

JOSS: RUBRICS

C. L. Baker

March 1967

JOSS: RUBRICS

C. L. Baker^{*}

The RAND Corporation, Santa Monica, California

This paper presents a collection of principles that should be kept in mind when designing a JOSS-like system, when contemplating additions to the JOSS language, or when comparing JOSS with similar systems.

- JOSS[†] is a complete, self-contained system.
- The machine hardware and software are completely "sealed off," in that their nature cannot be detected by the user. JOSS is independent of its implementation.
- The ability to "seal off" JOSS has been "sealed off," in that the user must always deal with JOSS, and not with a higher level system constructed in JOSS language.
- JOSS acts as a single active agent; that is, the user is always communicating with the same set of software that always recognizes messages in the same way. A message does not have to be "passed off" to another processing system to determine its nature.

* Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of The RAND Corporation or the official opinion or policy of any of its governmental or private research sponsors. Papers are reproduced by The RAND Corporation as a courtesy to members of its staff.

† JOSS is the trademark and service mark of The RAND Corporation for its computer program and services using that program.

- JOSS steps are quantized; that is, JOSS never presents itself in a state where it has only partially interpreted and executed a step. Either a step "runs" to its conclusion, or the effect is as if the step had caused no change at all. ("Type" may not finish, but since "Type" causes no changes in the state of JOSS, the effect is the same.)
- Interpretation is distinct from execution and serves to verify that execution will proceed without difficulty. For example, a "Do" command is interpreted only once and may be deleted during execution without ill effects.
- There is a single interpretation for any element within a statement; for example, "\$" (the current value of the line number) always has the same value for all expressions within a "Type" command, even though this command may type many lines.
- Errors detected by JOSS do not lead to catastrophic results; that is, they are detected as potential errors and reported to the user before trouble is caused. The user can repair the damage and continue as if the repair had been made before the step in which the error had occurred had been interpreted.
- JOSS never perceives its own internal state differently than it presents this state to the user. For example, if JOSS reports to the user "Error at step 5.1: x = ???", the response to "Type x." is never "x = 3".
- JOSS doesn't exhibit surprising, anomalous, or

self-contradictory behavior; for example, "Type x if $x \neq 1$." never results in " $x = 1$ ", and error messages report only true errors. (Deliberate "spoofs" can, of course, be constructed, because of the richness and flexibility inherent in the language.)

- JOSS uses a standard typewriter keyboard, and produces output on standard $8\frac{1}{2}$ by 11-in. typewriter paper.
- Control of the JOSS typewriter is proprietary: Either the user has control for typing (termed the green state) or JOSS has control for calculation and possible output (termed the red state). Visual (lights), tactile (keyboard lock), and audible (beep tone) indications inform the user as to which state prevails.
- Except for error messages, user-specified output, or to report completion of file actions, JOSS does not need to produce a typed response to acknowledge completion of a task; the unambiguous switch-to-green action at the console is sufficient for this purpose.
- Input to JOSS is typed in green, output in black.
- JOSS provides a complete hard-copy record of a user session. It is possible to reconstruct any situation from the protocol (except that file item contents may have to be reconstructed from previous protocols, and several attempts may have to be made to reproduce an interrupt at exactly the same spot).
- JOSS uses familiar decimal arithmetic and operates on numbers exactly as would a person with pencil and paper and a good set of tables. JOSS gives true results, rounded to nine decimal digits, for the

elementary operations (add, subtract, multiply, divide, square root, and selected cases of exponentiation; in transcendental function evaluation, great care is taken to hit "magic" values on the nose).

- JOSS provides exact input and exact output for all elements of the language: values, formulas, forms, steps, and parts.
- Editing is not a special mode, but is provided implicitly by step, form, formula, and value labels, augmented by statements in the language to "Type," "File," or "Delete" elements and their aggregations.
- JOSS insists on letter-perfect input and output, and never tries to anticipate or second-guess the user's meaning. The rules for constructing JOSS commands are derived from those for writing well-formed English sentences and linearized algebraic expressions.
- JOSS commands can be read out loud subject only to the difficulty of reading algebraic expressions. Thus, the language can be "thought in" as well.
- The JOSS language is constructed so that wherever possible the general case applies. This leads to simple rules with no exceptions.
- There are well-defined boundaries for the applicability of rules, expressions, etc., which can be discovered without extensive probing.
- There are no experts required to explain JOSS to the user, to extract him from difficulty, or to provide him with special instruction in hardware operation.

- JOSS's behavior can be described, in English, without recourse to a description of the mechanism by which this behavior is implemented.
- File and item keys are provided only to protect the user against inadvertent loss of information. No attempt is made to provide security against deliberate violation of privacy or destruction of information.
- JOSS runs on a dedicated machine, and apart from scheduled maintenance periods and unscheduled machine failures, around-the-clock service is available.
- JOSS is an operatorless system, and the machine is usually unattended.
- The JOSS language cannot, in general, be extended simply, because each of the above factors must be taken into consideration.