a precedent for similar cases later on, the executive officer or manager may decide that the case should be registered as a precedent. In this way, it is possible to build up a precedent register which may simplify decision-making. The table *Precedent* is created to register precedents.

Precedents are particularly important in connection with laws and regulations. They complement the regulations and contribute to secure equality in the processing of cases. For this to happen, the precedents must be known to the executive officer at the time of decision-making or preparation. In this context, a precedent register in or linked to a Noark system may be be an important working tool.

Even if it is customary to talk about precedent cases, it is usually one (or a few) of the documents which constitutes the precedent. Apart from registering the whole case, one must be able to identify the document(s) which contain(s) the precedent decision.

K4.50	It should be possible to register cases as precedent cases in a precedent register, cfr. the table *Precedent*.	O
K4.51	In the enhanced version, it should be possible to refer to one or more documents (registry entries) within a precedent case.	O1
K4.52	A precedent may to refer to cases in all parts of the Noark base.	O
K4.53	In the enhanced version, a precedent may also refer to cases which are not part of the base. It should, in such situations, be possible to include a text which describes the reference as exactly as possible.	O1

4.2.10 Disposal and preservation

Administrative bodies should have rules for the disposal of cases which are not to be preserved for posterity. Disposal means that records material is removed and destructed after having been kept in the records. In the context of Noark, this applies only to (case) documents. Records information and other information registered in Noark is usually not disposed of. Disposal must be authorized by the National Archivist through general or specific rules.

For paper documents, disposal means that the documents are physically taken out of the registry and destructed. In the context of electronic recordkeeping, disposal should as a minimum involve breaking the link between the registry entry and the document, so that the document is no longer available from the recordkeeping system, and so that it is not included during export from the records database. However, when documents contain sensitive information (according to professional secrecy, grading, etc.), it is necessary to follow the rules which apply to physical destruction of such documents.

In a disposal plan, disposal provisions (see *Disposal code and Preservation time*) may in some cases be associated with a specific order value (file code). Default values for disposal may therefore be specified in the table *Order value*, cfr. paragraph 14.4.9. The disposal code may for instance be K for "kasseres" (to be disposed of) or G for "gjennomgås" (to be reviewed) with a view to appraisal. In addition to the defined default values, other values may be specified. In the attribute *Preservation time* is specified the number of years it should take from the case is completed until it is disposed of. When a case is completed, the system should automatically calculate the time for disposal or appraisal, cfr. the attribute *Year of disposal.*

In the context of electronic recordkeeping, disposal of cases where the year of disposal has been specified, may be carried out in an automated operation. A summary of cases for disposal may be obtained by extracting a disposal list - see paragraph 11.3.7.

In the context of physical records, cases for disposal must be retrieved manually. As cases are removed from the records, the disposal code in the Noark base may be updated to U for "utført" (done). It is also possible to use the automated disposal function, but if this is done, it is recomended that the cases first be physically removed from the records.

For deletion of different versions of an electronic document, see chapter 6, K6.30–K6.33.

K4.54	It should be possible to register disposal code on a case, cfr. the attribute *Disposal code*. Permissible values should be registered in the table *Disposal code* or in a similar manner in advance. The values which are specified in Noark-4, are default values, but other values may also be entered.	O
K4.55	When the disposal code is registered for a case, it should be mandatory to specify the number of years it should take before the case may/should be disposed of or appraised with a view to disposal, cfr. the attribute *Preservation time*.	O
K4.56	It should be possible to configure the automatic filling in of *Disposal code* and *Preservation time* based on the file code (cfr. the disposal attributes in the table *Order value*).	A
K4.57	When a case is completed (*case status* set to A, cfr. chapter 6), *Year of disposal* is filled in automatically by the system. Its value is calculated by adding the year from the attribute *Last document date* to the number of years specified in *Preservation time*. In the event of a case being reopened, *Year of disposal* should be reset.	O
K4.58	There should be functions for searching for cases which have reached the time for disposal or appraisal, and possibly for effectuating the disposal.	O
K4.59	Disposal should be carried out as a semi-automated (the user being prompted for each individual case) or fully automated process. Only cases with a year in the attribute *Year of disposal*, i.e., completed cases, may be disposed of. (The report *List of disposal* lists the cases for disposal.) When disposal is effectuated, disposal code is changed to U and *Year of disposal* locked for changes.	O1
K4.60	Only authorized personnel (role 1 - AR1, cfr. K8.13) is allowed to effectuate the automated disposal function.	O1
K4.61	It should not be possible to dispose of a precedent case, even if it has been indicated that it is obsolete. If precedent has been indicated, the only permissible disposal code is B ("bevares", i.e., to be preserved) or blank.	O
K4.62	Disposal in the context of electronic recordkeeping should lead to the link (relation) being broken between the registry entries of the disposed case and its associated documents. Any links between such documents and *other* registry entries should be maintaied. If there are other links to such documents in Noark or an associated system, it should not be possible to delete the documents from the electronic document store. If no such links exist, it should be possible to delete the document.	O2

4.2.11 Automated functions

Some automated functions, such as assigning case number, associating case with records section, etc., are specified in various places in part I of this report. In addition, the following requirements apply with regard to automated functions (note that requirements with regard to automation are also specified in connection with the individual attributes in part 2, chapter 14):

K4.63	The following attributes should be filled in automatically and, if appropriate, updated automatically by the system: • *The number of registry entries* (table: *Case*) should be updated when a new record is registered. Not to be modified by the user. • *Last document date* (table: *Case*) should be updated with record date when a new record is registered. Not to be modified by the user. • *Case date* (table: *Case*) should be filled in with the current date at the time of case creation. May be modified by the user. • *Record date* (table: *Registry entry*) should be filled in with the current date at the time of record creation. May be modified by the user.	O
K4.64	The following attributes are obligatory, i.e., they must be filled in in order to store a case or a registry entry with its associated sender/addressee and any electronic documents: • Case title (table: *Case*) • Content description (table: *Registry entry*) • Sender/addressee • Authority to exempt from public access (all tables) – only if the registered access code is different from XX. • Document title (table: *Document description*) – only if the document is stored electronically.	O
K4.65	When a new registry entry is created, it should be possible to fill in the attributes by copying another record in the same case. The user should be able to modify the information.	O1
K4.66	When a registry entry is registered as a reply to a document (see K4.41), the following information should be retrieved automatically from the record to the received document: • Content description • The sender of the received document should be copied to addressee on the reply document • Document type should be set to U if the received one has type I; document type should be copied if it is N or X. The user should be able to modify the information.	O1
K4.67	If K4.48 is implemented, the system should offer functionality corresponding to that of K4.66 for incoming documents which reply to outgoing ones.	A
K4.68	It should be possible to copy *Case title* to *Content description* in a registry entry.	O
K4.69	The content description of a registry entry should, as default, be copied automatically to *Document title* in *Document description* (for electronic documents). The user should be able to modify the document title.	O2

4.2.12 Other functions

In addition to the functionality described above, the records management of Noark-4 should include the following:

- *Registering maturity and processing deadlines:* Maturity normally means the deadline the sender has indicated on a received dicument (document types I and N). Processing deadline is the deadline which the addressee (recipient) himself imposes. This may correspond to the maturity date or be independent of it. Thus, in cases involving several adressees (document type N), each addressee may impose his or her deadline.

K4.70	For document types I and N, it should be possible to register maturity date, cfr. the attribute *Maturity date* in the table *Registry entry*.	O
K4.71	In the enhanced version, the individual addressee should be able to register an internal processing deadline. It should be possible to copy this deadline from *Maturity date,* if it has been indicated, or to register it independently of this.	O1

- *Registering (re)activation date:* Activation date means a date fixed by the executive officer or manager for having the whole case forwarded from the registry. With paper-based recordkeeping, this should be combined with procedures where the registry retrieves any NB cases and makes sure they forwarded to the manager/executive officer. In the context of electronic recordkeeping, the most natural procedure would be for the executive officer to search for the case, or having a system that automatically forwards it through an internal system for document flow/case handling.

K4.72	It should be possible to register (re)activation date on cases in the attribute *NB.*	O
K4.73	If the recordkeeping system is integrated with a case handling system, the system should be able to search for NB cases with a given date automatically and log a message with reference to the executive officer.	S1

- *Registering loans:* It should be possible to lend out an entire case (file with all documents associated with the case) or single documents including any attachments. It should be possible to include the borrower and the loan date in the registration. Handing in is indicated by deleting the loan information. This function is of no interest if the case is stored electronically.

K4.74	It should be possible to register loans by filling in the attributes *Loan date* and *Lent to*. The registration might include an entire case (table: *Case*) or single documents including any attachments (table: *Registry entry*).	O

- *Registering dispatch and dispatch method:* For all types of documents, it should be possible to register the dispatch method for the various addressees, i.e., standard mail, fax, e-mail, etc. If the records are electronic, it must also be possible to register the fact that a dispatch (document and any attachments) has been effectuated, including date/time and sender.

K4.75	For each recipient of a document, it should be possible to register dispatch method (mail, fax, e-mail, etc.) – cfr. the attribute *Dispatch method* in *Sender/addressee*.	O1
K4.76	For each recipient of a document of type U, N, X or S, it should be possible to register the dispatch date and the initials of the person who effectuated the dispatch.	O2

Functionality related to dispatch by means of e-mail is described in more detail in chapter 10.1.

4.2.13 Linking up with other modules

The records management module is vital to the recordkeeping system and closely linked up with the other modules. The linking is as follows:

- *Access-control module:* All use of the system is subject to user management and access control as specified in the records management module - see chapter 8. This includes the users' right to register or correct as well as to search for and read information in the records management module. Some of these rights are associated with various stages in the decision-making process and managed by attributes in the tables *Case* and *Registry entry*. These are described in chapter 6.

- *Record structure module:* All the cases in the records management module are associated with an administrative unit, a registry management unit, a records section and entity of records. These structures are defined in the record structure module and described in chapter 7.

- *Electronic records:* Individual registry entries play a vital role as gateways to the stored case documents, be they hardcopies (on paper) or stored electronically. However, electronic storage and handling of case documents are maintained by the module for electronic recordkeeping, and this is also where the tables reside which maintain the relationships between registry entries and case documents, cfr. chapter 5.

- *Board-handling module:* When collegiate bodies (advisory and governing boards, etc.) take part in the decision-making process, the board-handling module is used to keep track of the document flow and work flow. However, the processing usually involves cases which arise and are completed in the records management module, and a considerable proportion of the board documents are registeres in the registry. Registered board documents such as drafts, minutes and opinions have special functions in the decision-making process, but as far as recordkeeping is concerned, they are on the same level as other registered documents. All the details concerning board-handling are described in chapter 9.

4.3 Procedure requirements

The basic procedures for entry into records and records management are described in the regulations[1] and established through 10-12 years of computer-based recordkeeping using Noark systems. It is therefore not considered necessary to present a complete description of such procedures here.

4.3.1 Prodecures related to special functions

Procedures related to new functions in Noark-4 are described in other chapters, e.g., in connection with the other modules. All of these are, however, closely related to the records management module:

- Chapter 6 describes the work flow and document flow of the decision-making process, both for paper-based and electronic recordkeeping and decision-making environments.
- Chapter 7 describes procedures related to the record structure of the organization.
- Chapter 8 describes procedures for access control.
- Chapter 5 describes procedure changes in connection with the transition to electronic recordkeeping.
- Chapter 9 describes procedures related to political decision-making and document handling in connection with this.
- Chapter 10 describes procedures related to the use of integrated e-mail and digital signatures.

In addition, it is worth paying attention to the following functions, which require set procedures:

- When registry entries are moved from one case to another (split or combined – see paragraph 4.2.4 above) and when documents within a case are renumbered, one must be careful to update the numbers on the associated paper documents in accordance with the changes that are made in the base.
- When sender and addressee are registered on documents, clearly defined procedures should be followed, even if Noark-4 provides for much flexibility. As a rule of thumb, the addressee for incoming letters (type I) should be the organization itself (system default). The internal distribution is maintained by registering administrative unit and case-responsible/executive officer. Likewise, the sender for outgoing letters (type U) should normally be the organization itself (system default). The need to register several addressees and senders is mainly restricted to the following:
- Several recipients of an outgoing letter (type U)
- Several recipients of an internal document (types N and X)
- Internal and external CC addressees of all types of documents
- Several senders on a received document (type I, N or X) - preseumably happens rarely
- Procedures for handling internal documents are closely related to the record structure, and are dealt with in chapter 7 (7.3.2)

[1] Arkivforskriften (the Archives Regulation), cfr. kgl.res. (royal decree) 11.12.1998.

4.4 Essential tables in the module

For a complete list of tables and attributes, see chapter 14.2.

Table name	**Text**
Case	Contains information which identifies a case (case number and title) as well as information common to all documents within the case.
Registry entry	Contains information which describes the documents in the individual cases (cfr. the document record of Noark-3) as well as information regarding processing and status.
Sender/addressee	Makes it possible to register more than one sender/addressee for a document. Contains name, address, etc., of sender/addressee, as well as information on executive officer, decpreciation and dispatch. This provides for varied follow-up of the decision-making process for various recipients (particularly important for internal documents).
Notes	Contains notes (remarks) on a case, a registry entry or a version of a case document. This could be a comment on the processing itself or comments or corrections regarding the contents of the case or the document.
Additional information	Contains various kinds of additional information on a case, a registry entry or a version of a case document. The main types of additional information are: • Processing information (activity log) • Change logs, etc.
Filing plan code	Contains assigned values in terms of subjet (topic) codes or object codes. The code is specified as primary, secondary, tertiary, etc., if it is used as an additional filing plan code.
Part in case	Contains information (name, address, etc.) which describes the internal unit(s), external organization(s) or private person(s) who is/are part(s) in the case.
Confer case	Contains references from one case to other cases, single documents within these cases and cases in the precedent register. Is used to link references between cases.
Precedent	Contains information on precedent cases.

4.5 Changes from Noark-3 and Koark

Below is given a brief summary of changes in the records management module as compared to Noark-3 and Koark. A complete specification of all differences is given in chapter 16, which also discusses compatibility and conversion opportunities.

Basic version (requirement type O):

- The concept of *records management* is new. The *module for registration and records management* in Noark-4 corresponds to the *registry section* of Koark, while the entire Noark-3 is defined as a computer-based *system for entry into records.*
- The concept of *Registry entry* (table) replaces *Document (register).*
- The concept of *Content description* (attribute) replaces *Document description*.
- The concept of *Access code* (attribute) replaces *Grading code*, cfr. ch. 8.
- The concept of *Processing deadline* (attribute) is introduced, as well as *Maturity*.
- To case number, document number and serial number has been added an extra digit.
- The record structures has been modified somewhat. This has consequences for registration of records references (entity of records, records section, etc.). The record structure permits several types of filing plans (state plans, K codes, etc.). This is in accordance with Koark, but in enhanced versions the technical solutions are modified, as discussed below. The new record structure is described under the record structure module (chapter 7).
- The *Cross reference* field has been omitted. The functions covered by this field have been replaced by two new attributes: *Project* and *Confer case.* See also the previous indent on record structure, as well as below on file codes in the enhanced version.
- The functionality related to *disposal* and *precedent* has been enhanced.
- The screening codes under *grading* have been replaced by functions for checking off the information to be screened off, cfr. chapter 8.
- The attribute *Case status* has been kept from Koark, and is not found in Noark-3.
- The *Doc no.* field under *Depreciation* is replaced by two attributes: *Depreciated by document* and *Replies to document*.

Enhanced version (requirement type O1):

- The concept of *Registry status* (attribute) replaces *Document status* (Koark).
- There are attributes for registering an arbitrary number of file codes. These may be ordered hierarchically (primary, secondary, tertiary code, etc.) or be independent (e.g., additional codes), cfr. record structure in ch. 7.
- Registering administrative units – see *Case-responsible* and *Executive officer* – is much more flexible, see chapter 7.
- Some attributes have been omitted from *Registry entry* and moved to a separate table, S*ender/addressee.* This provides for registering several sender/addressees on one and the same document, and for separate follow-up of internal documents with individual recipients.
- A new table has been created for registering parts in a case - *Case part*.
- A new table has been created for storing change logs and other additional information - *Additional information.*

Electronic records (requirement type O2):

- The fields for references to electronic documents have been omitted and replaced by more comprehensive functionality for electronic recordkeeping - see chapter 5.

- A new table has been created for registering notes - *Notes*.

Recommended functionality (requirement type A):

- A case may be divided into several *case sections*, as discussed in chapter 7.