



## Object Operating System

INRIA – LORIA – MIRÓ

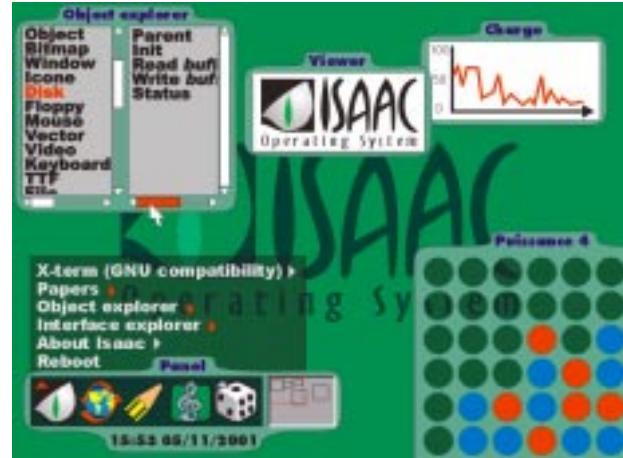
Benoît Sonntag, [sonntag@loria.fr](mailto:sonntag@loria.fr)

Dominique Colnet, [colnet@loria.fr](mailto:colnet@loria.fr)

<http://IsaacOS.com>

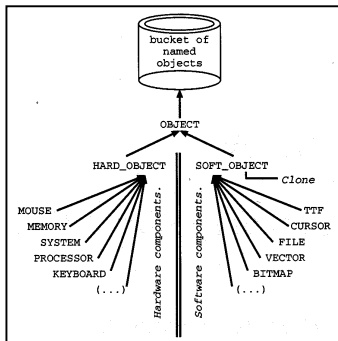
### L'objet au cœur du système d'exploitation.

L'évolution des technologies et du matériel nous amène à revoir la conception du noyau des systèmes d'exploitation actuels. Nous pensons que l'introduction de concepts **objets de haut niveau** au cœur même des systèmes d'exploitation simplifie et optimise l'utilisation des divers composants physiques d'un ordinateur. Ce type de démarche apporte un réel progrès aussi bien dans l'adaptation dynamique du système, que dans la simplification et le pouvoir expressif du code.



### Modularité, flexibilité et expressivité

*Isaac* est avant tout un nouveau système d'exploitation objet. Il allie modularité, flexibilité, adaptabilité, évolutivité et rapidité. D'un genre nouveau, il ne possède aucun noyau, ni micro noyau et fonctionne directement sur l'architecture de la machine. Aucune machine virtuelle n'est nécessaire et aucune autre entité que l'objet n'est présent. Actuellement, les objets qui constituent *Isaac* tirent pleinement partie de sa flexibilité pour modéliser de manière élégante et répondre aux attentes d'un système traditionnel du noyau jusqu'à l'interface utilisateur. La plus grande particularité d'*Isaac* est de garder suffisamment de sémantique dans le binaire pour que les objets conservent toutes leurs expressivités objets **durant l'exécution**. Le système est capable d'avoir une vision suffisamment fine des objets pour les manipuler, les inter-connecter, les cloner, ou changer les liens de multi-héritages durant l'exécution même.



### Uniformité, rapidité

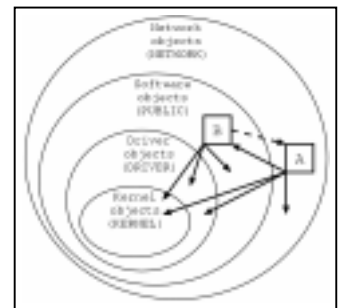
*Isaac* est constitué de son propre modèle objet à base de prototypes, le langage *Lisaac* qui a permis de le réaliser ressemble conceptuellement à *Self* avec des atouts directement liés à la conception de système portable et proche de la machine. Son compilateur produit du code **binaire efficace** grâce à des optimisations de haut niveau : produit cartésien, prédiction de type, suppression de la liaison dynamique, suppression de la récursivité, spécialisation du code, ... Par son modèle **simple et homogène**, nous sommes en droit de penser qu'il devient possible de construire et de faire évoluer les applications par simple ajout d'objet ou par mutation des prototypes existants via le clonage et l'héritage dynamique.

### Légèreté, adaptabilité

Son spectre d'application est varié : sa forte modularité et sa légèreté (*moins d'un méga indispensable*) lui permettent de s'adapter naturellement à des architectures exotiques comme les **assistants personnels (palm)** ou **l'informatique embarquée**. En effet, il devient facile d'assembler son système d'exploitation 'à la carte' pour répondre au mieux à des contraintes industrielles et matérielles.

### Sécurité, stabilité

Sans une baisse des performances, *Isaac* possède des mécanismes très particuliers de protection directement gérée par le matériel qui permet de garantir une **sécurité inviolable** sur les objets essentiels du système. Sa forte modularité provenant des concepts objets lui permet d'aguerrir une certaine **stabilité naturelle** : Un objet endommagé n'affecte pas (*ou peu*) l'intégrité du système.



*The power of simplicity at work for operating system !*