Public Government, A Domain Analysis

Dines Bjørner[1,2], Chen XiaoYi[2]

1: Computer Science and Engineering
Informatics and Mathematical Modelling
Technical University of Denmark
DK-28000 Kgs.Lyngby
Denmark

2: Graduate School of Information Science
Japan Advanced Institute of Science & Technology
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa 923-1292
Japan

bjorner@gmail.com, xychen@jaist.ac.jp

June 14, 2006. Compiled June 27, 2006

Abstract

Contents

1 An Informal View of Public Government

Public government, in this seminar, consists of the **lawmakers:** parliament. the **law enforcers:** central and local government. and the **law interpreters:** judiciary system. **Citizens** interact with all three branches of government.

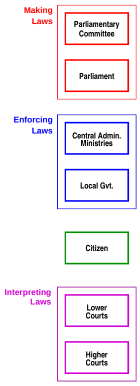

Figure 1: The Three Branches of Government

Citizens through the process of debate provoke their parliament to discuss societal problems. A parliament committee discusses a specific societal problem. Their deliberations are "sent" as a law proposal to parliamment which debates the issue and passes some law.

The law is passed on to an appropriate ministry (of the central government) and that ministry formulates basic rules & regulations for how local governments shall administrate uses of the law. The local governments makes provisions for handling the law locally.

The citizen is either contacted by the local government and asked to report on some issue (tax, traffic violation, or other), and the citizen replies. or the citizen contacts the local government in order to apply for something (passport, pension benefits, ot other) And the local government replies.

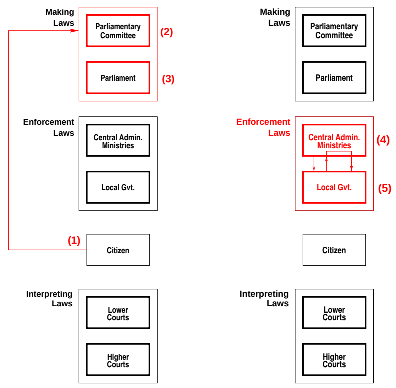

Figure 2: From citizens to lawmakers and lawmaking

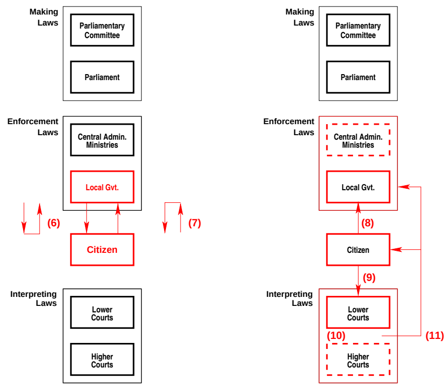

Figure 3: Citizens between local government and the judiciary

The citizen is either accepts the dicision of local government, or the citizen does not accept the dicision, and complains to the courts. The "due process of law" takes place. Eventually the judiciary system hands down a decision either in favour of the citizen, or in favour of the governement, or both!

2 Flow of Documents in Public Administration

2.1 Between Citizens and Lawmakers

Citizens may direct a problem petition to parliament — in the form of a document signed by many citizens. Parliament decides to "do something" (or not to do anything) about the problem. A response document is produced. A designated parliament committee requests an appropriate ministry to prepare some "background" document. The parliament committee passes discussion and a law proposal documents to parliament.

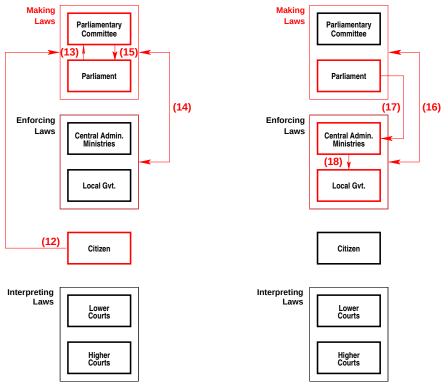

Figure 4:

2.2 Between Lawmakers and Ministries

Parliament requests further "background" documents from the central administration, and receives these. Parliament debates the law proposal and passes a law, which, as a document is sent to the appropriate ministry for further handling — and otherwise published in the law gazette. The ministry and its departments, i.e., the central administration, formulates procedures for the enforcement of the law and sends these, as documents to local administrations.

2.3 Between Citizens and Local Government

A citizen applies for some permission, that is, an application document is sent to a local administration — or a citizen breaks the law (symbolised with the virtual arrow from citizen to a local authority). The local administration sends a receipt (a citation) document, possibly forwards further documents to be filled in, and gives a conditional date by which a decision can be expected. The citizen sends in the possibly further requested documents. The local administration communicates various documents related to the case to/from other public government offices. And finally the citizen receives a response document.

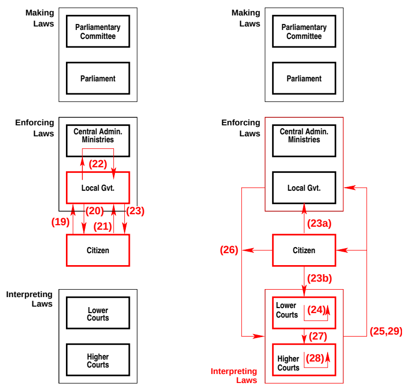

Figure 5:

2.4 Between Citizens and The Judiciary

The citizen either sends an acceptance document to the local administration, or rejects it, informing the local administration of this, and directs a complaint at the law courts. The first instance law court deliberates (i.e., documents are produced), a decision is sent to the citizen and the local admin., Either the local administration or the citizen both accepts the decision and further actions are curtailed, or at least one of them appeals the decision. Lower court decision documents are passed on to a higher court. (28.–29.) And steps (24–25) are repeated till a final decision is passed.

2.5 Summary of Documents

Citizen petition Parliament response Background briefing Subcommittee discussion and law proposal Further background briefing Law and record of paliament debate Handling procedures and forms Citizen application (or "breakin law event") Local authority reply (or citation) Citizen response to local authority reply Public admin. docs. handling citizen request or infringement "Final" local authority reply/decision Citizen acceptance or rejection Citizen lawsuit Lower law court handling Lower law court decision Citizen and public administration reaction Transfer of lower law court dcosuments Higher law court handling Higher law court decision (final, final)

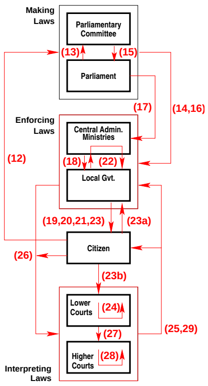

Figure 6: Summary of document flow

3 Documents — A Closer Analysis

We present a "story" of documents that is a bit different from what you may be used to. The reason is that we are building up, towards the end of this seminar, to a "story" on computers, communication and documents. Our "story" on documents is also "completely" independent of our previous "narration" of public government. Again there is a reason: towards the end of this seminar we merge the "story" on documents with the "narration" of public government into the subject of $\mathcal{E}$-Government.

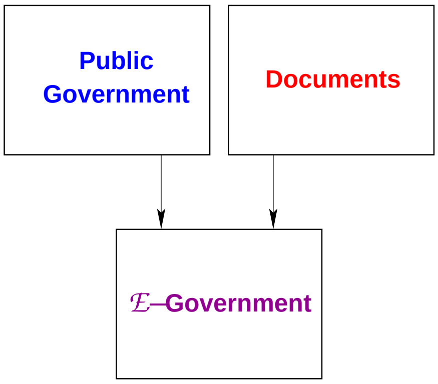

Figure 7: From public government to documents — and beyond

3.1 Overview of Document Issues

Document Operations: Documents are created, are edited, and can be read, copied, moved, can be the basis for searchees and calculations, and can be shredded — by actors. **Document Authorisation:** Actors have varying degrees of creation, editing, reading, copying, distribution and shredding authority. **Document History:** Document may reveal their creation, editing, reading, copying distribution, search & calculation history — who did what, when, where! **Document Licensing:** Unauthorised actors must be prevented from performing designated operations on documents. We achieve this indirectly by introducing and enforcing a regime of licensing based on authorisation.

3.2 Actors: Citizens and Agents

Document operations are performed by **actors.** **Actors** are either **agents** or **citizens,** An **agent** is either a person working for a branch of public government, or is that branch of public government. Of course **citizens** may be agents in some other context, that is, a **citizen** is a person interacting, not as an agent, with agents of public government.

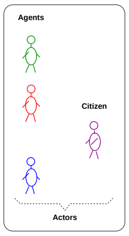

Figure 8: Actors: citizens and three kinds of agents

3.3 Document Operations

Document operations are performed by **actors.** Documents can be **created.** Documents can be **edited.** Documents can be **read!** Documents can be **copied.** Documents can be

moved. Documents can be **shredded** (not shown). An **actor** can at most be performing one operation at a time. Documents are "marked" by the **time** and **location** of the operation and by the identity of the performing **actors.** Together the **time, location** and **actor identity** forms a **unique document identification.**

Documents can be **created.** Created documents contain no substantial information other than administrative information about time and location of creation, identity of actor who created the document.

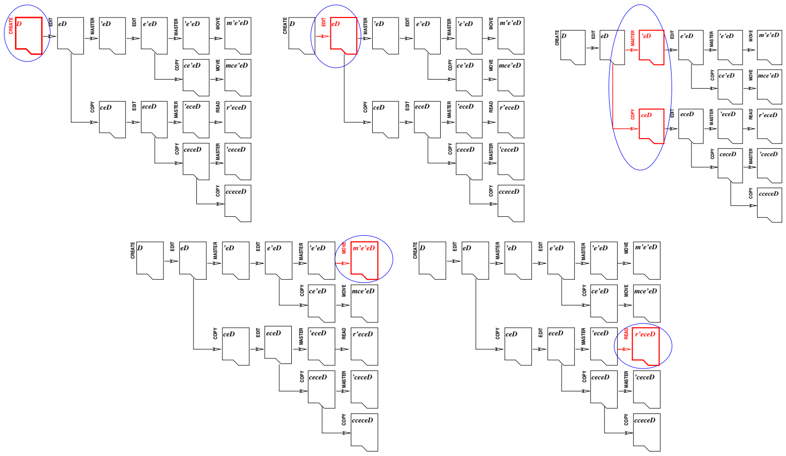

Figure 9: Five document operations

Documents can be **edited.** Edited documents are **versions** of the document on the basis of which they were edited. Edited documents contain substantial information. One can read documents while editing them! One cannot copy, move or shred documents during editing. An edited document is different from the document input. The difference amounts to the changed text, time and location of edit, and identity of editing actor.

Documents can be **copied.** As a result the **master** "goes on" to exist and a copied, **"the copy",** document is constructed. One cannot edit, move, read or shred the master or the copy documents while copying. The location and time of copying is the same for both master and copy. To **audially communicate,** i.e., to **tell** (speak about) a document (content) to other listeners is the same as **copy** ing it.

Documents can be **moved.** They are physically moved from one **location** to another distinct **location.** The document "is almost" the same before start and after end of move, only location has changed. Documents cannot be copied, edited, read or shredded while being moved. Instead of **moved** we shall sometimes use the term **distributed.**

Documents can be **read** ! One cannot edit, copy, move or shred document while only reading them. Reading leaves the document unchanged — except that it has now been read, at some time and at some location, by some actor.

3.4 Document Family and Document Versions

Document Family The structure to the right designates one document family. Every create gives rise to a document family. It may grow (copying) and shrink (shredding). **Document Version** A "just" create document has version 0. Every edit of a document creates a new version of that “same” document. All other operations leave the version attribute unchanged. (Let us not be bothered by how version numbers are generated!)

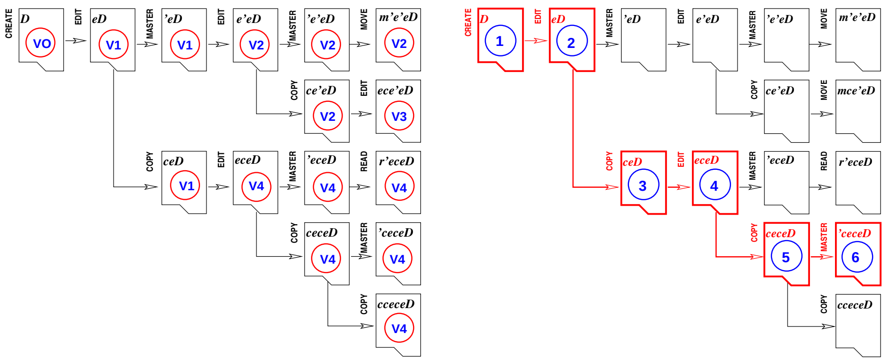

Figure 10: Document family and document versions

3.5 Document History

From a document we can, “in theory”, **trace its unique past.** In reverse order of operations: (6) Take the lower right MASTER. (5) It was most recently the BASIS for a **copy** ing. (4) Before that it was the BASIS for an earlier **copy** ing. (3) Before that it was the RESULT of a **copy** ing. (2) Before that it was an **edit** ed version (1) of a **create** d document.

(1) **Create** An actor named **Nm1** performs operation **create** at time **time1** on location **loc1.** A document is created with just about the only information you see in the lower left corner to the right:

(2) **Edit** An actor named **Nm2** performs operation **edit** at time **time2** on location **loc1** Document *D* is extended with text **text** into document *eD*.

(3) **Copy** An actor named **Nm3** performs operation **copy** (on *eD* and we focus on the ‘copy’) at time **time3** on location **loc1.** Document *eD* text **text** remains unchanged but is referred to as *ceD*.

(4) **Edit** An actor named **Nm4** performs operation **edit** at time **time4** on location **loc1.** Document *ceD* text **text** is changed into document *eceD* **text’.**

(5) **Copy** An actor named **Nm5** performs operation **copy** (on *eceD* and we focus on the ‘copy’ *ceceD*) at time **time5** on location **loc1.** Document *ceceD* text **text’** remains unchanged.

(6) **Master** An actor named **Nm6** performs operation **copy** (on *ceceD* and we focus on the ‘master’ *′ceceD*) at time **time6** on location **loc1** Document *′ceceD* text **text’** remains unchanged.

3.6 Document Authorisation

3.6.1 Rationale for Authorisation

We explain the need for introducing a concept that we shall call ‘authorisation’. Some documents may contain information that not all **actor** s should be aware off. For an **actor** to be allowed to perform an operation upon (create or edit or copy or read or move or shred)

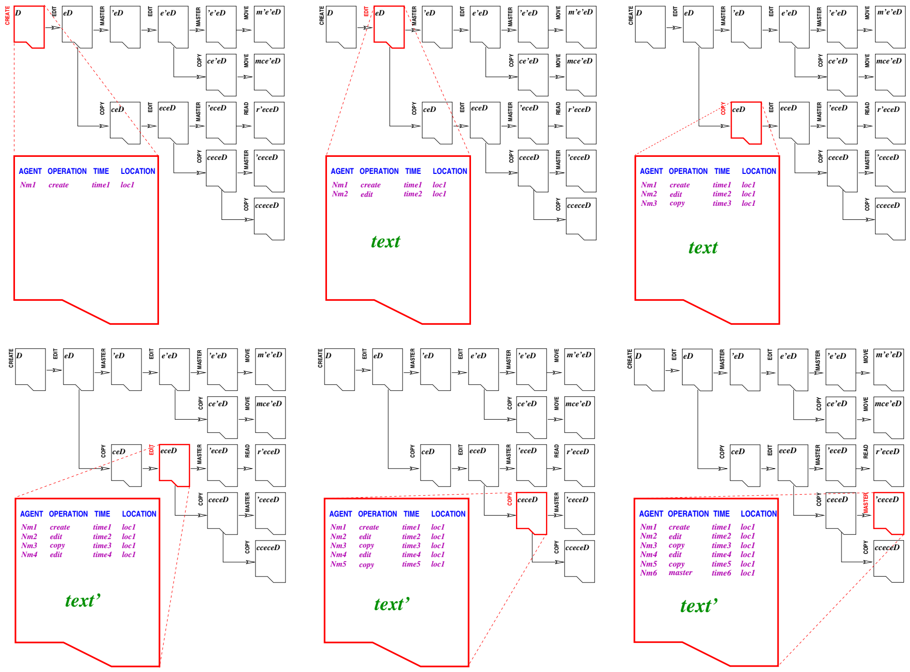

Figure 11: Document annotation

a document that **actor** must be so authorised (by some **agent**). **Authorisation** can be in the orm of a **license,** a **permit,** to perform an **operation** on some entity, as here, a **document.**

3.6.2 Authorisation of Actors

Actors can be authorised with respect to (wrt.)one specific document whose identity will then be given, or wrt. a finite, identified set of documents, or wrt. a potentially indefinite class of documents, so we have to introduce a notion of a document class. (An example class of documents could be the class of all social security application forms [based on same template]. More on this later.)

Actors can, within the designated set of such documents, be authorised wrt. which operations can be performed on these documents: **create, edit, copy, search, move, read, shred and compute.**

Actors may be authorised to **grant** document handling authorisations to (other) actors, **extend** or **limit** previously (by others or by that same actor) granted such document handling authorisations, or outright **withdraw** such document handling authorisations altogether.

The problem of how to initialise the system of document handling authorisations is an interesting one — which we may have time to come back to later.

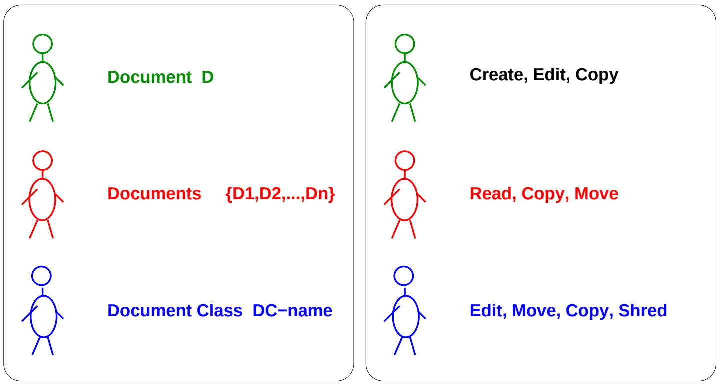

Figure 12: Document and operation authorisations

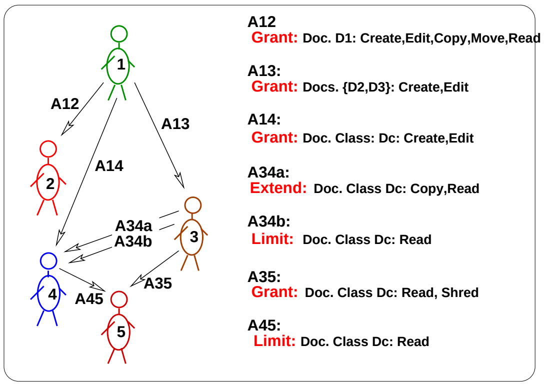

Figure 13: Granting, Extending and Limiting Licences

3.6.3 License Scripts

A license script is a named text which is issued by one actor to another. The license specifies a grant, an extension, a limitation or a withdrawal of specified document operations on specified (licensed) documents. A license can also grant permission to issue further licenses.

Example Licenses

ln1: **actor** a1 **grants operations** {op1,op2,...,opn}
to actor a2 **on documents** {d1,d2,...,dm}

actor a3 **extend license** ln2 **of actor** a4 **with operations** $\{op',op'',...,op'''\}$

actor a5 **limit license** ln3 **of actor** a6 **with operations** $\{opa,opb,...,opw\}$

actor a7 **withdraw license** ln4 **from actor** a8

actor a9 **grants licensing right**
ln5: **actor** a10 **grants operations** {opx,opy,...,opz}

to actor a11 **on documents** {da,db,...,dc}

actor a12 **withdraw license** ln6 **from actor** a13

3.7 Special Edit Operations

We have hinted at only a very rudimentary **edit** operation. One that takes only one document and basically adds, modifies or deletes **text.** More general, in fact a whole family of **edit** operations are present in everyday handling of documents: **merge** of two documents into a third, implying a **copy** ing and a **create** operation. into one of the two documents, implying a **copy** ing and a simple **edit** operation. **split** of one document into two documents implying a copy ing and two **create** operations. *Etcetera*

3.8 A Document Class Concept

3.8.1 Examples

General Public Administration Document Classes Different branches of government work on and produce different classes of document: Parliamentary committees (PC) handle **societal background problem** documents and produce **committee discussion** and **law proposal** documents.

General Public Administration Document Classes — Continued Parliament (P) handles **committee discussion** and law proposal documents and produces **parliament discussion** and law documents. ... And law courts receive **law suit** documents, deliberate over **law court proceeding** documents and issue **verdicts** (i.e., documents).

Specific Public Administration Documents

Taxation Document Classes A tax office issues **tax declaration template** documents and handles **tax declaration form** [1] documents.

Traffic Police Documents A traffic police officer handles **traffic violation citation template** documents and issues **traffic violation citation forms** (i.e., documents).

Budget and Account Documents An accountant handles **budget** documents, fills in **account template** and **account form** documents, and aggregates (calculate) over **account form** and **budget** documents to produce ...

3.8.2 Document Classes

Generic Document Classes By the generic class of documents we understand a class of documents that is indeoendent of the specific application domain. We make the following generic classification: **General, unformatted, un-structured text documents.**

[1] A form is a filled-in template document

Comment : No computations can be done on such documents. Example : Any "spur-of-the-moment" note **Formatted, semi-structured text documents.** Comment : Trivial searches can be done over such documents. Example : Most public administration documents. **Speficially formatted, template-based document.** Comment : Non-trivial compuations can be done over such documents. Example : Most application and related documents.

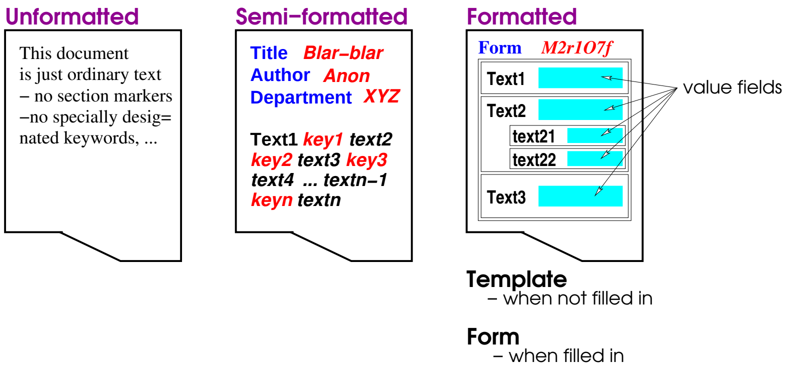

Figure 14: Indication of Document Classes

Document Classes

Specific Government Document Classes For the application-specific category of public administration we suggest to let the class of government documents be determined by their relevance to specific laws or law proposals:[2] Some specific examples:

Law proposal background doc. Law proposal discussion doc. Law. Law admin./handling doc. Law rules & reg. doc. Local admin. letter to citizen. Citizen reply to local admin. letter. Citizen complaint to law court. Law court inquiry doc. Law court decision.

3.9 Document [Cross–]References

Documents in one family of documents may cross-refer to documents in another family of documents as shown to the right. Reasons seem obvious: chronologically "later" documents were derived in the context of knowledge of the chronologically "earlier" documents which can be made aware of the of the "later" uses.

Examples of Document [Cross–]References Law proposal background doc. may refer to other laws. Law proposal discussion doc. may refer to background doc. Law usually refers to many other laws. Law admin./handling doc. usually refers to many other handling docs. and refers to the law. Law rules & reg. doc. refers to the law. Local admin. letter to

[2]— all other documents are simple administration documents found also in most other forms of administration.

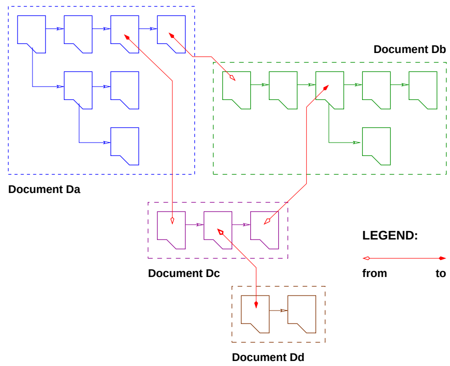

Figure 15: Cross-references between documents of four document families

citizen refers to the law. Citizen reply to local admin. letter refers to that letter. Citizen complaint to law court refers to local admin. letter and law. Law court inquiry doc. refers to citizen and local admin. letters and the law. Law court decision refers to the law and other law court decisions.

3.10 Document Computations

We can distinguish amongst different kinds of computations over documents.

Document Identifier Searches : Example : Intra- and inter–document-family tracing.

Attribute Searches : Example : Searches based on document attributes — incl. form numbers.

Trivial Document "Syntactic Content" Searches : Example : Documents related to a specific agent or topic (a citizen, a law, or other). Searches look for text parts only.

Non-trivial Document "Semantic Content" Searches : Example : Documents containing strongly formatted text parts, e.g., XML embedded and coded texts. Searches try to deduce meaning ("data mining").

Full-blown Computations : Example : Tax computations.

3.11 Summary of Document Attributes

These are the generic classes of document attributes.
document name **document version number** **document classes** (generic and special) **history trace** of operations[3] (operation, location, time, actor) **reference to relevant license** A unique document identifier can be calculated from the above.

3.12 Actor Attributes

Unique Actor Identifier **Actor Location** **Licenses** Current "Own" Licenses (i.e., work being pursued) Current Granted Licenses (i.e., work being managed) Which **Documents** (by rerence) have be **Operated** upon at **Locations** and at **Times**

[3]including computations

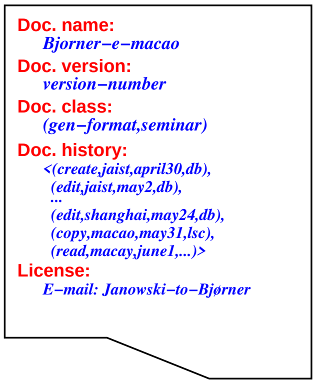

Figure 16: Document attributes

4 $\mathcal{E}^2\mathcal{G}$: $\mathcal{E}$ssential $\mathcal{E}$-$\mathcal{G}$overnment

Law-based public administration handles, like any other public or private administration, "thousands" of kinds of "zillions" of documents. Of these we single out now, we focus only on those public government documents which are **based in law,** which ultimately **refer to laws.** That is we **do not** in the following consider such kinds of documents as procurement, (ordinary) budget, accounting, personnel, etc. documents whose handling is like in any other, not law-based administration, whether public or private. **Assumption: paperless administration** !

4.1 The Meaning of $\mathcal{E}^2\mathcal{G}$

So, by $\mathcal{E}$ssential $\mathcal{E}$-$\mathcal{G}$overnment we shall thus mean: Any electronic handling of documents, that is, **planning, discussion, decision, preparation, communication, etcetera** and which are based on and results in (possibly new, possibly edited) documents **such that these documents directly or indirectly refer to laws.**

4.1.1 From Domain to $\mathcal{E}^2\mathcal{G}$ Requirements

To illustrate what "such electronic handling" might mean, we **systematically go through** the domain(s) of public government such as outlined in topic 1 and of documents, such as outlined in topic 2 in order to **decide** whether "such–and–such" agent (institution, i.e., branch of government) and citizen interactions, documents and document attributes, functionalities, "etcetera" **must** be **"made more–or–less electronic".**

4.1.2 So Here We Go: Towards $\mathcal{E}^2\mathcal{G}$

Which branches of government should be included in $\mathcal{E}^2\mathcal{G}$? One answer could be: All branches interfacing with citizens. The double–arrowed line indicates this. Within these branches we must list the relevant departments. Another answer could be: Only such and such a subset of the indicated branches. (We then omit some of the $\leftrightarrow$s.) See Fig. 17 on the facing page.

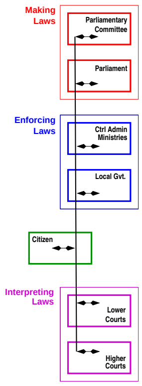

Figure 17: Branches of government included in $\mathcal{E}^2\mathcal{G}$

Which document classes should be included in $\mathcal{E}^2\mathcal{G}$? One answer could be: All the (in this example) **19** document classes implied by the **red** arrows in figure to the right. For each arrow-head we then list relevant document classes. Another answer could be: Only such and such a subset of the indicated document classes. (That is, we omit some of the $\rightarrow$ document classes.) See Fig. 18.

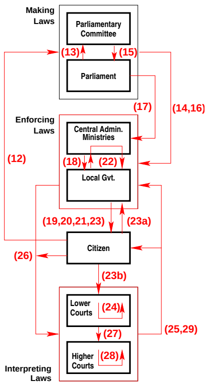

Figure 18: Document classes included in $\mathcal{E}^2\mathcal{G}$

Which authorisation on document classes should be $\mathcal{E}^2\mathcal{G}$ scripted? For each selected document class scripts of zero, one or more authorisations need be designed and their instantiation wrt. specific documents must be computerised, as must the monitoring and control of the implied document family. See Fig. 19 on the following page.

Which new operations should be $\mathcal{E}^2\mathcal{G}$ supported? For each selected document class as well as combinations of related document classes all conceivably $\mathcal{E}^2\mathcal{G}$ supported operations must be defined: All relevant **searches.** All relevant $n \geq 1$ **computations.** All **trace** functions. A new **release** operation. The **shred** operation. See Fig. 20 on the next page.

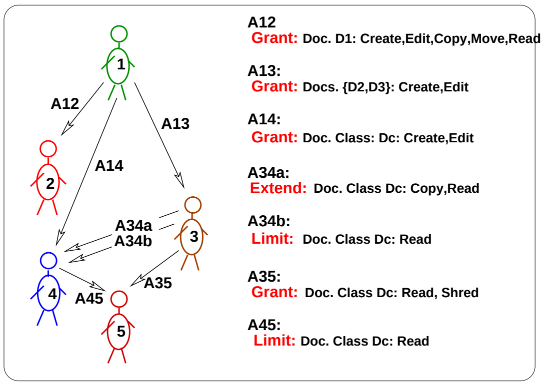

Figure 19: Document authorisations to be $\mathcal{E}^2\mathcal{G}$ scripted

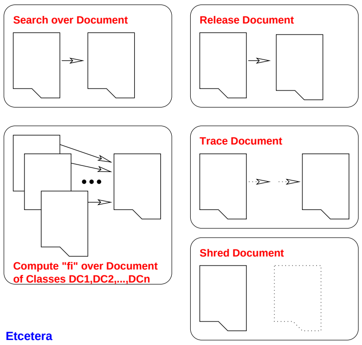

Figure 20: New operations to be $\mathcal{E}^2\mathcal{G}$ supported

More on $\mathcal{E}^2\mathcal{G}$ Document Computations (II) One class of computations have the compuation be based on **value** s of a possibly only partially **filled-in template** (= **form**). The computation would then yield, not necessarily a document, but new values that can serve as basis for further computations and possibly be fed to a repository of documents. Such computations presume design of templates (i.e., forms), and the implementation of the computations. See left part of Fig. 21 on the facing page.

Yet More on $\mathcal{E}^2\mathcal{G}$ Document Computations (III) Another class of computations collects data from several possibly distinct documents and produce an aggretated document. An example could be income and property tax returns of several tax payers aggregated into a community tax budget. The leftmost, i.e., the resulting document may be an edited (i.e., an update) version of one of the rightmost, i.e., input documents. See right part of Fig. 21 on the next page.

Transparency Implies $\mathcal{E}^2\mathcal{G}$ Traceability Given a document, say the "bottom-right-most" **$\surd$-checked** document one may wish to trace all its "ancestor" documents. Such a trace may — as indicated — lead to several traces, that is, to sequences of document

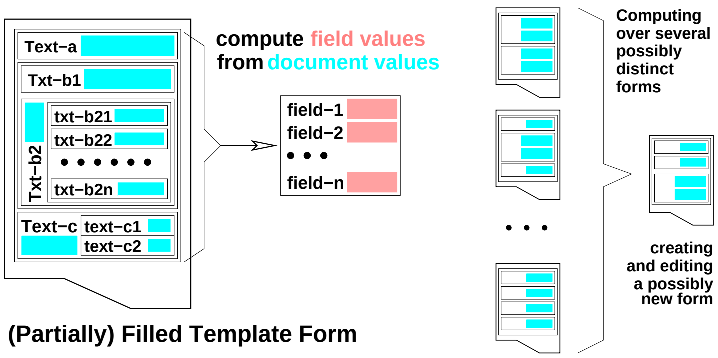

Figure 21: $\mathcal{E}^2\mathcal{G}$ document computations

references. For an agent or a citizen such **traceability** implies **transparency** of public government. See Fig. 22.

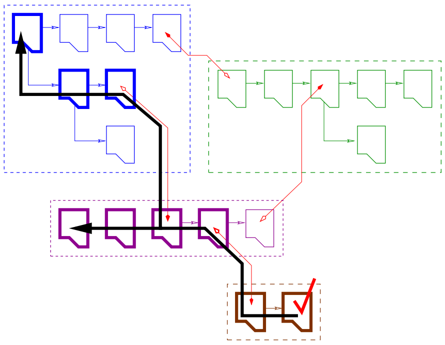

Figure 22: Traces

4.2 Summary of $\mathcal{E}^2\mathcal{G}$

4.2.1 First Summary

So $\mathcal{E}$ssential $\mathcal{E}$-$\mathcal{G}$overnment, $\mathcal{E}^2\mathcal{G}$, assumes **paper-less** administration, i.e., all electronic documents and focuses of all/most such **document** -manifested actions which are **based in law.** Such documents entail: Which **branches** of public government are **included/excluded** ? Which **document classes** are **included/excluded** ? Which **authorisations,** and which **scripts** are **mandated** ? Which **operations** on documents are **supported** ? Which form of **traceability** is **supported** ?

4.2.2 Second Summary of $\mathcal{E}^2\mathcal{G}$

Figure 23 on the next page intimates some properties of an assumed modular, i.e., parameterised $\mathcal{E}^2\mathcal{G}$ server. The dotted rounded-edge left box stands for such a server. You may think of one server per actor. The figure shows an agent actor. For the citizen actor one omits the license and document interface. The syntax of the up and down pointing license and document interfaces is a simplification of the possibility of n such sets of license and document interfaces, one pair to each "other" agent being interfaced. The syntax of the leftward license and document interface to citizens is a simplification of the possibility of m such sets of license and document interfaces, one pair to each citizen being interfaced.

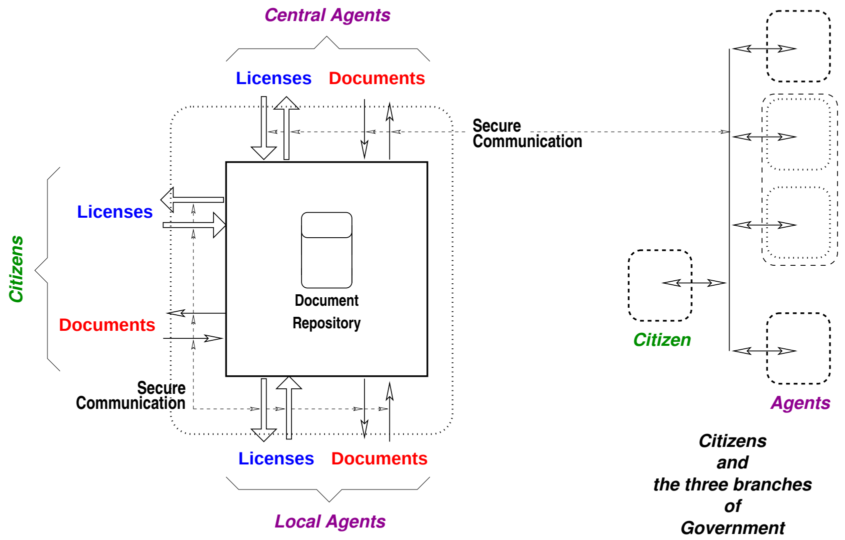

Figure 23: A possible modular software server

Figure 23 intends to show that every actor receives licenses and documents, that is, with each document received there follows a license which details the authorisations, that is, the rights, that that actor is gievn wrt. that document.

Figure 23 further intends to show that any actor may send licenses and documents, that is, each document sent (moved, distributed) there is accompanied with license which details the authorisations, that is, the rights, that the receiving actor is given wrt. that document.

Finally Fig. 23 should be augmented to imply that the square full line box within the dotted rounded-edge left box checks that licenses sent harmonises with licenses received for "derived" documents.

5 Closing

5.1 What Have We Covered?

Three branches of government $\leftrightarrow$ citizens Flow of documents Actors: citizens and agents Documentscreate, edit, copy, move, read, shred, search, compute document version, history, reference, trace authorisation, license $\mathcal{E}^2\mathcal{G}$: $\mathcal{E}$ ssential $\mathcal{E}$-$\mathcal{G}$ overnment

5.2 What Did We Try to Achieve?

A systematic approach to understanding the **domain** of public government. Another approach to asking for $\mathcal{E}^2$-$\mathcal{G}$ overnment. A focus of **law-based documents** A view of **good**

governance : **Transparency of document handling** $\Rightarrow$ **transparency of the 'Rule of Law'.**